Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia - Social and Behavioral Sciences 80 (2013) 41 – 60
Procedia Social and Behavioral Sciences

Computational precision of traffic equilibria sensitivities in automatic network design and road pricing

Hillel Bar-Gera[a], Fredrik Hellman[b], Michael Patriksson[c]

[a]*Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be´er-Sheva 84105 Israel; e-mail: bargera@bgu.ac.il*
[b]*Department of Information Technology, Uppsala University, Box 337, SE-751 05, Uppsala, Sweden; e-mail: fredrik.hellman@it.uu.se*
[c]*Department of Mathematical Sciences, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden; e-mail: mipat@chalmers.se*

Abstract

Recent studies demonstrate the importance of computational precision of user equilibrium traffic assignment solutions for scenario comparisons. When traffic assignment is hierarchically embedded in a model for network design and/or road pricing, not only the precision of the solution itself becomes more important, but also the precision of its derivatives with respect to the design parameters should be considered.

The main purpose of this paper is to present a method for precise computations of equilibrium derivatives. Numerical experiments on a small network are used for two evaluations: 1. precision of computed equilibrium derivatives; and 2. the impact of precise derivatives on the quality of network design solutions.

Selection and peer-review under responsibility of Delft University of Technology

Keywords: Traffic equilibria, sensitivities, computational precision
2000 MSC: 90C30, 90C35, 90C31, 49K40, 49J40

1. Introduction and motivation

The purpose of traffic network equilibrium (or, traffic assignment) models is to obtain a prediction of the distribution of flows on links and routes in a traffic network. During later years its use has been extended to also be part of hierarchical (that is, bilevel optimization) models (e.g., Marcotte (1986); Suh & Kim (1992); Suwansirikul *et al.* (1987); Friesz *et al.* (1990); Josefsson & Patriksson (2007)). In such a model, design and/or pricing parameters are chosen and optimized wrt. a desired state of the network as reflected through the objective function; the traffic equilibrium model constitutes part of the constraints of this hierarchical model, evaluating the effect that the design, or pricing scheme, has on the users of the traffic network and hence on congestion levels, environmental effects, and so forth.

Indeed, practical network design is often influenced by various considerations which cannot be represented by a mathematical model. However, automatic network design by hierarchical models can serve practical purposes as well. Following the introduction in 2006 of congestion charges in Stockholm, the PhD thesis of Kristoffersson (2011) includes experiments conducted utilizing bilevel optimization methods (Ekström *et al.* (2009)) on both dynamic and static traffic models of Stockholm, in order to study welfare

Selection and peer-review under responsibility of Delft University of Technology
doi:10.1016/j.sbspro.2013.05.005

and equity effects of the scheme. Further, Vitins *et al.* (2012) demonstrate how automatic network design tools can help evaluating rules and guidelines that are often used in the practical process of network design.

The focus of this paper is on computational *precision*, i.e., how close computed values are to numerically satisfying the mathematical conditions stated. Differences between computed values and reality, denoted by the term *accuracy*, are another concern that is not in the focus of this paper. Nevertheless, it should be acknowledged at the outset that the accuracy of transport models is rather limited, due to uncertainties about the inputs, and due to imperfections in the assumptions. Therefore, one may ask about the importance of precise computations when the model is not necessarily accurate. Our perspective is that while model accuracy is an important concern, plausible model behavior is also highly desirable. Hence, computational precision should be sufficient to ensure plausibility, both at the level of the traffic network equilibrium model, as well as at the level of the hierarchical model, particularly in the context of automatic network design.

Precision of equilibrium solutions has been a focus in transportation research for many years. Presently, few algorithms exist that may reach very precise equilibrium solutions (i.e., approaching machine limits) very fast for large-scale instances; among them we currently count Algorithm B (Dial (2006)), the cutting plane algorithm in Babonneau *et al.* (2006), and TAPAS (Bar-Gera (2010)). These algorithms enable the examination of the precision needed in equilibrium computations, in order to plausibly forecast the impacts associated with road and public transport projects, as demonstrated by recent detailed studies of large-scale traffic equilibrium models (e.g., Boyce *et al.* (2004) and Slavin *et al.* (2009)).

The need for computational precision solutions becomes even more pronounced when traffic models are part of a hierarchical model, and for this there are at least two reasons. First, the precision of the user equilibrium calculation has an immediate effect on the precision of the value of the design/pricing objective function, which is important for the correct identification of promising designs/pricing schemes and the ranking among them, as well as for the identification of potential updates of the design itself within the optimization method chosen. Second, several descent-based algorithms for the hierarchical model (e.g., Josefsson & Patriksson (2007)) rely on computations of directional derivatives of the design/pricing objective function; this directional derivative is calculated through a sensitivity analysis of the equilibrium flows with respect to changes in the costs and demands which in fact amounts to solving a special, reduced type of traffic equilibrium model with affine link costs and demands (Qiu & Magnanti (1989); Patriksson & Rockafellar (2003); Patriksson (2004)).

To the best of the authors' knowledge, previous studies of hierarchical models (e.g., Lim (2002); Josefsson & Patriksson (2007); Chiou (2008)), devoted limited attention to these issues of precision. In some cases less precise equilibrium solvers were used, and in some cases details regarding the methodology and its precision were not disclosed at all.

The purpose of this paper is to explore the merits of precise equilibrium solutions and precise derivative computations in the context of hierarchical models for network design and/or toll optimization.

We present a method to compute directional derivatives of equilibria, combining the sensitivity analysis approach of Patriksson (2004) with the outputs of TAPAS (Bar-Gera (2010)). We show how TAPAS, in addition to efficiently finding very precise equilibrium solutions, also produces outputs that can be very effectively utilized to compute directional derivatives of equilibrium solutions. These derivatives can be used in an implicit programming setting to solve the design problem, either with a general-purpose nonlinear programming solver, or a specially designed one.

The numerical evaluation in this paper consists of two components. The first component is a numerical evaluation of the precision of computed derivatives. Performing such numerical evaluation is not trivial. To keep the paper self-contained, we describe the method used here for this purpose, which might be relevant for numerical evaluation of derivative precision in general. The second component is an evaluation of the impact of precise derivative computations on the quality of solutions to one example of a network design problem.

Our development here concerns the simplest case of a fixed demand traffic network equilibrium model with one user group. While it is not realistic to assume that network designs and/or pricing schemes do not affect demands, the assumption makes illustrations and discussions much simpler. Extensions of our proposed methodology to elastic demands, multiple modes, etcetera, are straightforward to include in principle, but will require further developments of the TAPAS code in order to yield efficient solutions.

