The Traffic Equilibrium Problem with Nonadditive Path Costs

Steven A. Gabriel [1]
Building 221
Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439
gabriel@mcs.anl.gov

David Bernstein[2]
Department of Civil Engineering and Operations Research
Princeton University
Princeton, NJ 08544
dhb@dragon.princeton.edu

Preprint # MCS-P533-0895
Mathematics and Computer Science Division
Argonne National Laboratory
August 21, 1995

Abstract

In this paper we present a version of the (static) traffic equilibrium problem in which the cost incurred on a path is not simply the sum of the costs on the arcs that constitute that path. We motivate this nonadditive version of the problem by describing several situations in which the classical additivity assumption fails. We also present an algorithm for solving nonadditive problems that is based on the recent NE/SQP algorithm, a fast and robust method for the nonlinear complementarity problem. Finally, we present a small example that illustrates both the importance of using nonadditive costs and the effectiveness of the NE/SQP method.

[1]The research was conducted while this author was a member of the Mathematical Sciences Department of The Johns Hopkins University and the Mathematics and Computer Science Division of Argonne National Laboratory (ANL). At ANL, this work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

[2]Corresponding Author.

Keywords: traffic equilibrium problem, nonlinear complementarity problem, networks, road pricing

1 Introduction

Paraphrasing Wardrop [34], the (static) traffic equilibrium problem is to find a set of path flows that satisfy certain demand constraints and have the property that the cost on all used paths connecting an origin-destination pair is equal and less than or equal to the cost on all unused paths connecting that pair. In order to prove existence/uniqueness results and develop convergent algorithms, this problem has been formulated as a *nonlinear program* (NLP) [5], a *nonlinear complementarity problem* (NCP) [2, 33], a *variational inequality problem* (VI) [10, 11, 12, 30], and a *fixed-point problem* (FP) [4].

Though the traffic equilibrium problem generally is stated in terms of path flows, paths usually are considered to be a nuisance by the developers of algorithms. There are two reasons. First, path "names" generally consist of the list of links or nodes that constitute the path, and these "names" can become quite long and hence difficult to store and manipulate. Second, since the path set cannot (efficiently) be completely enumerated, the number of paths is not known *a priori*, thereby complicating memory management and creating other software engineering difficulties.

In order to avoid the nuisance of storing and manipulating paths, it is quite common (in both the theoretical literature and in practice) to assume that the cost on a path is the sum of the costs on the links that make up that path. This assumption makes it possible to use (what is generally referred to as) an arc formulation of the problem and not store path flows.

The purpose of this paper is twofold: (1) to show that, although it is convenient, the additivity assumption is inappropriate in a variety of different situations and (2) to describe a method for solving a nonadditive version of the problem. The particular model and algorithm we present allows for asymmetric elastic demand functions and asymmetric and nonadditive cost functions. Though we do not discuss it here, these results can also be applied to probabilistic versions of the problem (e.g., stochastic user equilibrium and entropy models), although we do not discuss such cases here.

2 Equilibrium with Nonadditive Path Costs

The (static, deterministic) traffic equilibrium problem (TEP) is typically set on a network comprising a set of *arcs*, $\mathcal{A}$, and a set of *nodes*, $\mathcal{N}$, with cardinalities $n_{\mathcal{A}}$ and $n_{\mathcal{N}}$, respectively. Associated with this network is a set of *origin-destination pairs*, I, with cardinality n_I. People travel between a particular origin-destination (O-D) pair $i \in I$ on a path in the set P_i, the set of paths connecting O-D pair i. The cost experienced by a person using path p is given by $C_p : R_+^{n_P} \to R_+$ where n_P denotes the cardinality of the set of paths $P = \cup_{i \in I} P_i$.

In the most general version of this problem, path costs can be a function of the entire vector of path flows and the number of people traveling between O-D pair i, and the demand function $D_i : R_+^{n_I} \to R_+$ is a function of the vector of (minimum) O-D travel costs, $(\min_{r \in P_j} C_r(F) : j \in I)$. In this case, an equilibrium is typically defined as follows (see [7] for a discussion of alternative definitions):

Definition 1 *A path flow vector, $F \in R_+^{n_P}$, is said to be an* elastic traffic equilibrium *iff*

$$F_p > 0 \Rightarrow C_p(F) = \min_{r \in P_i} C_r(F) \tag{1}$$

for all $i \in I$, $p \in P_i$, and

$$\sum_{p \in P_i} F_p = D_i(\min_{r \in P_j} C_r(F) : j \in I) \tag{2}$$

for all $i \in I$.

The inelastic equilibrium problem is a special case of the elastic problem in which $D_i(\cdot)$ is constant.

Additive Costs

Perhaps the most natural way to formulate the TEP as an NLP, NCP, VI, or FP is to use path variables. However, these formulations have not been widely used because they are thought to be difficult to solve. In particular, when path variables are used, either the paths must be completely enumerated before the algorithm begins or the paths must be identified "on the fly". The first solution is computationally burdensome, and the second is thought to be cumbersome because the incidence relationship between paths and arcs must be maintained and manipulated and the number of paths is not known *a priori*.

To overcome these difficulties, one often assumes that the cost on a path p is simply the sum of the costs on each arc in p. Specifically, letting $\Delta = [\delta_{ap}]$ represent the arc-path incidence matrix, $c : R_+^{n_A} \rightarrow R^{n_A}$ the arc cost function, and $f \in R_+^{n_A}$ represent the arc flow vector, the *additive* model assumes

$$C(F) \quad = \Delta^T c(f), \tag{3}$$

with $f = \Delta F$; here C is the vector of path cost functions.

The significance of this assumption is that it allows the path flow variables to be removed from the objective function of the NLP formulation of TEP (in the case of symmetric arc cost functions) and from the inequality in the VI formulation of TEP (in the case of general arc cost functions). Although the path flow variables remain in the constraint set, it becomes possible to solve TEP without storing path flows. This has two important implications from a software development standpoint. First, it means that the number of decision variables that need to be stored is known in advance (i.e., $n_A + n_I$), thus greatly simplifying memory allocation. Second, the decision variables that are being stored can be easily identified (i.e., by their arc number in the case of arc flows and by their O-D number or associated pair of node numbers in the case of O-D demands/costs).

Situations in Which Costs Are Not Additive

Unfortunately, although they have been essentially ignored in the past by both researchers and practitioners, there are many situations in which the additivity assumption is inappropriate. These situations are particularly important today, in light of recent legislation such as the Intermodal Surface Transportation Efficiency Act (which promotes congestion pricing programs) and the Clean Air Act Amendments (which mandate a reduction in automobile emissions in many cities). That is, a variety of transportation policies are being considered today that cannot adequately be evaluated by using additive path costs.

Nonlinear Valuation of Travel Time

The cost on a path typically includes, at a minimum, the time costs and the money costs of using that path. Using an additive model, one typically assumes that the arc cost functions have the following form:

$$c_a(f) = \lambda_a + \eta_1 t_a(f) + \eta_2 t_a(f)$$

for all arcs a, where λ_a is the (distance-based) financial cost of using arc a, (e.g., tolls and distance-based operating costs such as maintenance), $t_a(f)$ is the time to traverse arc a given the current arcs flows f, η_1 is the time-based operating costs (e.g., gasoline consumption), and η_2 is the dollar value of time.

However, it has often been observed [19] that people value time nonlinearly. That is, small amounts of time have relatively low value whereas large amounts of time are very valuable. As a result, one must first calculate the total time on the path and apply the value of time function to this total. Assuming that time-based operating costs are still a linear function of the total travel time, one is left with path cost functions of the following form:

$$C_p(F) = \sum_{a \in \mathcal{A}} \delta_{ap}(\lambda_a + \eta_1 t_a(f)) + g_p\left(\sum_{a \in \mathcal{A}} \delta_{ap} t_a(f)\right),$$

where $g_p(\cdot)$ is an increasing function that converts time to money for path p. In actual applications, g_p is unlikely to vary across paths although one can imagine situations in which the value of time varies with the attributes of the path (e.g., how pleasant the path is).

Nonadditive Tolls and Fares

When discussing the nonlinear valuation of travel time we assumed that the toll on a path was simply the sum of the tolls on the arcs that make up that path. Unfortunately this is often not the case. It is quite common for both highway tolls and transit fares (which are of interest because it is quite common to consider multimodal equilibria) to be nonadditive. For example, consider the following fares on the BART system:

		To				
		Fremont	Union City	South Hayward	Hayward	Bay Fair
	Fremont		0.90	0.90	0.90	1.90
From	Union City			0.90	0.90	0.90
	S. Hayward				0.90	0.90

It is easy to see, for example, that the fare from Fremont to Bay Fair ($1.90) does not equal the fare from Fremont to Union City ($0.90) plus the fare from

Union City to South Hayward ($0.90) plus the fare from South Hayward to Hayward ($0.90) plus the fare from Hayward to Bay Fair ($0.90).

In fact, almost no toll roads or transit systems in the United States have an additive toll/fare structure. Instead, one must work with the path-specific financial costs directly, which, even ignoring nonlinear value time functions, makes the path cost functions nonadditive.

Emissions Fees

It has long been argued that emissions fees should be used to internalize the externalities associated with automobile emissions. This strategy may result in nonadditive costs for two reasons. First, there is some evidence that emissions of hydrocarbons and carbon monoxide are a nonlinear function of travel times. Second, there is little doubt that social costs are a nonlinear function of emissions [20]. Hence, in order to set tolls equal to the the difference between the social marginal cost and the private average cost, they will need to be path-specific. Such a path-specific toll structure immediately leads to nonadditive path cost functions.

The Nonlinear Complementarity Formulation

Combining the three observations above, one sees that a general path cost function would have the following form:

$$C_p(F) = \Lambda_p(F) + \sum_{a \in p} \eta_1 t_a(f) + g_p \left(\sum_{a \in p} .\delta_{ap} t_a(f) \right), \tag{4}$$

where Λ_p now denotes the path-specific fincancial costs (which are allowed to vary with flow levels to allow for different kinds of pricing schemes). Most existing "path flow" formulations of TEP continue to be appropriate when using such a nonadditive path cost function. For our purposes, the most important of these is the NCP formulation [2]. In this formulation, the problem is to find the (path flows, O-D costs) vector pair (F, u) such that

$$\begin{array}{lll} G_F(F,u) = C(F) - \Gamma u \geq 0 & F \geq 0 & G_F(F,u)^T F = 0 \\ G_u(F,u) = \Gamma^T F - D(u) \geq 0 & u \geq 0 & G_u(F,u)^T u = 0, \end{array} \tag{5}$$

where Γ is the path–Origin–Destination pair incidence matrix.

We make the weak assumptions that the functions C and D are differentiable and that for each path p, the function C_p is positive. Additionally, we assume that D_i is a nonnegative function for all $i \in I$. As a result, an equivalent system has $\Gamma^T F - D(u) \geq 0$ replaced by $\Gamma^T F - D(u) = 0$, which is the more usual form of the conservation of demand constraint [2]. We note that if the ith O-D pair has positive demand (i.e., $D_i(u) > 0$) the u_i variable measures the cost on the cheapest path value for O-D pair i. However, when $D_i(u) = 0$, it is possible that the u_i variable can be less than or equal to the cheapest path value; see Lemma 1 in Section 3.

In what follows, we will need to assume that the traffic equilibrium problems being solved are guaranteed to have a solution. We make use of the following result, which is Theorem 5.4 in [2].

Theorem 1 *Suppose (N, A) is a strongly connected network and that $C_p : R_+^{n_p} \rightarrow R_+$ is a positive continuous function for all $p \in P$. Also suppose that for all $i \in I$, $D_i : R_+^{n_i} \rightarrow R_+$ is a nonnegative continuous function that is bounded from above. Then TEP has a solution.*

3 An Algorithm for Solving the Nonadditive Problem

As discussed above, perhaps the biggest advantage of the additive model is that it can be solved without the necessity of storing path flows. Many of the most widely applied algorithms take advantage of this fact [14, 23, 26, 27]. However, two types of schemes do generate and store path flows as needed: simplicial decomposition and column generation.