In the next section, we present the problem framework and establish the notation used throughout. In Section 3 we describe the principles for computing sensitivities of equilibria. Section 4 gives a brief description of the traffic assignment method TAPAS, and shows how it can be used also for computing sensitivities. The remainder of the paper is devoted to numerically illustrating the importance of calculating equilibria and their sensitivities as exactly as possible. A general framework for derivative precision evaluation is presented in Section 5. This framework is used in Section 6.1 to evaluate the precision of gradients generated by the proposed method. In Section 6.2 we compare the proposed method with other methods on the capacity expansion problem, revisiting two instances of the problem for the Sioux Falls network. We conclude in Section 7 with remarks on future research possibilities.

2. The equilibrium, network design and pricing problem

2.1. The traffic equilibrium problem

The *equilibrium condition* refers to the concept initially introduced by the statistician J. G. Wardrop of the British Road Research Laboratory. Ever since his seminal paper (Wardrop (1952)), the user equilibrium conditions are known as *Wardrop's first principle*. It states that "The journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route." To describe these conditions in mathematical notation, we consider a given origin–destination (OD) pair (p, q) of nodes, a (finite) set $\mathcal{R}_{pq}$ of simple (cycle-free) routes starting at node p and ending at node q, and consider further h_r to be the volume of traffic on route $r \in \mathcal{R}_{pq}$. If we also let c_r be the travel cost on this route as experienced by an individual user, given the current volume h of traffic, and fix the travel costs to these values, then the user equilibrium conditions can be written as follows:

$$h_r > 0 \implies c_r = \pi_{pq}, \qquad r \in \mathcal{R}_{pq}, \qquad (p, q) \in \mathcal{C}, \tag{1a}$$

$$h_r = 0 \implies c_r \geq \pi_{pq}, \qquad r \in \mathcal{R}_{pq}, \qquad (p, q) \in \mathcal{C}, \tag{1b}$$

where π_{pq} denotes the minimal (that is, equilibrium) route cost for OD pair (p, q). In order to actually find a vector h of traffic volumes satisfying the user equilibrium conditions (1), we restate the latter as a variational inequality, as follows.

Suppose that the travel cost on a route is the sum of the travel costs on each of the links composing the route; this link cost is denoted $t_l(v)$, where $l \in \mathcal{L}$ denotes an individual link in the link set $\mathcal{L}$, and $v \in \mathcal{R}^{|\mathcal{L}|}$ is the vector of link flows. The relationships between the flow vectors h and v on the one hand, and the cost vectors c and t on the other hand, are provided by the following equations:

$$v = \Lambda h; \qquad c(h) = \Lambda^{\mathrm{T}} t(v),$$

where $\Lambda \in \mathbb{R}^{|\mathcal{R}| \times |\mathcal{L}|}$ is the link–route incidence matrix which identifies the links composing each route. In other words, the flow on a given link is the sum of flows on the routes that pass it, and the travel cost of a route is the sum of the costs of the links composing the route.

We introduce also the OD-pair–route incidence matrix, $\Gamma \in \mathbb{R}^{|\mathcal{R}| \times |\mathcal{C}|}$, to identify the set of routes that connect a given pair of origin and destination. Given this notation, it is possible to re-state the Wardrop conditions (1), together with the condition for OD-flows being consistent, as follows:

The route-based variational inequality formulation is to find a vector $h^* \in H$, with

$$H := \{ h \in \mathbb{R}_+^{|\mathcal{R}|} \mid \Gamma^{\mathrm{T}} h = d \}$$

such that $c(h^*)^{\mathrm{T}}(h - h^*) \geq 0$ for all $h \in H$. The link-based variational inequality is to find a vector $v^* \in V$ with

$$V := \left\{ v \in \mathbb{R}^{|\mathcal{L}|} \,\middle|\, \exists h \geq 0 : \Gamma^{\mathrm{T}} h = d, \quad \Lambda h = v \right\}, \tag{2}$$

such that $t(v^*)^{\mathrm{T}}(v - v^*) \geq 0$ for all $v \in V$.

To motivate these statements notice that, for the route-based variational inequality formulation, the variational inequality states that the flow vector h^* solves the problem of minimizing the function $c(h^*)^{\mathrm{T}} h$ over the feasible flow region, $h \in H$; this is exactly the linear optimization formulation of the problem to find the shortest routes in the associated graph, and Wardrop's user equilibrium conditions are that flows are found precisely on the shortest routes given the current costs $c(h^*)$.

In order to make it easier to discuss important elements of sensitivity analysis in the next section, we need to rephrase the variational inequality as that of a set inclusion, based on the normal cone operator. For a given closed and convex set $P \subset \mathbb{R}^n$, we define the normal cone operator as follows:

$$N_P(x) := \{\, p \in \mathbb{R}^n \mid p^{\mathrm{T}}(y - x) \leq 0, \qquad \forall y \in P \,\}. \tag{3}$$

Suppose the cost operator is $f : \mathbb{R}^n \to \mathbb{R}^n$. Then it is immediate to see that $x^* \in P$ is a solution to the variational inequality described by $x^* \in P$ satisfying

$$f(x^*)^{\mathrm{T}}(x - x^*) \geq 0, \qquad \forall x \in P,$$

if and only if

$$-f(x^*) \in N_P(x^*)$$

holds.

2.2. The design/pricing problem

We now introduce a set $X := \{\, x \in \mathbb{R}^{|\mathcal{L}|} \,\big|\, x^L \leq x \leq x^U \,\}$ of link "design" parameters. They may describe physical changes made to the traffic network, and/or the workings of a system of link tolls. (In order to ease the presentation, if for a given link l no design or toll is considered we will simply let the lower and upper bounds—x_l^L and x_l^U, respectively—of the link's design parameter x_l be equal to zero.)

Utilizing the above development, we represent Wardrop's user equilibrium conditions as follows:

$$0 \in t(x, v) + N_V(v), \tag{4}$$

where $N_V(v)$ is the normal cone of the feasible set of link flows V. Let the mapping $S : X \to \mathbb{R}^{|\mathcal{L}|}$ be defined by

$$S(x) := \left\{\, v \in \mathbb{R}^{|\mathcal{L}|} \,\middle|\, 0 \in t(x, v) + N_V(v) \,\right\}; \tag{5}$$

at x the set $S(x)$ contains all user equilibrium link flows v.