In simplicial decomposition, the set of feasible flows is given as a bounded convex polyhedron so that each element can be described as a convex combination of the extreme flows of this set. The algorithms of this type proceed by working on the convex hull of a working set of extreme points, checking for termination conditions to be met. The weights associated with the current set of extreme points are then taken to be the decision variables. As a result, a significantly smaller number of variables is needed. Some recent examples of the simplicial decomposition approach include the work of Pang and Yu [29], who combined a linearization of the VI form of the problem with simplicial decomposition, and Lawphongpanich and Hearn [22], Smith [31, 32], and Marcotte and Guélat [25], whose simplicial decomposition approaches used a so-called gap function, a nonnegative measure that was zero only at an equilibrium point. For an extensive history on this approach, the interested reader should see the recent survey by Larsson and Patricksson [21].

In the column generation approach, path flows are generated only when needed, thereby reducing the computational burden. The key is to have the algorithm identify those paths that will have flow on them in an equilibrium solution. Examples of column generation methods as applied to the TEP include the early work of Leventhal, Nemhauser and Trotter [24], who studied the case of separable costs; Bertsekas and Gafni [8], who combined a projection method for the associated VI with a decomposition by O-D pairs; and Aashtiani [1] whose Ph.D. dissertation concerned a similar approach. See the survey [21] for further details.

In this section, we present a new algorithm for solving the path-flow formulation of the TEP that is based on the recent NE/SQP method (for nonsmooth equations/sequential quadratic programming) for solving the NCP [28, 17]. The primary advantage of this algorithm is its robustness; unlike other approaches, each subproblem is guaranteed to have a solution.

Note that throughout this discussion, for vectors $v \in R^n$, we have indicated subvectors by either v_y or v_γ. Here y is a vector of variables and so v_y refers to all components of v relating to these variables. Alternatively, we have also used the index set $\gamma \subseteq \{1, 2, \ldots, n\}$ to describe a subvector v_γ of v; matrices follow the same convention.

A Review of the NE/SQP Method

NE/SQP is a recent method for solving general nonlinear complementarity problems. It is has been shown to be globally convergent and fast (Q-quadratic rate) as well as robust, in the sense that the direction-finding subproblems are always solvable [28, 17].

For a function $G : R^n_+ \to R^n$, NCP(G) is to find an $x \in R^n$ such that

$$x \geq 0, \quad G(x) \geq 0, \text{ and } \quad G(x)^T x = 0.$$

The basis for the NE/SQP method is to solve NCP(G) by first transforming it into the problem of finding the zero of a certain set of nonsmooth equations.

Specifically, let the function $H : R^n_+ \to R^n$ be defined by

$$H(x)_i = \min(x_i, G_i(x)) \quad i = 1, \cdots, n. \tag{6}$$

It is not hard to see that a zero of this function H corresponds exactly to a solution to NCP(G). Unfortunately, because of the presence of the min operator, this function is not differentiable (in the sense of Fréchet), so that standard algorithms such as Newton's method cannot directly be applied. However, NE/SQP is actually a nonsmooth extension of the Gauss-Newton method as applied to this function H.

Very much related to H is the *norm function* $\theta : R^n_+ \to R^n_+$ defined by

$$\theta(x) = \frac{1}{2}\|H(x)\|^2, \tag{7}$$

where we take $\|\cdot\|$ to be the Euclidean norm throughout this paper. As a result, we see that NCP(G) can be cast as the nonsmooth, nonconvex optimization problem

$$\begin{array}{ll} \text{minimize }_x & \theta(x) \\ \text{such that} & x \geq 0. \end{array} \tag{8}$$

The basic scheme with NE/SQP is as follows: having an estimate x^k of the solution, a new iterate x^{k+1} is generated according to the rule

$$x^{k+1} = x^k + \tau_k d^k,$$

where d^k is a suitable search direction and τ_k is the associated step length needed for global convergence of the method. The calculation of the search direction entails the solution of a certain convex quadratic program (QP), which we will now explain.

Let $\phi : R^n_+ \times R^n \to R^n_+$ be defined as

$$\phi(x,d) = \frac{1}{2}\|H(x) + M(x)d\|^2, \tag{9}$$

where $M(x)$ is the $n \times n$ matrix defined as follows

$$M(x) = \begin{pmatrix} I_{\alpha\alpha} & 0 \\ \nabla_\alpha G_\beta & \nabla_\beta G_\beta \end{pmatrix}, \tag{10}$$

where

$$\begin{array}{lll} I_G(x) = \{i : G_i(x) < x_i\}, & I_x(x) = \{i : G_i(x) > x_i\}, & I_e(x) = \{i : G_i(x) = x_i\}, \\ \alpha = I_x(x) \cup I_e(x), & \beta = I_G(x), & \end{array}$$

and $I_{\alpha\alpha}$ is the identity matrix of order α.

Having the iterate x^k, the associated direction-finding convex quadratic subproblem is of the form

$$\begin{array}{ll} \text{minimize }_d & \phi(x^k, d) \\ \text{subject to} & x^k + d \geq 0. \end{array} \tag{11}$$

Note that the direction $d = 0$ is always feasible to this QP because each iterate x^k is maintained nonnegative; see (8). As a result, the feasible region is a nonempty polyhedron. Hence, taken together with the fact that the objective function is a quadratic bounded below by zero, this QP will always have a solution (by the Frank-Wolfe theorem [15]). In addition, this is a relatively easy QP to solve because it has a convex objective function and simple bound constraints.

Two conditions are sufficient to guarantee the convergence of the NE/SQP method. The first condition is *s-regularity* and generalizes the idea of nonsingularity.

Definition 2 *A nonnegative vector x is said to be s-regular if the following linear inequality system has a solution in y:*

$$\begin{array}{lll} x_i + y_i & = 0 & i \in I_x^+(x) \\ G_i(x) + \nabla G_i(x)^T y & = 0 & i \in I_G^+(x) \\ x_i + y_i & \geq 0 & i \in I_G^0(x) \\ G_i(x) + \nabla G_i(x)^T y & \geq 0 & i \in I_G^0(x) \\ x_i + y_i & \leq 0 & i \in I_e^+(x) \\ G_i(x) + \nabla G_i(x)^T y & \leq 0 & i \in I_e^+(x) \\ y_i & = 0 & i \in I_e^0(x), \end{array} \tag{12}$$

where

$$I_G^+(x) = \{i : x_i > G_i(x), x_i > 0\} \quad I_G^0(x) = \{i : x_i > G_i(x), x_i = 0\}$$

and similarly for $I_e(x)$ and $I_x(x)$.

The second condition is *b-regularity* and ensures the boundedness of the sequence of search directions $\{d^k\}$.

Definition 3 *A nonnegative vector x is said to be b-regular if for every index set δ with the property that*

$$I_G^+(x) \subseteq \delta \subseteq I_G(x) \cup I_e(x),$$

the principal submatrix $\nabla_\delta G_\delta(x)$ is nonsingular.

The main convergence results for NE/SQP can now be summarized as follows:

Theorem 2 *Let $G : R^n_+ \to R^n$ be a once continuously differentiable function, and $x^0 \geq 0$ be arbitrary. Suppose that x^* is an accumulation point of an infinite sequence of iterates $\{x^k\}$ generated by the NE/SQP method, and x^* is both b-regular and s-regular. Then, x^* solves NCP(G). Moreover, the following statements hold:*

(a) *there exists an integer $K > 0$ such that for all $k \geq K$, the step size $\tau_k = 1$, hence, $x^{k+1} = x^k + d^k$;*

(b) *the sequence $\{x^k\}$ converges to x^* Q-superlinearly, i.e.,*

$$\lim_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0;$$

(c) *if ∇G is Lipschitzian in a neighborhood of x^*, then*

$$\limsup_{k\to\infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2} < \infty.$$

Using NE/SQP to Solve the Elastic, Nonadditive TEP

In this section, we modify the basic NE/SQP method presented above for use with the NCP formulation of the traffic equilibrium problem as given in (5). The essential idea is to keep a working set of paths W whose elements can have nonzero flow. The associated path flows vector of size $n_W \times 1$ is denoted as F_W; here $n_W = |W|$. The remaining inactive paths have their flow automatically set to zero and the associated indices are collected into the set $\bar{W}$ where $|\bar{W}| = n_{\bar{W}}$. It is understood that the number of active paths n_W is generally much less than the number of total paths n_P. In combination with the n_I O-D minimum times collected into the vector u, we attempt to solve the associated NCP of size $n_W + n_I$ rather than the NCP with the $n_P + n_I$ complete set of variables. The collection of indices for each of these reduced NCPs is given by $S = W \cup I$, where $|S| = n_S = n_W + n_I$. Of course the selection of which paths will be in the initial working set is important as well as the method for updating the set W. We discuss these issues in more detail in what follows.

A crucial point in making the path generation NE/SQP method work is to identify conditions that will allow us to conclude that we have actually solved the overall NCP of size $n = n_P + n_I$ without enumerating all paths. We will provide a lemma that will outline these conditions, but first we need to introduce some notation that associates the functions used in the NE/SQP method with the size of the reduced NCP under consideration.

We can expand the function $G(\cdot)$ given in (5) to

$$\begin{array}{rcl} G_{F_W}(F_W, F_{\bar{W}}, u) & = & C_W(F_W, F_{\bar{W}}) - \Gamma_{W\bullet} \quad u \\ G_{F_{\bar{W}}}(F_W, F_{\bar{W}}, u) & = & C_{\bar{W}}(F_W, F_{\bar{W}}) - \Gamma_{\bar{W}\bullet} \quad u \\ G_u(F_W, F_{\bar{W}}, u) & = & \Gamma^T_{\bullet W} F_W + \Gamma^T_{\bullet \bar{W}} F_{\bar{W}} - D(u), \end{array} \tag{13}$$

where $A_{\alpha\bullet}, A_{\bullet\beta}$ denote respectively, rows and columns of the matrix A indexed by the sets α and β.

The reduced NCP automatically sets the inactive path flows equal to zero (i.e., $F_{\bar{W}} = 0$) and ignores the components $G_{F_{\bar{W}}}$ so that we get the reduced NCP as

$$\begin{array}{llll} G_{F_W}(F_W, u) & = C_W(F_W) - \Gamma_{W\bullet} u & F_W \geq 0 & G^T_{F_W} F_W = 0 \\ G_u(F_W, u) & = \Gamma^T_{\bullet W} F_W - D(u) \geq 0 & u \geq 0 & G_u(F_W, u)^T u = 0. \end{array} \tag{14}$$

We have made the rather weak assumption that for path p, the cost function $C_p(F)$ does not depend on paths with zero flow; that is, $C_p(F) = C_p(F_W)$.

The related function $H : R^n_+ \to R^{n_S}$ is given as

$$H_S(x_S) = \min(x_S, G_S) \tag{15}$$

where the subscript S refers to those active indices in $S = W \cup I$ with $F_{\bar{W}} = 0$, for example, the vector

$$x_S = \begin{pmatrix} F_W \\ u \end{pmatrix}.$$

Also, we define $\theta_S : R^n_+ \to R_+$ as

$$\theta_S(x_S) = \frac{1}{2}\|H_S(x_S)\|^2 \tag{16}$$

and the subproblem objective function $\phi_S : R^n_+ \times R^n \to R_+$ as

$$\phi_S(x_S, d_S) = \frac{1}{2}\|H_S(x_S) + M_{SS}(x_S) d_S\|^2, \tag{17}$$

where d_S is conformal with x_S and $M_{SS}(x_S)$ is a principal submatrix of $M(x)$ with

$$x = \begin{pmatrix} x_S \\ x_{\bar{S}} \end{pmatrix}, \text{ and } x_{\bar{S}} = 0. \tag{18}$$

Note that, without loss of generality, we have arranged the vector x so that the first n_S components relate to x_i for $i \in S$. The related reduced QP subproblem is thus of the form

$$\begin{array}{ll} \text{minimize}_{\ d_S} & \phi_S(x_S^k, d_S) \\ \text{subject to} & x_S^k + d_S \geq 0. \end{array} \tag{19}$$

Lastly, let the forcing function $z_S : R_+^n \times R^n \rightarrow R_+$ be defined as

$$z_S(x, d) = \frac{1}{2}\|M_{SS}(x_S)d_S\|^2. \tag{20}$$

Note that by setting all the inactive paths to zero, without loss of generality, we can express the functions above in terms of the vector x_S rather than the entire vector $x^T = (F_W^T \;\; u^T \;\; F_{\bar{W}}^T)$. We see that the functions H, θ, ϕ, and z take the n vector x as their argument of which $n_{\bar{W}}$ of the components are fixed at a value of zero. In that sense, one can think of them also as taking vectors of size n_S as their arguments. The important point is that from (13), including the values of $F_{\bar{W}} = 0$ into the vector x or leaving them off makes no difference in the value of (G_{F_W}, G_u) and related functions.