For the design/pricing problem, we denote our objective function by $F(x, v)$. The problem at hand can then be written as that to

$$\begin{aligned} \underset{(x,v)}{\text{minimize}} \quad & F(x, v), \\ \text{subject to} \quad & x \in X, \\ & v \in S(x), \end{aligned} \tag{6}$$

where $S(x)$ is given by (5). This is a special case of an MPEC (Mathematical Program with Equilibrium Constraints, Luo *et al.* (1996)) which extends the notion of a bilevel program to the case where the lower-level problem is more generally modelled as a variational inequality rather than as an optimization problem.

In the next section we discuss when and how sensitivities of equilibria are defined.

3. Sensitivity analysis of traffic equilibria

In an implicit programming (IMP) approach to solving the problem (6) one considers the implicit nonlinear program (NLP) to

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \hat{F}(x) := F(x, S(x)), \\ \text{subject to} \quad & x \in X. \end{aligned} \tag{7}$$

Here, we assume that the equilibrium flow v is uniquely given by the parameter vector x through the relation $v = S(x)$ [as will be the case provided that the link cost function $t(x, \cdot)$ is strictly monotone]; hence, the two problems (6) and (7) express equivalent problems.

General-purpose NLP solvers often use gradients, or subgradients, of the objective function to compute descent directions and check for optimality. The objective function $\hat{F}$ in (7) is however not differentiable in general; this is in part due to the fact that the (uniquely given) equilibrium link flow $v(x)$ corresponds to a whole range of possible route flow solutions $h(x)$, some of which may have the property that some route is a cheapest one but is nevertheless not used. For the purpose of a simple exposition, we shall not provide the details of this analysis here, but refer to Qiu & Magnanti (1989); Patriksson & Rockafellar (2003); Patriksson (2004) for more detailed accounts.

For (7) to be practically solvable with general-purpose NLP solvers, we assume that S is semi-differentiable at any $x \in X$, which is the same as saying that the traffic equilibrium link flows must be unique at x and vary in a Lipschitz continuous manner around x (Patriksson & Rockafellar (2003)). [A sufficient condition for this property is that t is differentiable and $t(x, \cdot)$ is strongly monotone; this is then particularly true for strictly monotone, affine cost functions $t(x, \cdot)$.] Then, according to the theory developed in Patriksson & Rockafellar (2003), (sub)gradients of S at x (actually Jacobians) can be calculated almost everywhere as composed by solutions to affine approximations of the traffic equilibrium conditions, that is, approximation of the problem in v defined in (5). Depending on the cost function model, S might or might not be trivially shown to be semi-differentiable. For the polynomial BPR link cost functions, for example, the case is not trivial due to the zero derivative at zero flow links. It can, however, be shown (see Hellman (2010)) that a sufficient condition for S to be semi-differentiable is that the unused links that are part of unused and cheapest routes do not form cycles. This condition is fulfilled in all experiments presented below.

The proposed method to solving (7) is to efficiently and precisely compute $\hat{F}(x)$ and its gradient (or, subgradient) in order to make (7) a possible subject to general-purpose NLP solvers which may utilize the high precision of the computations of the included functions. Since $F(x)$ (implicitly) depends on $S(x)$, we are interested in the gradient (or subgradient) of $S(x)$. Computing $S(x)$ amounts to solving the traffic equilibrium problem, for which efficient and precise methods exist, as discussed above. The contribution of this paper is a method for efficient and precise computations of gradients or subgradients of S, based on features from the very efficient and exact traffic equilibrium method TAPAS (Bar-Gera (2010)). Next, we will focus on directional derivatives of S, which will assemble a gradient if it exists.

We will use the theory developed in Patriksson & Rockafellar (2003) on how sensitivity analysis of the variational inequality in (4) yields directional derivatives of S. It involves studying additive perturbations v' of v, and h' of h, while perturbing x additively by x'. In our simplified case, the demand is fixed; hence we let $d' = 0$. The indexing introduced above for addressing elements of the flow vectors (e.g., h_r) will be used also for flow perturbations. As h and v were linearly related in (2), so will also h' and v' be, i.e., $\Gamma^{\mathrm{T}} h' = 0$ and $\Lambda h' = v'$.

From Patriksson & Rockafellar (2003) we have that under the semi-differentiability condition of S at a reference point $x = x^*$, the directional derivative $DS(x^*)(x')$ is the solution to the following variational inequality for v'; it is a first-order approximation of the traffic equilibrium problem around $v^* = S(x^*)$:

$$-\nabla_x t(x^*, v^*)x' \in \nabla_v t(x^*, v^*)v' + N_K(v'), \tag{8}$$

where K [or, $K(v^*)$] is the critical cone of V at v^*, i.e.,

$$K = \left\{ v' \in \mathbb{R}^{|\mathcal{L}|} \,\middle|\, v' \in T_V(v^*) \text{ and } (v')^{\mathrm{T}} t(x^*, v^*) = 0 \right\}, \tag{9}$$

and $T_V(v^*)$ is the tangent cone of V at v^*. The cone K is composed by the link flow shifts v' for which perturbed link flows are first-order feasible and optimal with respect to the traffic equilibrium problem around v^*. Since K is a convex polyhedron, the variational inequality (8) can be stated as an affine variational inequality in general, and as a quadratic program in special cases. The representation of the cone K is central and will be the topic for the following discussion.

For a strictly complementary traffic equilibrium the cone K is a subspace, due to the fact that both negative and positive route flow perturbations h'_r are possible for all used routes r. In this case, (8) defines a linear mapping from x' to v'. For a non-strictly complementary traffic equilibrium, the cone K is not a subspace, since there is at least one degenerate route r for which route flow perturbations h'_r must not be negative. In this case, (8) does not define a linear mapping.

Non-strictly complementary traffic equilibria, i.e. solutions with degenerate routes, can be considered from different perspectives. If degenerate solutions are identified, one way to evaluate their sensitivities is to replace K with the subspace

$$\bar{K} = K \cap (-K), \tag{10}$$

where $-K = \left\{ -v' \in \mathbb{R}^{|\mathcal{L}|} \,\middle|\, v' \in K \right\}$. The resulting approximation of (8) defines a linear mapping from x' to v'. A thorough analysis of this approximation can be found in Patriksson (2004). A result from the same source implies that the n solutions to the approximation of (8) resulting from using $\bar{K}$ instead of K for unit vectors $x' = e_i$, $i = 1, \ldots, n$, form a subgradient of $S(x^*)$ under the condition that $\bar{K} = K(S(x^* + \tau x'))$ for some x' and all sufficiently small $\tau > 0$.