The first result states conditions indicating when a solution of the current reduced NCP coincides with a solution of the overall TEP.

Lemma 1 *Let $S = W \cup I$, $x_S \in R_+^{n_S}$ and $F_{\bar{W}} = 0$. Then, the vector $x \in R^{n_P+n_I}$ given by (18) solves the TEP if and only if*

(i) *$\theta_S(x_S) = 0$, and*

(ii) *u_i is less than or equal to the length of a shortest path for O-D pair i, for all $i \in I$.*

Proof

Note first that the terms $\theta_S(x_S)$ and $\theta_S(x)$ are equal and are used interchangeably in what follows. The former term implicitly sets $x_i = 0 \; \forall i \in \bar{W}$ whereas the latter term does the same only explicitly. A similar convention is adopted for other functions involving x. We have the following

$$\begin{aligned} \theta(x) = \; & \textstyle\sum_{j \in W \cup I} \theta_j(x) + \\ & \textstyle\sum_{j \in \bar{W} \cap (I_x \cup I_e)} \theta_j(x) + \\ & \textstyle\sum_{j \in \bar{W} \cap I_G} \theta_j(x). \end{aligned}$$

We will first show that (i) and (ii) imply that x is a solution to TEP. Since $\theta_S(x_S) = 0$, we see that the first summand is zero. Also, since $x_j = 0$ for all

$j \in \bar{W}$ and $\theta_j(x) = \frac{1}{2}x_j^2$ for all $j \in \bar{W} \cap (I_x(x) \cup I_e(x))$, the second term is also equal to zero. And lastly, for any path $j \in \bar{W} \cap I_G(x)$, with associated O-D pair k, letting the matrices $\Delta = [\delta_{ap}]$ and $\Gamma = [\gamma_{pi}]$, we must have

$$G_j(x_S) = C_j - \sum_{i \in I} \gamma_{ji} u_i < 0 = x_j$$

or that path j has less cost than u_k. So in light of the shortest path premise (ii), the set $\bar{W} \cap I_G(x)$ must be empty. Noting that the empty sum is equal to zero, we see that (i) and (ii) imply that $\theta(x) = 0$ or that x is a solution to NCP(G).

As for the other direction, we note first that θ is the sum of nonnegative terms θ_j. Therefore, x is a solution to TEP if and only if $\theta_j(x) = 0 \ \forall j = 1, 2, \ldots, n$. As a result,

$$\theta_S(x) = \sum_{j \in W \cup I} \theta_j(x) = 0.$$

It remains to show that u_i is less than or equal to the length of a shortest path for O-D pair i for each $i \in I$. Assume not, then there must be a path j serving O-D pair k such that

$$G_j(x) = C_j - \sum_{i \in I} \gamma_{ji} u_i < 0.$$

Clearly, j cannot be in $I_x \cup I_e$, because if so, this would mean that $F_j \leq G_j(x) < 0$, a contradiction to the fact that path flows are maintained nonnegative. If $j \in I_G$, then

$$0 = \theta_j(x) = \frac{1}{2} G_j(x)^2 > 0,$$

a contradiction. Consequently, we see that condition (ii) is satisfied. $\square$

The importance of this result is that we need only solve NCPs of reduced size and check shortest path conditions to actually solve the overall NCP. This result is the main justification for the path generation method. It is assumed that the shortest path calculations can be performed efficiently on the n_I origin-destination pairs and that the reduced NCPs of size $n_S = n_W + n_I$ are still computationally manageable.

The path generation NE/SQP approach can now be presented. The main idea is to apply NE/SQP to the reduced NCP of order $n_S = n_W + n_I$. If a correct set of active paths W is selected, then, barring any lack of descent in θ (due to the condition $\phi_S(x_S^k, d_S^k) = \theta_S(x_S^k)$ or the lack of regularity at an accumulation point), by Lemma 1, if all the variables u_i represent times less than or equal to the shortest O-D paths, solving the smaller problem is sufficient to solving the overall NCP of size $n_P + n_I$.

During an intermediate step of this modified NE/SQP algorithm, the method may stall because the wrong set of active paths has been identified. Stalling here means that $\theta_S(x_S^k) = 0$, but there still exists an inactive path p serving O-D pair k with lower cost than the current value of u_k. In particular, then we must have $G_p(x_S^k) < 0$. As a result, a change of the index sets W and $\bar{W}$ is needed. Note that we have ignored the case that $\theta_S(x_S^k) = \phi_S(x_S^k, d_S^k)$, which could also have produced nondescent or stalling in θ. Ignoring this case is reasonable because in the numerical experiments in [28, 17], this condition was not checked yet convergence of the method was not hampered. Also, even if this condition were encountered during the running of the algorithm, we could just restart at a new point.

Of course, it is important to consider which path or paths should be brought into the working set W. We discuss first the case of allowing just one path to enter. Since we are ultimately checking for paths that violate the shortest path conditions for u, it is reasonable to bring in such a violating path. That is,

$$\begin{aligned} W &\leftarrow W \cup \{p\} \\ \bar{W} &\leftarrow \bar{W} - \{p\}, \end{aligned}$$

where $p \in \bar{W}$ is such that $G_p(x_S^k) < 0 = F_p^k$. We will call p a *candidate* entering path. At this point, there are two questions to answer. First, having selected a path p to potentially join the working set, to what level above zero do we raise its value? Second, if there is more than one path to choose from, that is, $|\{p \in \bar{W} : G_p(x_S^k) < 0\}| > 1$, how do we select which violating paths to enter? In what follows, we will answer both questions and provide a computationally attractive approach that avoids solving the QP subproblem when a new path is added to the working set.

Suppose that we have identified a path p as described above. One option is to simply restart NE/SQP with the new set of indices

$$\begin{aligned} S' &= W' \cup I \\ W' &= W \cup \{p\} \\ \bar{W}' &= \bar{W} - \{p\}. \end{aligned}$$

This version of the algorithm would necessarily include each candidate entering path, taken one at a time. The new NCP would be of size $n_W + 1 + n_I$

and the current iterate of size $|S'| = n_{S'}$ would be

$$x_{S'}^k = \begin{pmatrix} x_S^k \\ 0 \end{pmatrix}.$$

The mechanics of the algorithm would work as follows. Having the original working set of paths, the value of θ_S would be driven to zero. If the u variables were less than or equal to shortest path values, then by Lemma 1 we would be done. Otherwise, at this iteration, which we will denote as k_1, we would let the first new path $p \in \bar{W}$ enter the working set W. When this new path p entered, we see that

$$\begin{aligned} \theta_{S'}(x_{S'}^k) &= \textstyle\sum_{j \in S'} \theta_j(x_{S'}^k) \\ &= \textstyle\sum_{j \in S} \theta_j(x_{S'}^k) + \theta_p(x_{S'}^k) \\ &= \theta_S(x_S^k) + \frac{1}{2} G_p(x_S^k)^2 \\ &> \theta_S(x_S^k), \end{aligned}$$

so that the norm function θ would necessarily increase. The algorithm would then attempt to drive $\theta_{S'}$ to zero. This pattern would be repeated a finite number of times, with k_i representing the iteration number at which the ith new path is added to the working set. Eventually, this method would converge to a solution of the overall problem, or we would use all the paths ($W = P$) without converging.

However, some computational savings can be gained if we made use of the information obtained from iteration k_i in iteration $k_i + 1$. More specifically, since the purpose of the QP subproblem is to generate a descent direction for θ at the current iterate, if we can find such a direction without actually solving a new subproblem, then we will have made some computational savings. This is the approach we have adopted in this paper.

Lemma 2 is related to avoiding solving the QP subproblem exactly.

Lemma 2 *Let $S = W \cup I$ with $\theta_S(x_S) = 0$. Then, $M_{SS}(x_S)d_S = 0$ where d_S is an optimal subproblem solution.*

Proof

Using the nonnegativity of ϕ and the optimality of d_S, we conclude that

$$\theta_S(x_S) = 0 \le \phi_S(x_S, d_S) \le \phi_S(x_S, 0) = \theta_S(x_S),$$

so that

$$\phi_S(x_S, d_S) = \phi_S(x_S, 0).$$

This statement is equivalent to

$$M_{SS}(x_S)d_S = 0$$

by Proposition 2 (b) [28] and the definition of $z_S(\cdot,\cdot)$. $\square$

The relevance of this lemma with the algorithm being presented is that at iteration k_i, since a new path p is to be added, we must have

$$\theta_S(x_S) = 0 \Rightarrow M_{SS}d_S = 0.$$

We will use this condition in later computations.

At iteration k_i, we are given the current set of indices $S^{k_i} = W^{k_i} \cup I$ and the current iterate $x_{S^{k_i}}^{k_i}$ as well as the new counterparts $S^{k_i+1} = W^{k_i+1} \cup I$ and $x_{S^{k_i+1}}^{k_i+1}$. For notational simplicity, we will denote, respectively, the current set and iterate by $S = W \cup I$ and x_S and the new versions by $S' = W' \cup I$ and $x_{S'}$, where $W' = W \cup \{p \in \bar{W} : G_p(x_S^k) < 0\}$. The search direction d, computed as a solution to the QP subproblem at iteration k_i, will also follow the same notational convention.

In what follows, we will present calculations that will be useful for avoiding a complete resolving of the QP subproblem at iteration k_i. Notationally, we will take the matrix $M_{SS}(x_S)$ and add columns and rows referring to the new paths being added. The result is a matrix of the following form (the argument x_S has been dropped for notational convenience):

$$\begin{pmatrix} M_{SS} & M_{SN} \\ M_{NS} & M_{NN} \end{pmatrix},$$

where N is the set of indices for the new paths being added, $n_N = |N|$, $M_{SS} \in R^{n_S \times n_S}$, $M_{SN} \in R^{n_S \times n_N}$, $M_{NS} \in R^{n_N \times n_S}$, and $M_{NN} \in R^{n_N \times n_N}$. It is not hard to see that when $|N| > 1$, the new value of θ (i.e., $\theta_{S'}$) will also necessarily increase and the previous logic remains valid [3]. As it turns out, the quantities M_{SN}, M_{NS}, and M_{NN} are quite easy to compute and are key in the analysis that will follow.

Lemma 3 *Let N denote the index set of candidate entering paths. Then we have the following:*

[3] Specifically, for $S' = S \cup N$, we have

$$\theta_{S'}(x_{S'}^k) = \sum_{j \in S'} \theta_j(x_{S'}^k) = \theta(x_S^k) + \sum_{j \in N} \frac{1}{2} G_j(x_S^k)^2 > \theta_S(x_S^k).$$

(i)

$$M_{NN} = \nabla_N C_N(F),$$

(ii)

$$M_{pS} = [-\gamma_{pi} : i \in I \quad \frac{\partial C_p(F)}{\partial F_j} : j \in W], \forall p \in N,$$

(iii)

$$M_{Sp} = \left[\begin{array}{l} 0 \in R^{|I_x \cup I_e|} \\ \gamma_{pj} : j \in I_G \cap I \\ \nabla_p C_j(F) : j \in I_G \cap W \end{array} \right] \quad \forall p \in N.$$

Proof
The result follows by considering the function

$$G(F,u) = \left\{ \begin{array}{l} G_F(F,u) = C(F) - \Gamma u \\ G_u(F,u) = \Gamma^T F - D(u) \end{array} \right\}.$$

□

Using (i), (ii), and (iii), in the next result, we present sufficient conditions, easily verified in practice, that will establish when the vector $d_{S'}^T = (d_S^T \;\; d_N^T)$ is a descent direction for $\theta_{S'}$ at the point $x_{S'}^T = (x_S^T \;\; 0^T)$; here $S' = S \cup N$. The computations will entail solving a simple QP of size n_N rather than one of size $n_{S'}$. This is meaningful because generally n_N will be much smaller than $n_{S'}$.

Theorem 3 *Let $S = W \cup I$ with $\theta_S(x_S) = 0$ and $N \subseteq \{p \in \bar{W} : G_p(x_S) < 0\}, N \neq \emptyset$. Also, define $S' = W' \cup I$, where $W' = W \cup N$ and*

$$\begin{array}{l} A = (M_{SN}^T M_{SN} + M_{NN}^T M_{NN}), \\ b = [G_N(x_{S'}) + (M_{NS} d_S)]^T M_{NN}, \\ c = G_N(x_{S'})^T (M_{NS} d_S) + \frac{1}{2}\|M_{NS} d_S\|^2, \end{array} \tag{21}$$

with

$$q(d_N) = \frac{1}{2} d_N^T A d_N + b^T d_N + c. \tag{22}$$

Then, with d_S an optimal search direction, there exists $\bar{d}_N \in R^{n_N}$ such that

$$d_{S'} = \left(\begin{array}{c} d_S \\ \bar{d}_N \end{array} \right)$$

is a descent direction for $\theta_{S'}$ at $x_{S'}$ if and only if $q(d_N^*) < 0$ where d_N^* solves

$$\textit{minimize}_{\ d_N} \quad q(d_N) : d_N \geq 0. \tag{23}$$

Proof

By Lemma 2 (b) [28], a sufficient descent condition for θ is that for some vector $d_{S'}$

$$\phi_{S'}(x_{S'}, d_{S'}) < \phi_{S'}(x_{S'}, 0). \tag{24}$$

We note that with the new index set S', and $x_{S'}^T = (x_S^T \ \ 0)$, we have

$$H_{S'}(x_{S'}) \ = \begin{pmatrix} 0 \\ G_N(x_S) \end{pmatrix}$$

by considering (15), because $H_S(x_S) = 0 \Leftrightarrow \theta_S(x_S) = 0$ and because $G_p(x_{S'}) = G_p(x_S) < 0 = F_p \ \forall p \in N$. Writing out (24) gives the following equivalent form:

$$\tfrac{1}{2}\left\| \begin{pmatrix} 0 \\ G_N(x_{S'}) \end{pmatrix} + \begin{pmatrix} M_{SS} & M_{SN} \\ M_{NS} & M_{NN} \end{pmatrix} \begin{pmatrix} d_S \\ d_p \end{pmatrix} \right\|^2 < \tfrac{1}{2}\left\| \begin{pmatrix} 0 \\ G_p(x_{S'}) \end{pmatrix} \right\|^2.$$

Using the fact that $\theta_S(x_{S'}) = \theta_S(x_S) = 0$, then by Lemma 2 we have $M_{SS}d_S = 0$. After rearranging terms we get

$$q(d_N) = \frac{1}{2} d_N^T A d_N + b^T d_N + c < 0$$

where A, b, and c are defined as in (21). It is not hard to see that $q : R^{n_N} \rightarrow R$ is a convex quadratic function and also that there exists a $\bar{d}_N \geq 0$ such that $q(\bar{d}_N) < 0$ if and only if $q(d_N^*) < 0$ where d_N^* solves (23). $\Box$

The above theorem specifies when a descent direction for $\theta_{S'}$ can be obtained from a previous search direction d_S and a relatively small new vector $\bar{d}_N$.

We note that the quadratic program given in (23) always has a solution. To see this we first write out the associated KKT optimality conditions. These conditions are to find a d_N such that

$$A d_N + b \geq 0 \quad d_N \geq 0 \quad (A d_N + b)^T d_N = 0. \tag{25}$$

These conditions constitute a linear complementarity problem (LCP) with data (A, b). We will make the weak assumption that $\frac{\partial C_p}{\partial F_p} > 0$ for all paths p. Then, since A is the sum of two matrices with nonnegative entries one of which has positive diagonals (i.e., $M_{NN}^T M_{NN}$), by Theorem 3.8.15 in [9], A is a Q-matrix for which the LCP above has a solution for all possible b.

It should be clear that great computational savings may be achieved by computing a search direction in the manner described above, essentially avoiding solving the QP subproblem of size $n_{S'}$.

The complete algorithm for the modified NE/SQP method can now be summarized as follows:

Step 0: Initialization

Select parameters $\sigma, \rho \in (0,1)$, and set $k = 0$. Arbitrarily select $n_W, F_W^0 \in R_+^{n_W}$ and $u^0 \in R^{n_I}$ such that the network is strongly connected given W. Set

$$x_S^0 = \begin{pmatrix} F_W^0 \\ u^0 \end{pmatrix}.$$

Compute W^0 and S^0.

Step 1: Generate Search Direction

Having the sets $W(= W^k), \bar{W}(= \bar{W}^k), S(= S^k)$, and the vector x_S, solve the QP (19) with solution d_S.

Step 2: Update

Case 1: (**No Change in Working Set of Paths, Descent in** θ)

If an optimal direction d_S^k satisfies

$$\phi_S(x_S^k, d_S^k) < \theta_S(x_S^k),$$

then do the following:

- **(a)** perform a standard Armijo-type backtracking to obtain a step length τ_k;
- **(b)** $x_S^{k+1} \leftarrow x_S^k + \tau_k d_S^k$;
- **(c)** $W^{k+1} \leftarrow W^k$, $\bar{W}^{k+1} \leftarrow \bar{W}^k$, $S^{k+1} \leftarrow S^k$;
- **(d)** $k \leftarrow k + 1$;
- **(e)** go to Step 3.

Case 2: (**No Change in Working Set of Paths, No Descent in** θ)
If the optimal direction d_S^k satisfies

$$\phi_S(x_S^k, d_S^k) = \theta_S(x_S^k) > 0,$$

then STOP. If u_i^k is less than or equal to the length of a shortest path for all $i \in I$ and

$$\begin{pmatrix} x_S^k \\ 0 \end{pmatrix}$$

is s-regular, then x solves the overall NCP. Otherwise, terminate the algorithm.

Case 3: (**Reduced NCP Solved and No Shortest Path Violations**)
If the optimal direction d_S^k satisfies

$$\phi_S(x_S^k, d_S^k) = \theta_S(x_S^k) = 0,$$

and u_i^k is less than or equal to a shortest path for all $i \in I$, then STOP. The vector

$$x = \begin{pmatrix} x_S^k \\ 0 \end{pmatrix}$$

is a solution to the overall NCP.

Case 4: (**Reduced NCP Solved but Shortest Path Violations Exist— New Paths Enter**)
If the optimal direction d_S^k satisfies

$$\phi_S(x_S^k, d_S^k) = \theta_S(x_S^k) = 0$$

and $\exists\ N \subseteq \bar{W}$, such that $G_p(x_S^k) < 0$ for all $p \in N$, then do the following for $N \subseteq \{p \in \bar{W} : G_p(x_S^k) < 0\}$ and N nonempty,

(a) let $W' \leftarrow W \cup N$, $\bar{W}' \leftarrow \bar{W} - N$;$S' \leftarrow S \cup N$;

(b)

$$x_{S'} \leftarrow \begin{pmatrix} x_S^k \\ 0 \end{pmatrix};$$

(c) $W \leftarrow W', \bar{W} \leftarrow \bar{W}', S \leftarrow S'$;

(d) go to Step 1.

Step 3: Termination Check
If θ does not satisfy a prescribed termination rule then go to Step 1. Otherwise, if $\exists\ p \in \bar{W}$ with $G_p(x_S^k) < 0$, then go to Case 4 of Step 2. Else, STOP. The vector

$$x = \begin{pmatrix} x_S^k \\ 0 \end{pmatrix}$$

is a solution to the overall NCP.

As can be seen from the algorithm statement, the method solves a sequence of quadratic program subproblems to force descent in θ_S, where S represents the current active indices. As long as strict descent is achieved, the algorithm functions essentially as the NE/SQP method. However, if stalling occurs, that is

$$\phi_S(x_S^k, d_S^k) = \theta_S(x_S^k) = 0,$$

we need to deviate from the NE/SQP approach. Specifically, if stalling is present and the u variables represent shortest paths, we have found a solution as outlined by Lemma 1. Otherwise, if stalling is present, we will have identified at least one violating path p and can let it enter the set W. Of course, when this happens, we could avoid resolving the new QP (Step 1) for the set $S' = S \cup \{p\}$ by generating a descent direction as a solution to a QP of size $|N| << |S'|$ via the results in Theorem 3. Lastly, if case 2's condition is encountered, we know by Proposition 1 in [28] that if $x^T = ((x_S^k)^T \ \ 0^T)$ is s-regular with u's representing shortest paths, this x is a solution to the overall NCP.

We now present the main convergence result concerning this method.

Theorem 4 *Let $\{x^k\}$ be an infinite sequence of iterates generated by the path generation NE/SQP method with $F_p^k = 0\ \forall p \in \bar{W}^k\ \forall k$. Then, if x^* is an accumulation point of $\{x^k\}$, with x^* both b- and s-regular, x^* solves the NCP (13).*

Proof
From the NE/SQP algorithm and the description of the path generation approach above, we see that either we increase the cardinality of W^k at each iteration or we stop because the conditions (i) and (ii) of Lemma 1 have been satisfied. In the latter case, we will have solved NCP(G) so we consider the other possibility. Since the number of paths is finite, we see that eventually, $W^k = P$ and we will be solving the overall NCP for which convergence is guaranteed by the NE/SQP method. $\square$

4 Numerical Examples

To provide concrete evidence of the importance of including nonadditivities and the potential viability of solving nonadditive problems, we now present some illustrative numerical examples. For these examples, unless otherwise specified, we have selected a starting point by solving n_I shortest path problems, one for each O-D pair. This method identifies $W^0 = \{$shortest paths $\}$ so that $|W^0| = n_I$. The initial path flows are all set equal to 125.0 [4], the arc flows f are calculated via $f = \Delta F$, and the starting value for each u_i, $i \in I$ is the value of the shortest path for that O-D pair.

The Network

The network used for these examples is shown in Figure 1. This network has 9 nodes, 28 arcs and a large number of paths connecting any two nodes. The travel time on each link is given by a so-called Bureau of Public Roads (BPR) function with the form

$$t_a(f_a) = A_a + B_a \left(\frac{f_a}{K_a}\right)^4, \tag{26}$$

where A_a denotes the free-flow travel time in minutes on arc a, K_a denotes the practical capacity of arc a in hundreds of vehicles, and B_a is the congestion parameter for arc a. The value of the parameters for each arc are shown in Table 1.

An Example with Separable Demand Functions

There are 72 origin-destination pairs in this example, and a logit function was used to model the O-D demand. This demand function can be thought of as representing the number of people that choose to drive rather than take transit given the cost of the two competing modes. Specifically, the demand functions had the form

$$D_i(u_i) = Q_i \frac{1}{1 + e^{(-\kappa_i + \omega_i u_i)}} \tag{27}$$

where Q_i can be interpreted as the total demand across all modes for O-D i, κ_i can be interpreted as the difference in the attractiveness of the two modes connecting i, and ω_i is a sensitivity parameter for i. The specific

[4] For the nonlinear case with tolls we used a starting point of 0.0 instead.

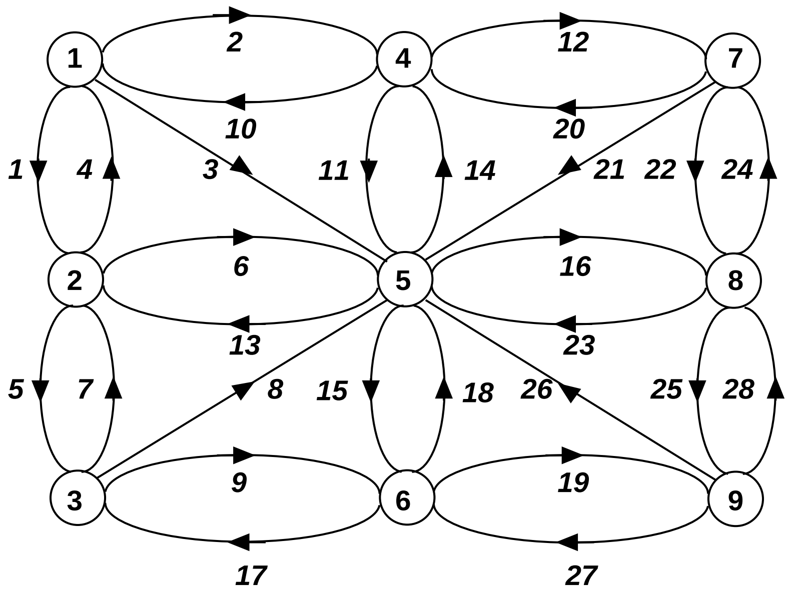

Figure 1: The Network

parameters used are shown in Table 2. In order to provide some intuition into the meaning of these parameters, they were determined in such a way as to represent transit travel times of approximately twice the auto free-flow times and transit fares between \$5.00 and \$15.00 (depending on the O-D pair).