From a practical computational point of view, at least in the context of network design, degeneracy is not a major concern due to several reasons. The equilibrium at almost all design points is not degenerate. Therefore degeneracy is rarely observed, although nearly-degenerate solutions may be found. Due to computational precision limitations it is very difficult practically to distinguish between degenerate and nearly-degenerate solutions. The traffic assignment method used here, TAPAS, was not designed for identifying degeneracy. Therefore the degeneracy may be "hidden" in the sense that it cannot be identified from the TAPAS outputs. Further observations regarding degeneracy in our computational results are presented in Section 6.2.

Regarding the representation of $\bar{K}$, we let it be the span of a set of vectors, i.e., $\bar{K} = \operatorname{span}(p_1, p_2, \ldots, p_k)$. Introducing a matrix $P = (p_1, p_2, \ldots, p_k)$, $\bar{K}$ is the column space of P. As will be described in Section 4, the traffic assignment method TAPAS generates a matrix which closely resembles P, and which can be used in the proposed method. However, the matrix P generated is in general not of full rank—an issue that is further discussed below.

Using the above-mentioned heuristic allows us to approximate (8) as the quadratic program to

$$\begin{aligned} \underset{(v',z')}{\text{minimize}} \quad & \frac{1}{2}(v')^{\mathrm{T}} D v' + b^{\mathrm{T}} v', \\ \text{subject to} \quad & v' = P z', \end{aligned} \tag{11}$$

where $z' \in \mathbb{R}^k$ is a set of auxiliary variables, $D = \nabla_v t(x^*, v^*)$ is a diagonal matrix and $b = \nabla_x t(x^*, v^*) x'$. The program in (11) has the same unique solution in v' as the following linear equation system,

$$\begin{aligned} P^{\mathrm{T}} D P z' &= -P^{\mathrm{T}} b, \\ v' &= P z'. \end{aligned} \tag{12}$$

Ideally, P is of full rank, implying $P^{\mathrm{T}}DP$ is positive definite (some diagonal elements of D might be zero, but recall that we work under the conditions of semi-differentiability) and (12) can be solved using for example, the conjugate gradient (CG) method. In practice, P might be rank deficient, leading to $P^{\mathrm{T}}DP$ being singular. The right-hand side $-P^{\mathrm{T}}b$ is in the range of $P^{\mathrm{T}}DP$, rendering the singular linear system compatible, i.e., with infinitely many solutions z rather than none. Due to semi-differentiability, the solution v' is unique regardless of solution z, and we can use a method capable of handling compatible singular systems resulting in bounded solutions of z. The minimum residual (MINRES) method (Paige & Saunders (1975)) was in Choi *et al.* (2011) shown to be such a method. We denote by $\widetilde{DS}(x^*)(x')$ the solution v' to (12).

Regardless of differentiability state of $S(x^*)$, a "Jacobian"

$$\widetilde{JS}(x^*) = \begin{pmatrix} \widetilde{DS}(x^*)(e_1)^{\mathrm{T}} \\ \widetilde{DS}(x^*)(e_2)^{\mathrm{T}} \\ \vdots \\ \widetilde{DS}(x^*)(e_n)^{\mathrm{T}} \end{pmatrix} \tag{13}$$

of $S(x^*)$ is formed using the heuristic approximation $\widetilde{DS}$ of DS, where e_i is the ith standard basis vector in $\mathbb{R}^n$. As reported above, if S is differentiable at x^*, then $\widetilde{JS}(x^*)$ is the true Jacobian. If S is not differentiable at x^*, then $\widetilde{JS}(x^*)$ still has the potential for representing a Jacobian-like approximation. Algorithm 1 shows the steps to compute $S(x^*)$ and $\widetilde{JS}(x^*)$ in practice.

Algorithm 1 Proposed algorithm for computing traffic equilibrium (sub)gradient

```
Use TAPAS to compute v* = S(x*) and a matrix P, according to the heuristic
for i = 1 → n do
    Choose x' as e_i
    Solve (12) for z using MINRES
    Compute DS~(x*)(e_i) as Pz
end for
Assemble JS~(x*) using DS~(x*)(e_i), i = 1, . . . , n according to (13)
```

In the following section we briefly describe the traffic assignment method TAPAS and how its outputs are instrumental in efficiently generating the matrix P. Section 5 presents a method for evaluating the errors in a derivative, and Section 6.1 utilizes that method to evaluate the errors of the gradients computed with the method proposed in this section.

4. Traffic assignment by paired alternative segments (TAPAS)

Traffic Assignment by Paired Alternative Segments (TAPAS) (see Bar-Gera (2010)) was developed to solve the user equilibrium problem efficiently and to high levels of convergence, while addressing the issue of route flow uniqueness by incorporating the assumption of proportionality.

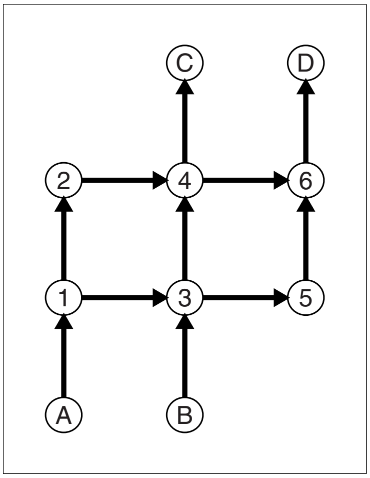
In TAPAS, the precision of a feasible solution pair (h, v) of route and link flows in the user equilibrium problem is measured in terms of the *average excess cost* (AEC). The AEC is, in short, the difference between the total travel cost (that is, the value of $c(h)^{\mathrm{T}}h$, or equivalently, $t(v)^{\mathrm{T}}v$) of the feasible solution pair (h, v) and the total cost of the all-or-nothing (that is, shortest route) solution that is produced given the current flow's travel costs $(c(h), t(v))$, divided by the total demand. TAPAS is able to reach an AEC value below 10^{-12} even on large-scale networks.

The main concept of the algorithm is the focus on pairs of alternative segments (PASs), achieved by storing a set of PASs, as well as a list of relevant origins for each PAS, which are updated throughout the algorithmic process. PASs are used for two purposes: 1. cost equilibration by simply shifting flow from one segment to the other as needed; and 2. proportionality adjustments.

The TAPAS algorithm chooses a set of PASs that spans the subspace of feasible flow shifts among used routes. Several considerations are taken into account by the algorithm in this choice, including preference to short segments and a compact representation, but there are other considerations, and linear independence is not guaranteed (see also the previous section).

The output of TAPAS includes a description of the chosen set of PASs. In the proposed method, this set is used to construct the matrix P in (12). Along with each PAS vector are the flows on its two segments given. If a PAS in the set has at least one segment with zero flow it is discarded. The remaining PAS vectors are assembled to form the columns of P. The discarded PAS vectors correspond to degenerate routes. Discarding these PAS vectors may or may not affect the rank of P. If the rank remains the same, then as far as we can tell from the algorithm output K and $\bar{K}$ are the same. If the rank of P decreases, this reduction of feasible flow shifts is the heuristic presented in (10), where K is approximated by $\bar{K}$. No attempt is made to remove the linear dependency of the PAS vectors, since that issue is resolved by using a linear equations solver capable of handling compatible singular systems.

For illustration purposes, the output PAS matrix for the network in Figure 1 may be the one shown in the corresponding table. This matrix is linearly dependent, as $PAS_1 + PAS_2 = PAS_3$. The projection on link flows of all possible flow shifts within the set of routes shown in Figure 1 is the linear subspace spanned by the PAS matrix. The resulting matrix can be used to identify first-order feasible link flow perturbations.



Link	[A,1]	[B,3]	[1,3]	[3,5]	[1,2]	[3,4]	[5,6]	[2,4]	[4,6]	[4,C]	[6,D]
PAS_1	0	0	1	0	−1	1	0	−1	0	0	0
PAS_2	0	0	0	1	0	−1	1	0	−1	0	0
PAS_3	0	0	1	1	−1	0	1	−1	−1	0	0

Fig. 1: a stylized network to illustrate the PAS matrix

This feature, together with solution precision, makes TAPAS a useful TAP solver for sensitivity analysis and network design/pricing optimization.

This concludes the exposition of the proposed method for computing traffic equilibrium (sub)gradients. The computational effort required by the algorithm, as described in Algorithm 1, is relatively modest. The next step is to evaluate the gradient precision numerically. The method we chose for the evaluation of derivative precision is presented next, followed by the computational results of the evaluation in Section 6.1.

5. Evaluating the numerical error in a derivative

This section presents a general method for evaluating the error in a derivative, by comparing it with finite difference approximations. This will be used in the subsequent section to evaluate the precision of gradient computations produced by the method proposed in Section 3.

Let f be a differentiable function with f' as its derivative. Both the evaluation of the function and its derivative suffer from numerical errors due to, e.g., limited floating-point precision or an inexact evaluation method. We let $\tilde{f}(x) := f(x) + e_f(x)$ and $\tilde{f}'(x) := f'(x) + e_{f'}(x)$ be the evaluated values of the two quantities, respectively, where $e_f(x)$ and $e_{f'}(x)$ are the errors. We consider three finite difference schemes:

1. forward-difference approximation, $\tilde{g}_F^{FD}(h; x) := \frac{\tilde{f}(x+h)-\tilde{f}(x)}{h}$,
2. central-difference approximation, $\tilde{g}_C^{FD}(h; x) := \frac{\tilde{f}(x+h)-\tilde{f}(x-h)}{2h}$, and
3. five point stencil approximation, $\tilde{g}_5^{FD}(h; x) := \frac{-\tilde{f}(x+h)+8\tilde{f}\left(x+\frac{1}{2}h\right)-8\tilde{f}\left(x-\frac{1}{2}h\right)+\tilde{f}(x-h)}{6h}$,

where h is the step length. We focus first on the forward-difference approximation and investigate its error. Expanding $\tilde{f}$ in $\tilde{g}_F^{FD}$, we obtain

$$\tilde{g}_F^{FD}(h; x) = \frac{f(x+h) - f(x)}{h} + \frac{e_f(x+h) - e_f(x)}{h}. \quad (14)$$

The first term in (14) is an exact forward difference for f and can be written as $f'(x) + O(h)$ as $h \to 0$. Assuming $\left|e_f(x+h) - e_f(x)\right| \leq \epsilon$, where ϵ is some non-negative constant independent of h, the second term can be written as $O(h^{-1})$ as $h \to 0$. This gives

$$\tilde{g}_F^{FD}(h; x) = f'(x) + O(h) + O(h^{-1}) \text{ as } h \to 0. \tag{15}$$

Further, we can compute the relative difference

$$r_F(h; x) = \frac{\left|\tilde{f}'(x) - \tilde{g}_F^{FD}(h; x)\right|}{\left|\tilde{f}'(x)\right|} \tag{16}$$

between the finite difference approximation and our approximation $\tilde{f}'(x)$. When expanding $\tilde{f}'(x)$ (as defined) and $\tilde{g}_F^{FD}(h; x)$ from (15) in the expression for $r_F(h; x)$ in (16), we obtain

$$r_F(h; x) = \frac{\left|e_{f'}(x) + O(h) + O(h^{-1})\right|}{\left|\tilde{f}'(x)\right|}.$$

Similarly, we can define the relative difference for the central difference and five point stencil as

$$r_C(h; x) = \frac{\left|e_{f'}(x) + O(h^2) + O(h^{-1})\right|}{\left|\tilde{f}'(x)\right|},$$

and

$$r_5(h; x) = \frac{\left|e_{f'}(x) + O(h^4) + O(h^{-1})\right|}{\left|\tilde{f}'(x)\right|},$$

respectively.

The derivative error evaluation method consists of examining the graph $(h, r_X(h; x))$, where X is one of F, C or 5 depending on finite difference scheme to use as a reference. From the expressions derived for the relative differences, we can see that for small h, the $O(h^{-1})$ term (from $\tilde{f}$) dominates, and for large h, the $O(h)$, $O(h^2)$ or $O(h^4)$ term (from the truncation error of the finite difference approximations) dominates. If the error $e_{f'}$ that we are looking for is large enough, it will dominate for some h neither too small nor too large. A series of examples follow where we incrementally add error sources to illustrate how we can use this method to evaluate the error in a derivative.