The path cost function used was of the form

$$C_p(F) = \zeta \frac{\sum_a \delta_{ap} t_a(\Delta F)}{30} + \xi \left(\frac{\sum_a \delta_{ap} t_a(\Delta F)}{30} \right)^2 + \Upsilon_p \tag{28}$$

where Υ_p represents the tolls on path p and the parameters were set to $\zeta = 1$ and $\xi = 3$. Again, to provide some intuition, a plot of the cost on a representative path versus the total travel time on that path is shown in Figure 2.

The particular choice of our nonadditive path cost functions allowed us to solve the shortest-path problems in the usual way; see the appendix for details. Other more general choices for nonadditive costs do not necessarily work out as well.

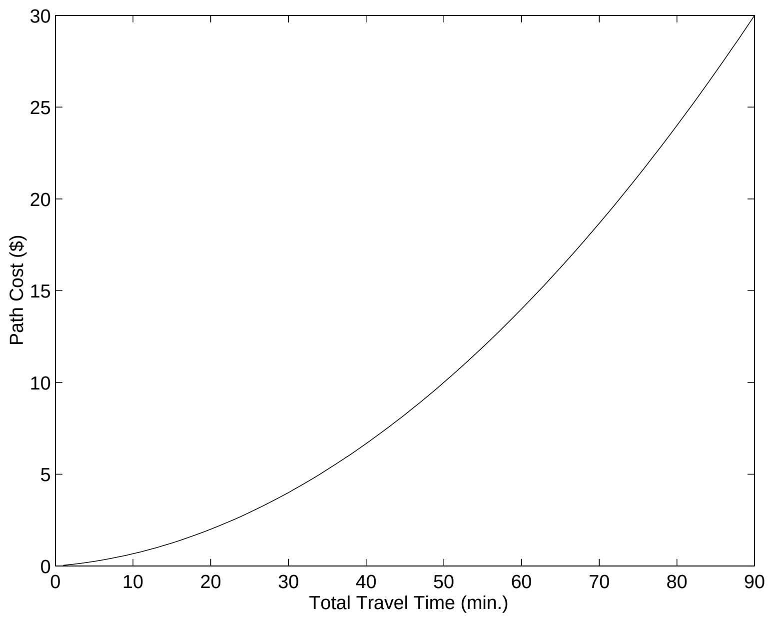

Figure 2: Path Costs

We solved this problem two times, once with no tolls and once with \$3.00 tolls on links 2, 10, 12, and 20. The solution with no tolls is shown in Tables 3, 4, and 5, which contain the equilibrium arc flows/costs, O-D demands/flows, and path flows/costs (on the used paths), respectively. Two factors need to be checked to demonstrate that this is an equilibrium. First, the actual amount of flow betwen every O-D pair must equal the demand for that O-D pair given the path costs; this is easily seen to be true in Table 4. Second, the cost on all used paths connecting a particular O-D pair must be equal and not greater than the cost on any unused paths; the fact that all used paths have equal cost can easily be seen in Table 5. That these costs exceeded the cost on all unused paths is somewhat more difficult to verify, but is in fact the case. A total of 113 paths was generated, though only the 85 paths with positive flow are shown.

The results with tolls are shown in Tables 6, 7, and 8. In this case the total number of paths generated increases to 116, and the total number of used paths falls to 83. Not surprisingly, the flow on the tolled arcs (2, 10, 12, and 20) decreases as a result of the toll, as does the flow on the paths that use these arcs (e.g., 1-4, 1-4-7, 2-1-4, and 2-1-4-7 for arc 2, which has a tail node of 1 and a head node of 4). Of the remaining 24 arcs, the flow decreases on 10 and increases on 14. As can be seen from looking at Tables 4 and 7, these changes in arc flows are primarily a result of changes in path flows, since the total demands remain relatively constant. The only exception is the O-D pairs that are "directly" affected by the tolls (e.g., 1-4, 1-7, and 4-7).

The Impact of Nonadditive Cost Functions

To illustrate the impact of using nonadditive cost functions, we took the above example one step further and solved for an equilibrium both with and without the toll assuming additive costs. Specifically, we identified a linear value of time function (namely, \$5.50 per half hour) that would yield results similar to the nonadditive model when there were no tolls. We then compared the predictions that would be made by the two models in the presence of tolls.

The results for the nonadditive and additive cases when there are no tolls are given in Table 9. As can be seen the solutions are quite similar; the largest difference in arcs flows is only 6%.

In the presence of tolls one would expect the two models to make very different predictions. In particular, for "shorter" paths one would expect the additive/linear model to predict smaller changes due to tolls, and for "longer" trips one would expect the additive/linear model to predict larger changes. The difference is because for the additive model, the toll is a smaller portion of the total path cost for short trips (as compared with the nonadditive/nonlinear model) and a larger portion of the total path cost for long trips (again as opposed to the nonadditive/nonlinear model).

The results of the two models in the presence of tolls are given in Table 10. As expected, the results of the two models are quite different. As shown in Table 11, the "short" paths 1-4, 1-4-7, and 2-1-4 have fairly similar flows in both the additive and nonadditive case, while the "long" path 2-1-4-7 has very different flows in the two cases. (Of course, when making such comparisons it is important to recall that equilibrium path flows are not unique.)

The implications can be quite important from a policy perspective. In particular, a toll designed to reduce congestion would have a much smaller impact than would be predicted by using an additive model with a linear value of time function.

An Asymmetric Example

We now present an example to illustrate that this method can also be applied to problems with asymmetric demand functions (note that the path cost functions above are already asymmetric even though the arc cost functions are separable).

In particular, we assume that while the total demand from each origin is known, the proportion of that demand bound for each destination is unknown. We use an exponential gravity model of the form

$$D_{ij} = 125.0 \frac{e^{-0.1 u_{ij}}}{\sum_{k \in \mathcal{N}} e^{-0.1 u_{ik}}}, \tag{29}$$

where (with a slight abuse of notation) D_{ij} is the demand for O-D pair ij and u_{ij} is the (minimum) O-D travel cost.

The solution for this problem is shown in Tables 12, 13, and 14. This solution is clearly an equilibrium.

Not surprisingly, O-D pairs that are relatively far apart (e.g., 1-9, 1-6) have significantly lower demand than those that are closer together (e.g., 1-2 and 1-4). Also not surprisingly, those paths with high cost (e.g., 1-5-8-9, 3-5-4-7) have relatively low flow whereas those with low cost (e.g., 1-2, 9-6) have relatively high flow.

Perhaps the most interesting result from this example is that while 108 paths were generated, only 76 are used in the equilibrium solution. Indeed, we were consistently able to find solutions in which a single path was used for many O-D pairs and at most two or three were used for all O-D pairs. We found this to be quite surprising because we expected to be able to find equilibria in which only five or more paths were used for most O-D pairs. The result suggests that path enumeration may not be such a tedious task after all.

5 Conclusions and Future Work

We have demonstrated two points in this paper. First, using both qualitative arguments and numerical examples, we have shown that many of today's important transportation policy questions cannot be answered using existing models that employ additive path cost functions. Second, we have shown that algorithms for solving large-scale, elastic, nonadditive traffic equilibrium problems probably can be developed. Several important tasks remain to be completed before the ideas presented here can be applied, however.

First, it will be necessary to formulate and estimate realistic path cost/utility functions. Clearly, a considerable amount of research has already been done on such factors as value of time functions, fuel consumption functions, vehicle operating cost functions, and travel disutility functions. However, the task of incorporating all of this research into a flexible, unified path cost/utility function still remains.

Second, work needs to be done to ensure that either the NE/SQP method or other methods can be used to solve large-scale problems. This will, at a minimum, involve developing methods for efficiently storing and manipulating path variables, calculating shortest paths when the costs are nonadditive. It would also be informative to see a comparison of different algorithms, since their performance on nonadditive problems is likely to be quite different from that on additive problems. For example, we have already learned that diagonalization methods do not work very well on nonadditive problems, apparently because the diagonalized subproblems are very bad approximations of the true problem.

In addition, it is important to consider how the ideas developed here can be applied to other path-based network equilibrium problems. For example, the simultaneous route and departure-time equilibrium problem [16] is most easily formulated using path variables. As another example, researchers have struggled with including entropy terms in TEP [6] because they include path variables. The ideas developed here should both facilitate the solution of those problems and allow them to be expanded to include nonadditive costs.

Appendix

A A Special Case of Nonadditive Path Cost Functions

As mentioned in the introduction, while it is quite common to assume that path costs are an additive function of link costs, there are many situations in which this assumption does not hold. In most such cases it is necessary to use a path formulation of TEP. However, there is one class of nonadditive path costs for which this is not necessary. In particular, suppose that $C_p(F) = g\left[\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right]$ for all $p \in P$, where $g : R_+ \to R_+$ is monotone increasing. Then, it turns out that one can find an equilibrium for the nonadditive problem by solving an appropriate additive problem.

This result may be somewhat surprising because it is not true of complementarity problems in general. In other words, given $G : R^n_+ \to R^n$ and $x \in R^n_+$ such that

$$\begin{aligned} G_i(x) &\geq 0 \quad i = 1, \dots, n \qquad (30) \\ G_i(x)x_i &= 0 \quad i = 1, \dots, n, \qquad (31) \end{aligned}$$

it is not, in general, the case that

$$g[G_i(x)]x_i = 0 \quad i = 1, \dots, n, \qquad (32)$$

even when g is monotone increasing. For example, suppose that $x_j > 0$ for some $j \in \{1, \dots, n\}$. Then it must be the case that $G_j(x) = 0$. However, this does not imply that $g[G_j(x)] \equiv g(0) = 0$.

As it turns out, the TEP is not an ordinary complementarity problem. To see this, first consider the inelastic version of the TEP where $u_i(F) = \min_{p \in P_i} C_p(F)$. For this problem we can demonstrate the following:

Theorem 5 *Suppose $F \in R^{n_P}_+$ satisfies the conditions that*

$$F_p > 0 \Rightarrow \sum_{a \in A} \delta_{ap} c_a(\Delta F) = \min_{p \in P_i} \left\{ \sum_{a \in A} \delta_{ap} c_a(\Delta F) \right\} \qquad (33)$$

for all $i \in I$ and $p \in P_i$. Then, it follows that

$$F_p > 0 \Rightarrow C_p(F) = u_i(F) \qquad (34)$$

for all $i \in I$ and $p \in P_i$.

Proof
We know that

$$\begin{aligned} F_p > 0 \quad &\Rightarrow \quad \sum_{a \in A} \delta_{ap} c_a(\Delta F) = \min_{p \in P_i} \left\{ \sum_{a \in A} \delta_{ap} c_a(\Delta F) \right\} \qquad (35) \\ &\Rightarrow \quad g\left[\sum_{a \in A} \delta_{ap} c_a(\Delta F) \right] = g\left[\min_{p \in P_i} \left\{ \sum_{a \in A} \delta_{ap} c_a(\Delta F) \right\} \right]. \qquad (36) \end{aligned}$$

And, since g is monotone increasing, it follows that

$$F_p > 0 \Rightarrow g\left[\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right] = \min_{p\in P_i}\left\{g\left[\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right]\right\}. \qquad (37)$$

Finally, since $g\left[\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right] \equiv C_p(F)$ and $\min_{p\in P_i}\left\{g\left[\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right]\right\} \equiv \min_{p\in P_i}\{C_p(F)\} \equiv u_i(F)$, the result follows. □

The implication of this result is that we can ignore the transformation g and solve an equlibrium problem with simple additive costs, since the feasible regions for the two problems are identical. That is, if we let $\tilde{C}_p(F) = \sum_{a\in A} \delta_{ap} c_a(\Delta F)$ for all $p \in P$ and find an equilibrium for $\tilde{C}$ it will also be an equilibrium for C.

When demand is elastic (and the inverse demand function exists), a similar result holds. In particular we have the following theorem.

Theorem 6 *Suppose g is invertible and $F \in R_+^{n_P}$ satisfies the conditions that*

$$F_p > 0 \Rightarrow \sum_{a\in A} \delta_{ap} c_a(\Delta F) = \min_{p\in P_i}\left\{\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right\} \quad i \in I, p \in P_i \qquad (38)$$

$$g^{-1}[D_i^{-1}(F)] = \min_{p\in P_i}\left\{\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right\} \quad i \in I. \qquad (39)$$

Then

$$F_p > 0 \Rightarrow C_p(F) = u_i(F) \quad i \in I, p \in P_i, \qquad (40)$$

and

$$D_i^{-1}(F) = u_i(F). \qquad (41)$$

Proof

We know from Theorem 5 that (40) holds whenever (38) holds. Hence, all that remains is to show that (41) follows from (39). To do so, we observe that

$$\begin{aligned} & g^{-1}[D_i^{-1}(F)] && = \min_{p\in P_i}\left\{\textstyle\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right\} \\ & \Rightarrow g\left(g^{-1}[D_i^{-1}(F)]\right) && = g\left(\min_{p\in P_i}\left\{\textstyle\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right\}\right) \\ & \Rightarrow D_i^{-1}(F) && = g\left(\min_{p\in P_i}\left\{\textstyle\sum_{a\in A} \delta_{ap} c_a(\Delta F)\right\}\right). \end{aligned} \qquad (42)$$

Now, since g is monotone increasing, it follows that

$$g\left(\min_{p\in P_i}\left\{\sum_{a\in A}\delta_{ap}c_a(\Delta F)\right\}\right) = \min_{p\in P_i}\left\{g\left(\sum_{a\in A}\delta_{ap}c_a(\Delta F)\right)\right\} \quad (43)$$

$$= \min_{p\in P_i} C_p(F) \quad (44)$$

$$= u_i(F) \quad (45)$$

and hence that $D_i^{-1}(F) = u_i(F)$ $\Box$.

Thus, when the demand is elastic, one can obtain a solution to a TEP with nonadditive path costs by solving a simple additive problem with transformed demand. In other words, if F is an equilibrium for $(\tilde{C}, g^{-1}[D^{-1}])$, it is also an equilibrium for (C, D^{-1}).

Unfortunately, these results do not hold in general. Most important, when $C_p(F) = g_p\left[\sum_{a\in A}\delta_{ap}c_a(\Delta F)\right]$ for all $p \in P$ (i.e., when the transformation is path specific), it does not seem possible to obtain an equilibrium by solving an "appropriate" additive problem. Hence, for such cases it is necessary to use a path formulation [5].

Remark:
It is interesting to note that we can generalize g and still get the same results as shown above. The key idea is that the functions g and min should commute. The result below describes necessary and sufficient conditions on the function g to make this true.

Proposition 1 *Let $g : D \subset R \to R$, and suppose that $x^{min} := min_{x\in D}x$ and $g^* = min_{x\in D}g(x)$ are well defined. Then*

$$g(x^{min}) = g^* \Leftrightarrow x^{min} \in \ argmin\ \{g(x) : x \in D\}.$$

Proof
For all $x \in D$, we have $g(x) \geq min_{x\in D}g(x)$. If $g(x^{min}) = g^* \Rightarrow g(x) \geq g(x^{min})$ as desired. In the other direction, if $g(x^{min}) \leq g(x)$ for all $x \in D$, this gives

$$g(x^{min}) \leq g^* \leq g(x) \ \ \forall x \in D.$$

[5]The nonadditive formulation that we have employed uses $g_p = g$ for all paths p but adds tolls. Selectively including or excluding tolled links (if the number of tolled links is small) allows all shortest-path calculations to be performed in a conventional manner despite the nonadditive costs.

With $x = x^{min}$ this forces the desired result.□

The importance of this result is that functions that are not monotone increasing can also be used, for example, $g(x) = \sin(x)$ with $D = [0, \pi]$ would be valid.

Acknowledgments

The authors thank Professor Philip Gill of the University of California, San Diego, for his help with the software package QPOPT; Professors Jong-Shi Pang and Alan J. Goldman of The Johns Hopkins University; and Professor David Boyce of University of Illinois at Chicago for several useful discussions and references related to this work.

References

[1] H. Z. Aashtiani, *The Multi-Model Assignment Problem*, Ph.D. Thesis, Sloan School of Management, Massachusetts Institute of Technology (1979).

[2] H. Z. Aashtiani and T. Magnanti, "Equilibria on a Congested Transportation Network," *SIAM Journal on Algebraic and Discrete Methods 2* (1981) 213–226.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows Theory, Algorithms, and Applications*, Prentice-Hall, New Jersey (1993).

[4] R. L. Asmuth, *Traffic Network Equilibria*, Ph.D. Thesis, Stanford University (1978).

[5] M. J. Beckmann, C. B. McGuire, and C. B. Winsten, *Studies in the Economics of Transportation*, Yale University Press, New Haven (1956).

[6] M. G. H. Bell, W. H. K. Lam, G. Ploss, and D. Inaudi, "Stochastic User Equilibrium Assignment and Iterative Balancing," *Transportation and Traffic Theory*, in C. F. Daganzo (ed.), Elsevier Science Publishers (1993) 427–439.

[7] D. Bernstein and T. E. Smith, "Network Equilibria with Lower Semicontinuous Costs: With an Application to Congestion Pricing," *Transportation Science* 28 (1984) 221–235.

[8] D. P. Bertsekas and E. M. Gafni, "Projection Methods for Variational Inequalities with Applications to the Traffic Assignment Problem," *Mathematical Programming Study* 17 (1982) 139–159.

[9] R. W. Cottle, J. S. Pang, and R. E. Stone, *The Linear Complementarity Problem*, Academic Press, Inc., Boston (1992).

[10] S. C. Dafermos, "Traffic Equilibrium and Variational Inequalities," *Transportation Science* 14 (1980) 42–54.

[11] S. C. Dafermos, "The General Multimodal Network Equilibrium Problem with Elastic Demands," *Networks* 12 (1982) 57–72.

[12] C. Fisk and D. Boyce, "Alternative Variational Inequality Formulations of the Equilibrium-Travel Choice Problem," *Transportation Science* 17 454–463.

[13] M. Florian, "Nonlinear Cost Network Models in Transportation Analysis," *Mathematical Programming Study* 26 (1986) 167–196.

[14] M. Florian and H. Spiess, "The Convergence of Diagonalization Algorithms for Network Equilibrium Problems," *Transportation Research B* 16B (1982) 447–483.

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

[16] T. L. Friesz, "Transportation Network Equilibrium, Design, and Aggregation: Key Developments and Research Opportunities," *Transportation Research A* 19A (1985) 413–427

[17] S. A. Gabriel and J. S. Pang, "An Inexact NE/SQP Method for Solving the Nonlinear Complementarity Problem," *Computational Optimization and Applications* 1 (1992) 67–91.

[18] P. T. Harker and J. S. Pang, "Finite-Dimensional Variational Inequality and Nonlinear Complementarity Problems: A Survey of Theory, Algorithms and Applications,"*Mathematical Programming* 48 (1990) 161–220.

[19] D. A. Hensher and T. P. Truong, "Valuation of Travel Times Savings," *Journal of Transport Economics and Policy*, (1985) 237–260.

[20] J. L. Horowitz, *Air Quality Analysis for Urban Transportation Planning*, The MIT Press (1982).

[21] T. Larsson, and M. Patricksson, "Simplicial Decomposition with Disaggregated Representation for the Traffic Assignment Problem," *Transportation Science* 26 (1992) 4–17.

[22] S. Lawphonpanich and D. W. Hearn, "Simplicial Decomposition of the Asymmetric Traffic Assignment Problem," *Transportation Research B* 18B (1984) 123–133.

[23] L. J. LeBlanc, E. K. Morlok, and W. Pierskalla, "An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem," *Transportation Research* 9 (1975) 309–318.

[24] T. Leventhal, G. Nemhauser and L. Trotter, Jr., "A Column Generation Algorithm for Optimal Traffic Assignment," *Transportation Science* 7 (1973) 168–176.

[25] P. Marcotte and J. Guélat, "Adaptation of a Modified Newton Method for Solving the Asymmetric Traffic Equilibrium Problem," *Transportation Science* 22 (1988) 112–124.

[26] A. Nagurney, "Comparative Tests of Multimodal Traffic Equilibrium Methods," *Transportation Research B* (1984) 18B 469–485.

[27] S. Nguyen, "A Unified Approach to Equilibrium methods for Traffic Assignment," in M. Florian (ed.) *Traffic Equilibrium Methods*, Springer-Verlag (1974).

[28] J. S. Pang and S. A. Gabriel, "NE/SQP: A Robust Algorithm for the Nonlinear Complementarity Problem," *Mathematical Programming* 60 (1993) 295–337.

[29] J. S. Pang and C. S. Yu, "Linearized Simplicial Decomposition Methods for Computing Traffic Equlibria on Networks," *Networks 14* (1984) 427–438.

[30] M. J. Smith, "The Existence, Uniqueness and Stability of Traffic Equilibria," *Transportation Research B* 13B (1979) 295–304.

[31] M.J. Smith, "The existence and calculation of traffic equilibria," *Transportation Research B* 17B (1983) 291-303.

[32] M. J. Smith, "An Algorithm for Solving Asymmetric Equilibrium Problems with a Continuous Cost-Flow Function," *Transportation Research B* 17B (1983) 365–371.

[33] T. E. Smith and D. Bernstein, "Programmable Network Equilibria," in T.R. Lakshmanan and P. Nijkamp (eds.) *Structure and Change in the Space Economy: A Festschrift in Honor of Martin J. Beckmann*, Springer Verlag (1993) 91–130.

[34] J. G. Wardrop, "Some Theoretical Aspects of Road Traffic Research," *Proc. Inst. Civil Engineers* Part II 1 (1952) 325–378.

Arc (a)	Tail Node	Head Node	A_a (min.)	B_a	K_a (100's)
1	1	2	5.00	10.00	60.00
2	1	4	5.00	10.00	60.00
3	1	5	3.00	15.00	60.00
4	2	1	5.00	10.00	60.00
5	2	3	5.00	10.00	60.00
6	2	5	3.00	15.00	60.00
7	3	2	5.00	10.00	60.00
8	3	5	3.00	15.00	60.00
9	3	6	5.00	10.00	60.00
10	4	1	5.00	10.00	60.00
11	4	5	3.00	15.00	60.00
12	4	7	5.00	10.00	60.00
13	5	2	3.00	15.00	60.00
14	5	4	3.00	15.00	60.00
15	5	6	3.00	15.00	60.00
16	5	8	3.00	15.00	60.00
17	6	3	5.00	10.00	60.00
18	6	5	3.00	15.00	60.00
19	6	9	5.00	10.00	60.00
20	7	4	5.00	10.00	60.00
21	7	5	3.00	15.00	60.00
22	7	8	5.00	10.00	60.00
23	8	5	3.00	15.00	60.00
24	8	7	5.00	10.00	60.00
25	8	9	5.00	10.00	60.00
26	9	5	3.00	15.00	60.00
27	9	6	5.00	10.00	60.00
28	9	8	5.00	10.00	60.00