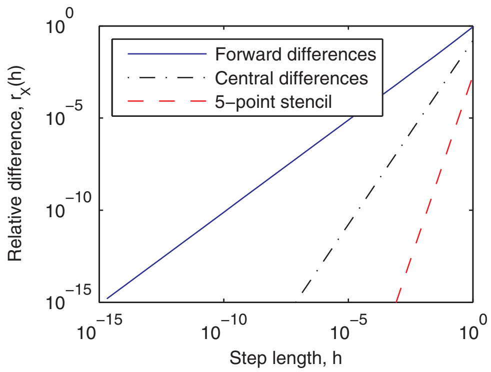
We consider the function $f(x) = \sin(x)$ with $f'(x) = \cos(x)$ and investigate its relative difference graphs (plotted in Figure 2) in four cases at $x = 1$. In some of the following examples, the function and its derivative are claimed to be evaluated exactly. For this illustration, they can be considered exact, but in practice they have been evaluated using 32 decimal digit arithmetic.

(a) *Error in neither $\tilde{f}(x)$, nor $\tilde{f}'(x)$.* In this case $\tilde{f}(x) = f(x)$, and $\tilde{f}'(x) = f'(x)$, i.e., the evaluation of the functions are exact. The graph $(h, r_X(h; x))$ for this case is plotted in Figure 2a for all three finite difference schemes ($X = F$, C and 5). Note that the slopes differ and correspond to the $O(h)$, $O(h^2)$ and $O(h^4)$ terms in the relative differences.

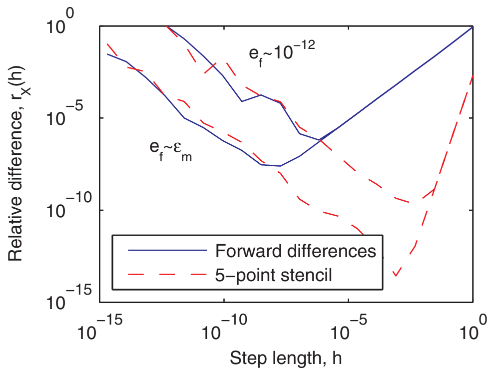
(b) *Error in $\tilde{f}(x)$, but no error in $\tilde{f}'(x)$.* In this case, the function evaluation has errors, while the derivative is exactly evaluated. The graph $(h, r_X(h; x))$ for this case is plotted in Figure 2b for the forward difference and five point stencil ($X = F$ and 5). We have two graphs for each scheme: one for a case where all computations are made in 8-byte floating point arithmetic (error ϵ_m), and one case where an additional relative error of 10^{-12} is added. Here we observe the $O(h^{-1})$ term dominating for small enough h.

(c) *No error in $\tilde{f}(x)$, but error in $\tilde{f}'(x)$.* In this case, the function evaluation is exact, while the derivative has errors. The graphs for this case are plotted in Figure 2c. Again, we have two graphs for each scheme: one for a case where a relative error of 10^{-6} is added to the derivative, and one case where a relative error of 10^{-8} is added. We can see the h-independent $e_{f'}$ term dominating for small enough h.

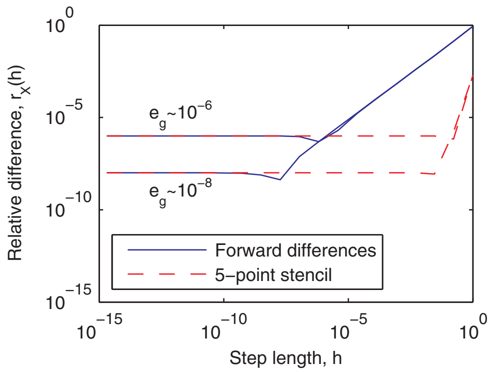
(d) *Error in both* $\tilde{f}(x)$ *and* $\tilde{f}'(x)$. In this case, both function and derivative evaluation suffer from errors. The graphs for this case are plotted in Figure 2d. Relative errors of 10^{-14} and 10^{-8} are added to the function and derivative respectively. In this figure, we can see all three terms dominating for different values of h in the five point stencil graph (dashed line), while the forward difference scheme (solid line) is too imprecise to render the derivative error.



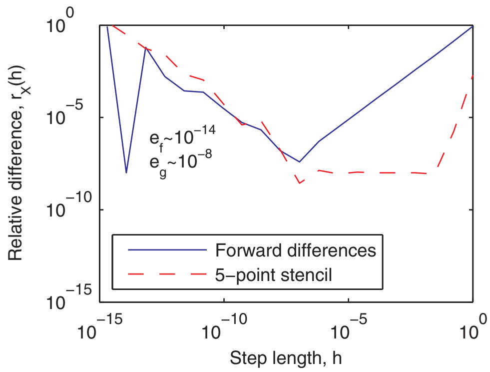
(a) No errors.



(b) Errors in $\tilde{f}$ only.



(c) Errors in $\tilde{f}'$ only.



(d) Errors in both $\tilde{f}$ and $\tilde{f}'$.

Fig. 2: An illustration of relative difference graph behavior for different error scenarios.

The derivative error evaluation method consists of examining graphs such as those plotted in Figure 2. One cannot, however, directly take the smallest relative difference value found in such a graph as an estimate of the derivative error since there is a risk that the errors cancel, with the effect of an optimistic estimate. In our experiments below, a moving average of relative differences over h is used to reduce this risk. Another problem with this method is that the reference derivative, that is, finite difference approximations and function evaluations, may not be exact enough to have the derivative error term dominate for any h. In that case we at least get an upper estimate of the derivative error.

6. Numerical experiments

6.1. Experiment 1—Evaluating gradient precision

Sensitivity analysis of an equilibrium solution allows us to compute gradients or subgradients of functions dependent on the traffic equilibrium variables, e.g., the objective function in the traffic design problem (7). The experiments of this section investigate the error of the derivatives and gradients acquired by the proposed method. The first experiment, in Section 6.1.1, is performed on the network due to Braess (1968) in a toll pricing setting, where we compare a numerically computed derivative with an analytical solution. In Section 6.1.2, we consider a design problem on a medium-size network and investigate the error in a directional derivative at a single point of the upper-level objective function using the framework presented in Section 5. In Section 6.1.3 we extend the previous experiment by investigating the directional derivative at 1000 points in order to get a statistically significant estimate of the derivative error.