Table 1: BPR Function Parameters

Origin	Destination	Q_i (100's)	κ_i	ω_i	Origin	Destination	Q_i (100's)	κ_i	ω_i
1	2	25.0	0.15	0.8500	5	6	25.0	0.15	0.7980
1	3	25.0	0.15	1.8000	5	7	25.0	0.15	1.7080
1	4	25.0	0.15	0.8500	5	8	25.0	0.15	0.7980
1	5	25.0	0.15	0.7980	5	9	25.0	0.15	1.7080
1	6	25.0	0.15	1.6320	6	1	25.0	0.15	2.6020
1	7	25.0	0.15	1.8000	6	2	25.0	0.15	1.6320
1	8	25.0	0.15	1.6320	6	3	25.0	0.15	0.8500
1	9	25.0	0.15	2.7180	6	4	25.0	0.15	1.6320
2	1	25.0	0.15	0.8500	6	5	25.0	0.15	0.7980
2	3	25.0	0.15	0.8500	6	7	25.0	0.15	2.6020
2	4	25.0	0.15	1.6320	6	8	25.0	0.15	1.6320
2	5	25.0	0.15	0.7980	6	9	25.0	0.15	0.8500
2	6	25.0	0.15	1.6320	7	1	25.0	0.15	1.8000
2	7	25.0	0.15	2.6020	7	2	25.0	0.15	1.6320
2	8	25.0	0.15	1.6320	7	3	25.0	0.15	2.6020
2	9	25.0	0.15	2.6020	7	4	25.0	0.15	0.8500
3	1	25.0	0.15	1.8000	7	5	25.0	0.15	0.7980
3	2	25.0	0.15	0.8500	7	6	25.0	0.15	1.6320
3	4	25.0	0.15	1.6320	7	8	25.0	0.15	0.8500
3	5	25.0	0.15	0.7980	7	9	25.0	0.15	1.8000
3	6	25.0	0.15	0.8500	8	1	25.0	0.15	2.6020
3	7	25.0	0.15	2.6020	8	2	25.0	0.15	1.6320
3	8	25.0	0.15	1.6320	8	3	25.0	0.15	2.6020
3	9	25.0	0.15	1.8000	8	4	25.0	0.15	1.6320
4	1	25.0	0.15	0.8500	8	5	25.0	0.15	0.7980
4	2	25.0	0.15	1.6320	8	6	25.0	0.15	1.6320
4	3	25.0	0.15	2.6020	8	7	25.0	0.15	0.8500
4	5	25.0	0.15	0.7980	8	9	25.0	0.15	0.8500
4	6	25.0	0.15	1.6320	9	1	25.0	0.15	2.6020
4	7	25.0	0.15	0.8500	9	2	25.0	0.15	1.6320
4	8	25.0	0.15	1.6320	9	3	25.0	0.15	1.8000
4	9	25.0	0.15	2.6020	9	4	25.0	0.15	1.6320
5	1	25.0	0.15	1.7080	9	5	25.0	0.15	0.7980
5	2	25.0	0.15	0.7980	9	6	25.0	0.15	0.8500
5	3	25.0	0.15	1.7080	9	7	25.0	0.15	1.8000
5	4	25.0	0.15	0.7980	9	8	25.0	0.15	0.8500

Table 2: Logit Demand Function Parameters

Arc	Tail Node	Head Node	Travel Time (min.)	Arc Flow (100's)
1	1	2	13.78	58.09
2	1	4	13.78	58.09
3	1	5	11.56	52.15
4	2	1	28.61	74.38
5	2	3	28.65	74.41
6	2	5	4.93	35.92
7	3	2	13.81	58.13
8	3	5	11.35	51.83
9	3	6	13.79	58.10
10	4	1	28.61	74.38
11	4	5	4.93	35.92
12	4	7	28.65	74.41
13	5	2	37.47	73.87
14	5	4	37.47	73.87
15	5	6	37.51	73.90
16	5	8	37.51	73.90
17	6	3	28.60	74.37
18	6	5	4.94	36.00
19	6	9	28.70	74.44
20	7	4	13.81	58.13
21	7	5	11.35	51.83
22	7	8	13.79	58.10
23	8	5	4.94	36.00
24	8	7	28.60	74.37
25	8	9	28.70	74.44
26	9	5	11.36	51.85
27	9	6	13.76	58.05
28	9	8	13.76	58.05

Table 3: Arc Flows with No Tolls

O	D	Flow (100's)	Demand (100's)	O	D	Flow (100's)	Demand (100's)
1	2	16.63	16.63	5	6	11.92	11.92
1	3	16.64	16.64	5	7	7.71	7.71
1	4	16.63	16.63	5	8	11.92	11.92
1	5	16.56	16.56	5	9	7.67	7.67
1	6	13.64	13.64	6	1	10.79	10.79
1	7	16.64	16.64	6	2	15.68	15.68
1	8	13.64	13.64	6	3	14.35	14.35
1	9	8.32	8.32	6	4	15.68	15.68
2	1	14.35	14.35	6	5	17.04	17.04
2	3	14.34	14.34	6	7	10.77	10.77
2	4	15.69	15.69	6	8	15.67	15.67
2	5	17.04	17.04	6	9	14.33	14.33
2	6	15.67	15.67	7	1	16.64	16.64
2	7	10.78	10.78	7	2	13.72	13.72
2	8	15.67	15.67	7	3	7.82	7.82
2	9	10.74	10.74	7	4	16.62	16.62
3	1	16.64	16.64	7	5	16.58	16.58
3	2	16.62	16.62	7	6	13.71	13.71
3	4	13.72	13.72	7	8	16.63	16.63
3	5	16.58	16.58	7	9	16.62	16.62
3	6	16.63	16.63	8	1	10.79	10.79
3	7	7.82	7.82	8	2	15.68	15.68
3	8	13.71	13.71	8	3	10.77	10.77
3	9	16.62	16.62	8	4	15.68	15.68
4	1	14.35	14.35	8	5	17.04	17.04
4	2	15.69	15.69	8	6	15.67	15.67
4	3	10.78	10.78	8	7	14.35	14.35
4	5	17.04	17.04	8	9	14.33	14.33
4	6	15.67	15.67	9	1	7.84	7.84
4	7	14.34	14.34	9	2	13.72	13.72
4	8	15.67	15.67	9	3	16.65	16.65
4	9	10.74	10.74	9	4	13.72	13.72
5	1	7.72	7.72	9	5	16.57	16.57
5	2	11.93	11.93	9	6	16.63	16.63
5	3	7.71	7.71	9	7	16.65	16.65
5	4	11.93	11.93	9	8	16.63	16.63

Table 4: O-D Demands with No Tolls

Path	Flow (100's)	Cost ($)	Path	Flow (100's)	Cost ($)
1-2	16.63	1.09	5-6	11.92	5.94
1-2-3	16.64	7.42	5-4-7	1.26	16.78
1-4	16.63	1.09	5-8-7	6.45	16.78
1-5	16.56	0.83	5-8	11.92	5.94
1-5-6	13.64	9.66	5-6-9	7.67	16.82
1-4-7	16.64	7.42	6-3-2-1	10.73	19.18
1-5-8	13.64	9.66	6-5-4-1	0.06	19.18
1-5-6-9	0.33	22.76	6-5-2	1.55	7.41
1-5-8-9	8.00	22.76	6-3-2	14.14	7.41
2-1	14.35	3.68	6-3	14.35	3.68
2-3	14.34	3.69	6-5-4	15.68	7.41
2-5-4	1.62	7.40	6-5	17.04	0.25
2-1-4	14.07	7.40	6-5-4-7	1.68	19.20
2-5	17.04	0.25	6-9-8-7	9.09	19.20
2-5-6	1.56	7.42	6-9-8	15.67	7.42
2-3-6	14.12	7.42	6-9	14.33	3.70
2-5-4-7	0.03	19.19	7-4-1	16.64	7.41
2-1-4-7	10.75	19.19	7-5-2	13.72	9.57
2-5-8	15.67	7.42	7-5-2-3	7.82	22.59
2-3-6-9	10.74	19.24	7-4	16.62	1.10
3-2-1	16.64	7.41	7-5	16.58	0.81
3-2	16.62	1.10	7-5-6	13.71	9.59
3-5-4	13.72	9.57	7-8	16.63	1.09
3-5	16.58	0.81	7-8-9	16.62	7.43
3-6	16.63	1.09	8-7-4-1	10.79	19.18
3-5-4-7	4.87	22.59	8-5-2	15.68	7.41
3-5-8-7	2.96	22.59	8-5-2-3	1.62	19.20
3-5-8	13.71	9.59	8-9-6-3	9.15	19.20
3-6-9	16.62	7.43	8-5-4	1.60	7.41
4-1	14.35	3.68	8-7-4	14.08	7.41
4-1-2	15.69	7.40	8-5	17.04	0.25
4-1-2-3	9.14	19.19	8-5-6	0.06	7.42
4-5-6-3	1.64	19.19	8-9-6	15.61	7.42
4-5	17.04	0.25	8-7	14.35	3.68
4-5-6	15.67	7.42	8-9	14.33	3.70
4-7	14.34	3.69	9-5-2-1	7.84	22.57
4-5-8	1.56	7.42	9-5-2	13.72	9.58
4-7-8	14.12	7.42	9-6-3	16.65	7.39
4-7-8-9	10.74	19.24	9-5-4	13.72	9.58
5-4-1	7.72	16.76	9-5	16.57	0.81
5-2	11.93	5.93	9-6	16.63	1.09
5-6-3	7.71	16.78	9-8-7	16.65	7.39
5-4	11.93	5.93	9-8	16.63	1.09

Table 5: Path Flows with No Tolls

Arc	Tail Node	Head Node	Travel Time (min.)	Arc Flow (100's)
1	1	2	13.61	57.79
2	1	4	8.86	47.30
3	1	5	11.16	51.53
4	2	1	27.08	73.14
5	2	3	28.35	74.17
6	2	5	5.89	39.75
7	3	2	14.07	58.56
8	3	5	10.89	51.10
9	3	6	14.18	58.73
10	4	1	20.47	66.92
11	4	5	5.31	37.60
12	4	7	20.50	66.95
13	5	2	38.95	74.65
14	5	4	39.87	75.13
15	5	6	37.73	74.01
16	5	8	38.99	74.67
17	6	3	29.57	75.12
18	6	5	4.69	34.77
19	6	9	29.68	75.20
20	7	4	8.87	47.32
21	7	5	10.96	51.21
22	7	8	13.62	57.81
23	8	5	5.89	39.74
24	8	7	27.08	73.14
25	8	9	28.42	74.22
26	9	5	10.90	51.11
27	9	6	14.19	58.74
28	9	8	14.00	58.44

Table 6: Arc Flows with Tolls

O	D	Flow (100's)	Demand (100's)	O	D	Flow (100's)	Demand (100's)
1	2	16.65	16.65	5	6	11.86	11.86
1	3	16.76	16.76	5	7	7.73	7.73
1	4	14.46	14.46	5	8	11.52	11.52
1	5	16.59	16.59	5	9	7.22	7.22
1	6	13.70	13.70	6	1	10.93	10.93
1	7	14.50	14.50	6	2	15.33	15.33
1	8	13.27	13.27	6	3	14.15	14.15
1	9	7.96	7.96	6	4	15.06	15.06
2	1	14.65	14.65	6	5	17.05	17.05
2	3	14.40	14.40	6	7	10.92	10.92
2	4	14.71	14.71	6	8	15.32	15.32
2	5	16.99	16.99	6	9	14.12	14.12
2	6	15.65	15.65	7	1	14.50	14.50
2	7	11.42	11.42	7	2	13.35	13.35
2	8	14.97	14.97	7	3	7.48	7.48
2	9	10.24	10.24	7	4	14.46	14.46
3	1	16.97	16.97	7	5	16.61	16.61
3	2	16.60	16.60	7	6	13.77	13.77
3	4	13.06	13.06	7	8	16.64	16.64
3	5	16.62	16.62	7	9	16.74	16.74
3	6	16.59	16.59	8	1	11.43	11.43
3	7	8.05	8.05	8	2	14.98	14.98
3	8	13.37	13.37	8	3	10.25	10.25
3	9	16.25	16.25	8	4	14.70	14.70
4	1	13.05	13.05	8	5	16.99	16.99
4	2	15.15	15.15	8	6	15.63	15.63
4	3	11.82	11.82	8	7	14.65	14.65
4	5	17.02	17.02	8	9	14.39	14.39
4	6	15.51	15.51	9	1	8.06	8.06
4	7	13.05	13.05	9	2	13.38	13.38
4	8	15.14	15.14	9	3	16.28	16.28
4	9	11.77	11.77	9	4	13.06	13.06
5	1	7.74	7.74	9	5	16.62	16.62
5	2	11.53	11.53	9	6	16.58	16.58
5	3	7.26	7.26	9	7	16.98	16.98
5	4	11.28	11.28	9	8	16.60	16.60