6.1.1. Gradient evaluation for Braess' network

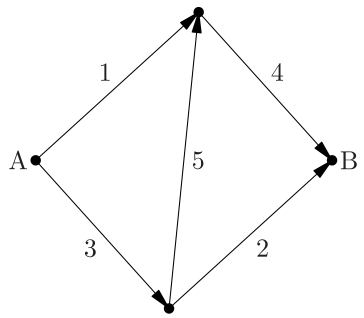


Fig. 3: Braess' network

Consider Braess' network depicted in Figure 3. In this network we have one OD-pair, from node A to B, with travel demand $d = 6$ units. There are three routes using links $(1, 4)$, $(3, 2)$ and $(3, 5, 4)$ with flows h_1, h_2 and h_3, respectively. The link travel costs are $t_i = 50 + v_i$ for $i = 1, 2$, $t_i = 10v_i$ for $i = 3, 4$ and $t_5 = 10 + v_5 + x$, where $x \geq 0$ is a toll price. For $x = 0$, we have equal flow on all three routes, i.e. $h_1 = h_2 = h_3 = 2$. For $x \geq 13$, we have no flow on h_3, i.e., $h_1 = h_2 = 3$ and $h_3 = 0$. When all three routes are used we have, by Wardrop's user equilibrium condition, that the route costs c_r of all three routes are equal to the same value c, and we obtain the following equations determining the flows on the three routes:

$$\begin{aligned} c_1 &= (50 + h_1) + 10(h_1 + h_3) = c, \\ c_2 &= 10(h_2 + h_3) + (50 + h_2) = c, \\ c_3 = 10(h_2 + h_3) + (10 + h_3 + x) &+ 10(h_1 + h_3) = c, \\ h_1 + h_2 + h_3 &= 6. \end{aligned} \tag{17}$$

Suppose we are interested in the derivative $\frac{\partial h_3}{\partial x}$. If we solve (17) for h_3 in x, we obtain

$$h_3 = 2 - \frac{2}{13}x,$$

where we see that the derivative sought is equal to $-\frac{2}{13}$. We choose a reference point $x^* = 1 < 13$, so that all routes are used and the analysis above is valid. We perform a sensitivity analysis according to the proposed method to generate the Jacobian $\widetilde{JS}(x^*) = \frac{\partial(v_1,v_2,v_3,v_4,v_5)}{\partial x}$ in (13) above, using 8-byte floating point arithmetic for any arithmetic operation. Since $h_3 = v_5$, we seek the fifth component of $\widetilde{JS}(x^*)$, corresponding to $\frac{\partial v_5}{\partial x}$. From the numerical experiment, we get

$$\widetilde{JS}(x^*) = \begin{pmatrix} 7.692307692307692735 \times 10^{-2} \\ 7.692307692307692735 \times 10^{-2} \\ -7.692307692307692735 \times 10^{-2} \\ -7.692307692307692735 \times 10^{-2} \\ -1.538461538461538547 \times 10^{-1} \end{pmatrix}^{\mathrm{T}}.$$

The fifth component of $\widetilde{JS}(x^*)$ sought is bit identical to $-\frac{2}{13}$ computed with 8-byte floating point arithmetic.

6.1.2. Gradient evaluation at a single point on a medium-sized network

In this experiment we will investigate the precision of sensitivities on a medium-sized network at a single point. We consider the traffic assignment subproblem of a capacity expansion design problem. The underlying network is the Anaheim network (Bar-Gera (2001)) with 1406 OD-pairs, 416 nodes and 914 links. We have a single variable x which is the (equal) capacity expansion on 30 links, whose costs are defined by BPR functions. More specifically, we define the capacity of link $l \in \mathcal{L}_E \subset \mathcal{L}$ as

$$C_l = C_l^0 + x,$$

where C_l^0 is the original link capacity of the link and $\mathcal{L}_E$ is a set of 30 links. Using the same link numbering as in the source Bar-Gera (2001), we let

$$\begin{aligned}\mathcal{L}_E = \{&2, 102, 103, 104, 106, 107, 134, 135, 137, 286, 288, 289, 290, 291, 292,\\ &293, 296, 297, 298, 299, 301, 302, 304, 305, 307, 308, 310, 311, 312, 782\}.\end{aligned} \tag{18}$$

This is a set of connected links with high marginal costs for which we can expect a capacity expansion to have a congestion-reducing effect. We will not study the full capacity expansion problem, but only the derivative of the link flows wrt. x. This makes it unnecessary to consider the upper-level objective function F and we hence let it remain undefined.

We select a reference point $x^* = 0$, and perform a sensitivity analysis according to the proposed method to compute $\widetilde{JS}(x^*)$, which is a 914×1 matrix in this case. Since we have a vector of derivative approximations $\widetilde{JS}(x^*)$ to compare with the exact derivatives $JS(x^*)$, we select one component to be our $f(x)$. We select the component for which the ratio between the approximated link flow derivative and the link flow is the largest. For that component the relative error due to function value error or rounding error in the finite difference approximations will be minimal and hence we will have a better estimate to compare our sensitivity analysis based derivative against. According to the above rule, link number 412 was selected, i.e., we evaluate the quality of $f(x) := v_{412}(x)$. The unperturbed equilibrium flow and cost at link 412 is 0.8645 and 1.9201, respectively, which can be compared with the average over all links: 20.0996 and 0.9054, respectively. We set the termination condition of TAPAS to average excess cost less than 10^{-15}.

The derivative approximation $\widetilde{JS}(x^*)$ was computed using the proposed method, and the relative differences according to the framework in Section 5 of its 412$^{\text{th}}$ component were computed for 30 step lengths h logarithmically distributed between 10^{-6} and $10^{-0.7}$. The resulting graph is plotted in Figure 4a. It is evident from the figure that neither forward differences nor central differences are precise enough to obtain the error in the derivative, but it is unclear from the figure whether the five-point stencil is. A zoom-in of the graph is plotted in Figure 4b, where 1000 step lengths h logarithmically distributed between $10^{-0.7}$ and $10^{-1.5}$ have been used for generating the graph. For the largest values of h (to the right in the figure), the $O(h^4)$ term dominates. Around 10^{-1} there is a shift from $O(h^4)$ to $O(h^{-1})$ as the dominating term. It is not easy to say whether there is a constant term dominating in-between, but we can at least draw the conclusion that the relative error in the sensitivity analysis based derivative of $v_{412}(x)$ at $x = 0$ is less than 10^{-10}.

6.1.3. Gradient evaluation at several points on a medium-sized network

Using the same design problem as in the previous section, we will evaluate sensitivity analysis based derivatives at several points x.