Table 7: O-D Flows with Tolls

Path	Flow (100's)	Cost ($)	Path	Flow (100's)	Cost ($)
1-2	16.65	1.07	5-6	11.86	6.00
1-2-3	16.76	7.27	5-8-7	7.73	16.75
1-4	14.46	3.56	5-8	11.52	6.37
1-5	16.59	0.79	5-6-9	7.22	17.39
1-5-6	13.70	9.60	6-5-2-1	0.93	19.03
1-4-7	14.50	9.85	6-3-2-1	10.00	19.03
1-5-8	13.27	10.05	6-5-2	0.34	7.80
1-5-6-9	2.51	23.19	6-3-2	15.00	7.80
1-5-8-9	5.45	23.19	6-3	14.15	3.90
2-1	14.65	3.35	6-5-4	15.06	8.11
2-3	14.40	3.62	6-5	17.05	0.23
2-5-4	7.79	8.51	6-5-8-7	1.38	19.05
2-1-4	6.92	8.50	6-9-8-7	9.53	19.05
2-5	16.99	0.31	6-9-8	15.32	7.82
2-3-6	15.65	7.45	6-9	14.12	3.93
2-1-4-7	11.42	18.50	7-4-1	14.50	9.85
2-5-8	14.97	8.21	7-5-2	13.35	9.97
2-3-6-9	10.24	19.79	7-5-2-3	5.29	23.02
3-2-1	16.97	7.02	7-5-6-3	2.18	23.02
3-2	16.60	1.13	7-4	14.46	3.56
3-5-4	13.06	10.28	7-5	16.61	0.77
3-5	16.62	0.76	7-5-6	13.77	9.53
3-6	16.59	1.14	7-8	16.64	1.07
3-5-8-7	4.50	22.31	7-8-9	16.74	7.29
3-5-4-7	3.56	22.31	8-7-4-1	11.43	18.49
3-5-8	13.37	9.96	8-5-2	14.98	8.20
3-6-9	16.25	7.87	8-9-6-3	10.25	19.77
4-1	13.05	5.08	8-5-4	7.77	8.51
4-5-2	2.59	8.01	8-7-4	6.93	8.51
4-1-2	12.56	8.01	8-5	16.99	0.31
4-1-2-3	11.82	18.07	8-9-6	15.63	7.47
4-5	17.02	0.27	8-7	14.65	3.35
4-5-6	15.51	7.61	8-9	14.39	3.64
4-7	13.05	5.08	9-5-2-1	4.52	22.29
4-5-8	2.49	8.02	9-5-4-1	3.55	22.29
4-7-8	12.66	8.02	9-5-2	13.38	9.94
4-7-8-9	11.77	18.12	9-6-3	16.28	7.84
5-2-1	7.74	16.74	9-5-4	13.06	10.29
5-2	11.53	6.36	9-5	16.62	0.76
5-6-3	7.26	17.34	9-6	16.58	1.14
5-4	11.28	6.63	9-8-7	16.98	6.99
			9-8	16.60	1.12

Table 8: Path Flows with Tolls

	Linear Value of Time		Nonlinear Value of Time		
Arc	t_a (min.)	f_a (100's)	t_a (min.)	f_a (100's)	Diff. in Flow (%)
1	13.30	57.28	13.78	58.09	-0.01
2	13.30	57.28	13.78	58.09	-0.01
3	13.20	54.48	11.56	52.15	0.04
4	35.31	79.17	28.61	74.38	0.06
5	35.40	79.23	28.65	74.41	0.06
6	4.75	35.06	4.93	35.92	-0.02
7	13.34	57.35	13.81	58.13	-0.01
8	12.87	54.04	11.35	51.83	0.04
9	13.32	57.31	13.79	58.10	-0.01
10	35.31	79.17	28.61	74.38	0.06
11	4.75	35.06	4.93	35.92	-0.02
12	35.40	79.23	28.65	74.41	0.06
13	43.86	77.08	37.47	73.87	0.04
14	43.86	77.08	37.47	73.87	0.04
15	43.97	77.14	37.51	73.90	0.04
16	43.97	77.14	37.51	73.90	0.04
17	35.29	79.15	28.60	74.37	0.06
18	4.77	35.17	4.94	36.00	-0.02
19	35.47	79.27	28.70	74.44	0.06
20	13.34	57.34	13.81	58.13	-0.01
21	12.87	54.04	11.35	51.83	0.04
22	13.32	57.31	13.79	58.10	-0.01
23	4.77	35.18	4.94	36.00	-0.02
24	35.29	79.15	28.60	74.37	0.06
25	35.47	79.27	28.70	74.44	0.06
26	12.89	54.06	11.36	51.85	0.04
27	13.27	57.22	13.76	58.05	-0.01
28	13.27	57.22	13.76	58.05	-0.01

Table 9: Comparison of Arc Flows for the Additive and Nonadditive Cases in the Absence of Tolls

	Linear Value of Time		Nonlinear Value of Time		
Arc	t_a (min.)	f_a (100's)	t_a (min.)	f_a (100's)	Diff. in Flow (%)
1	14.55	59.32	13.61	57.79	0.03
2	6.42	36.80	8.86	47.30	-0.29
3	12.39	53.37	11.16	51.53	0.03
4	31.71	76.70	27.08	73.14	0.05
5	33.80	78.16	28.35	74.17	0.05
6	8.38	46.43	5.89	39.75	0.14
7	14.30	58.93	14.07	58.56	0.01
8	12.11	52.97	10.89	51.10	0.04
9	13.97	58.39	14.18	58.73	-0.01
10	18.44	64.60	20.47	66.92	-0.04
11	4.06	30.96	5.31	37.60	-0.21
12	18.47	64.64	20.50	66.95	-0.04
13	49.20	79.49	38.95	74.65	0.06
14	46.11	78.12	39.87	75.13	0.04
15	44.52	77.39	37.73	74.01	0.04
16	49.28	79.52	38.99	74.67	0.06
17	38.47	81.15	29.57	75.12	0.07
18	4.03	30.71	4.69	34.77	-0.13
19	38.74	81.32	29.68	75.20	0.08
20	6.43	36.89	8.87	47.32	-0.28
21	12.08	52.92	10.96	51.21	0.03
22	14.52	59.26	13.62	57.81	0.02
23	8.35	46.38	5.89	39.74	0.14
24	31.67	76.67	27.08	73.14	0.05
25	33.99	78.29	28.42	74.22	0.05
26	12.12	52.98	10.90	51.11	0.04
27	13.98	58.41	14.19	58.74	-0.01
28	14.27	58.87	14.00	58.44	0.01

Table 10: Comparison of Arc Flows for the Additive and Nonadditive Cases in the Presence of Tolls

	Linear Value of Time		Nonlinear Value of Time	
Path	Path Flow (100's)	Path Cost ($)	Path Flow (100's)	Path Cost ($)
1-4	13.89	4.18	14.46	3.56
1-4-7	13.84	10.56	14.50	9.85
2-1-4	9.06	9.99	6.92	8.50
2-1-4-7	0.00	16.38	11.42	18.50

Table 11: Comparison of Some Path Flows for the Additive and Nonadditive Cases in the Presence of Tolls

Arc	Tail Node	Head Node	Travel Time (min.)	Arc Flow (100's)
1	1	2	14.06	58.54
2	1	4	14.06	58.54
3	1	5	10.69	50.77
4	2	1	23.49	69.96
5	2	3	23.49	69.97
6	2	5	6.04	40.24
7	3	2	14.06	58.54
8	3	5	10.69	50.77
9	3	6	14.06	58.54
10	4	1	23.49	69.97
11	4	5	6.04	40.24
12	4	7	23.49	69.96
13	5	2	34.19	72.05
14	5	4	34.19	72.05
15	5	6	34.19	72.05
16	5	8	34.19	72.05
17	6	3	23.49	69.96
18	6	5	6.04	40.25
19	6	9	23.49	69.96
20	7	4	14.06	58.54
21	7	5	10.69	50.78
22	7	8	14.06	58.54
23	8	5	6.04	40.25
24	8	7	23.49	69.96
25	8	9	23.49	69.96
26	9	5	10.69	50.77
27	9	6	14.06	58.54
28	9	8	14.06	58.54

Table 12: Arc Flows for the Gravity Model

O	D	Flow (100's)	Demand (100's)	O	D	Flow (100's)	Demand (100's)
1	2	22.95	22.95	5	6	21.55	21.55
1	3	14.17	14.17	5	7	9.70	9.70
1	4	22.95	22.95	5	8	21.55	21.54
1	5	23.86	23.86	5	9	9.70	9.71
1	6	11.30	11.30	6	1	6.42	6.42
1	7	14.17	14.17	6	2	15.01	15.01
1	8	11.30	11.30	6	3	20.95	20.95
1	9	4.31	4.31	6	4	13.89	13.89
2	1	20.95	20.95	6	5	26.36	26.36
2	3	20.95	20.95	6	7	6.42	6.42
2	4	15.01	15.01	6	8	15.01	15.01
2	5	26.36	26.36	6	9	20.95	20.95
2	6	15.01	15.01	7	1	14.17	14.17
2	7	6.42	6.42	7	2	11.30	11.30
2	8	13.89	13.89	7	3	4.31	4.31
2	9	6.42	6.42	7	4	22.95	22.95
3	1	14.17	14.17	7	5	23.86	23.86
3	2	22.95	22.95	7	6	11.30	11.30
3	4	11.30	11.30	7	8	22.94	22.95
3	5	23.86	23.86	7	9	14.17	14.17
3	6	22.95	22.95	8	1	6.42	6.42
3	7	4.31	4.31	8	2	13.89	13.88
3	8	11.30	11.30	8	3	6.42	6.42
3	9	14.17	14.17	8	4	15.01	15.01
4	1	20.95	20.95	8	5	26.36	26.36
4	2	15.01	15.01	8	6	15.01	15.01
4	3	6.42	6.42	8	7	20.95	20.95
4	5	26.36	26.36	8	9	20.95	20.95
4	6	13.89	13.89	9	1	4.31	4.31
4	7	20.95	20.95	9	2	11.30	11.30
4	8	15.01	15.01	9	3	14.17	14.17
4	9	6.42	6.42	9	4	11.30	11.30
5	1	9.70	9.70	9	5	23.86	23.86
5	2	21.54	21.54	9	6	22.95	22.95
5	3	9.70	9.70	9	7	14.17	14.17
5	4	21.54	21.54	9	8	22.95	22.95

Table 13: O-D Flows for the Gravity Model

Path	Flow (100's)	Cost ($)	Path	Flow (100's)	Cost ($)
1-2	22.95	1.13	5-6	21.55	5.04
1-2-3	14.17	5.95	5-4-7	2.70	13.01
1-4	22.95	1.13	5-8-7	7.01	13.01
1-5	23.86	0.74	5-8	21.55	5.04
1-5-6	11.30	8.21	5-6-9	7.01	13.01
1-4-7	14.17	5.95	5-8-9	2.70	13.01
1-5-8	11.30	8.21	6-3-2-1	6.42	14.45
1-5-8-9	4.31	17.86	6-3-2	15.01	5.95
2-1	20.95	2.62	6-3	20.95	2.62
2-3	20.95	2.62	6-5-4	13.89	6.73
2-1-4	15.01	5.95	6-5	26.36	0.32
2-5	26.36	0.32	6-9-8-7	6.42	14.45
2-3-6	15.01	5.95	6-9-8	15.01	5.95
2-1-4-7	6.42	14.45	6-9	20.95	2.62
2-5-8	13.89	6.73	7-4-1	14.17	5.95
2-3-6-9	6.42	14.45	7-5-2	11.30	8.21
3-2-1	14.17	5.95	7-5-2-3	4.31	17.86
3-2	22.95	1.13	7-4	22.95	1.13
3-5-4	11.30	8.21	7-5	23.86	0.74
3-5	23.86	0.74	7-5-6	11.30	8.21
3-6	22.95	1.13	7-8	22.94	1.13
3-5-4-7	4.31	17.86	7-8-9	14.17	5.95
3-5-8	11.30	8.21	8-7-4-1	6.42	14.45
3-6-9	14.17	5.95	8-5-2	13.89	6.73
4-1	20.95	2.62	8-9-6-3	6.42	14.45
4-1-2	15.01	5.95	8-7-4	15.01	5.95
4-1-2-3	6.42	14.46	8-5	26.36	0.32
4-5	26.36	0.32	8-9-6	15.01	5.95
4-5-6	13.89	6.73	8-7	20.95	2.62
4-7	20.95	2.62	8-9	20.95	2.62
4-7-8	15.01	5.95	9-5-2-1	4.31	17.86
4-7-8-9	6.42	14.45	9-5-2	11.30	8.21
5-2-1	2.70	13.01	9-6-3	14.17	5.95
5-4-1	7.01	13.01	9-5-4	11.30	8.21
5-2	21.54	5.04	9-5	23.86	0.74
5-2-3	2.70	13.01	9-6	22.95	1.13
5-6-3	7.01	13.01	9-8-7	14.17	5.95
5-4	21.54	5.04	9-8	22.95	1.13

Table 14: Path Flows for the Gravity Model