We choose to only use the five point stencil approximation of first derivative since the previous experiments show that it makes a better approximation than the others. We select 10 step lengths h_j logarithmically distributed between 10^{-2} and $10^{-0.7}$, i.e., $h_1 = 10^{-2}, h_{10} = 10^{-0.7}$ and $\frac{h_j}{h_{j-1}} = \frac{h_{j+1}}{h_j}$ for $j = 2, \dots, 9$. This is the step length interval where the five point stencil has shown best results.

For the five point stencil to give a good approximation, it is necessary that the function (of which we approximate a derivative) is smooth in the interval covered by the five point stencil. To ensure that, we pick 1000 reference points x_i^* according to the following scheme:

1. Generate a random number x_c^* from a uniform probability distribution on $[0, 10]$ as a candidate for the next x_i^*.

2. Check if S is smooth at x within the interval $x_c^* - 10^{-0.7} \leq x \leq x_c^* + 10^{-0.7}$. This is done by comparing the range of the PAS matrix of 512 linearly distributed points in the interval. If the ranges of those matrices for all points in the interval are equal, then let x_c^* be the next x_i^*. If not, generate a new candidate and perform the same check for the new candidate point.

When performing the experiment, we considered 2019 candidates to generate 1000 points x_i^*, i.e., 1019 candidates were discarded due to non-smoothness. For each point x_i^* we use the following scheme to compute an upper estimate $r_i^{\min}$ of the derivative error:

1. Compute the relative difference $r_{i,j}$ for the 10 step lengths h_j according to the rule in the previous experiment, i.e., we choose the link l whose ratio between link flow derivative and link flow is the largest and use that as our function $f(x) := v_l(x)$ when computing the relative difference.

2. Compute a moving average over step lengths, $r_{i,j}^{\text{avg}} = \frac{1}{3}\left(r_{i,j-1} + r_{i,j} + r_{i,j+1}\right)$ for $j = 2, \ldots, 9$. The rationale behind this computation is to remove the effect of coincidental matches between the finite difference approximation and the sensitivity analysis based derivative.

3. Select the minimum among the averages, $r_i^{\min} = \min_j r_{i,j}^{\text{avg}}$.

During traffic assignment we ensure that all user equilibrium solutions converge to an average excess cost of less than 10^{-15}.

A histogram of $r_i^{\min}$ for the 1000 points is shown in Figure 5. There is a large mass of relative differences in the range 10^{-12} to 10^{-11} and some mass just below 10^{-9}. From this histogram we can draw the conclusion that the relative error in the sensitivity analysis based derivative is not greater than 10^{-9}.

The appearance of the histogram suggests that it is a sum of distributions. A scatter plot of all relative differences versus their respective design variable values is shown in Figure 6. In the figure we can see that there are regions of variable values for which there are no reference points. These are regions for which the reference point candidates were excluded according to the above mentioned scheme, i.e., the candidates sampled there were too close to a potential non-smoothness in the link flow functions. It is evident from the scatter plot that the relative difference at a reference point is dependent on the smooth region in which the reference point resides.

In order to find the source of the larger error for some regions, the 18 reference points satisfying $2 < x_i^* < 4$, and with a minimum average relative difference greater than 10^{-11}, were selected for closer inspection. These points can be identified as those in Figure 6 forming the upper-most cluster. The average relative difference $r_{i,j}^{\text{avg}}$ is plotted against h_j for the 18 selected reference points in Figure 7. The plotted points belonging to the same reference point are connected with lines. Further, a least squares first-degree polynomial fit was performed to the data. The resulting line has a slope of −0.9466 after log-transforming the axes. This is a strong indicator for the relative difference to stem from an error in the finite difference approximation rather than the sensitivity analysis based derivative, since we would expect a slope of −1 and 0 in the former and latter case respectively.

This experiment shows that the low relative derivative error found in the previous subsection was no coincidence. The error of the sensitivity based derivative is kept at low levels for a range of design variable values and our evaluation method has only been able to give us upper estimates of the error.

6.2. Experiment 2—Network design: comparison with other methods

Our next goal is to examine the effect of precise derivative computations on the ability to identify optimal solutions for hierarchical (bi-level) models. For that purpose we examine the Sioux Falls capacity expansion problem presented in Suwansirikul *et al.* (1987). The problem is to minimize the sum of total travel cost and investment cost with respect to ten variables for capacity expansion on ten links. Solutions to this specific problem has previously been published for different initial guesses using the methods CSP (Chiou (2008)), PIPA (Lim (2002)), and SBD (Josefsson & Patriksson (2007)). We solve the same problem using two general nonlinear optimization solvers, SNOPT (Gill *et al.* (2005)) and PBUN (Lukǔ & Vlček (2001)), where the objective function value and gradient are computed by the method proposed in Sections 3 and 4. SNOPT is an SQP method for smooth problems, i.e., not ideally suited for the task, since we have a non-smooth objective function. PBUN is a bundle method developed to handle non-smooth objective functions.

We solve the problem twice, alternately using SNOPT and PBUN, from two initial guesses: all variables equal to 0 (Table 1); and all variables equal to 6.25 (Table 2). (Notice however that PIPA does not make use of an initial guess.) For all runs, $x_L = 0$ and $x_U = 25$ was used as lower and upper variable bounds, respectively. Resulting solution variable values and objective function values obtained are presented for the proposed method and the methods CSP, PIPA, and SBD in Tables 1 and 2.

Method	Proposed method SNOPT	Proposed method PBUN	PIPA	SBD
x_{16}	5.2487	5.3419	5.4680	5.3027
x_{17}	1.4993	2.0711	2.0039	2.0560
x_{19}	5.2768	5.3688	5.4471	5.3430
x_{20}	1.4755	2.0438	1.9395	1.9901
x_{25}	2.9207	2.5223	2.9448	2.5216
x_{26}	2.9681	2.5672	2.8191	2.5548
x_{29}	3.9282	2.7693	3.4039	2.9883
x_{39}	4.8922	4.7923	4.8061	4.8559
x_{48}	3.9622	2.8258	3.2364	3.0026
x_{74}	4.8819	4.7798	4.7779	4.8496
F (reported)	80.1461	79.9003	80.8669	79.9961
F (recalc.)	80.1461	79.9003	79.9601	79.9126
Eq. sol. count	72	55	?	?
Time (s.)	12.4	8.9	?	?

Table 1: Numerical results for Sioux Falls capacity expansion problem, initial guess 0.

Method	Proposed method SNOPT	Proposed method PBUN	CSP	SBD
x_{16}	5.3520	5.3275	4.1007	5.2773
x_{17}	2.0732	2.0629	4.1254	2.0533
x_{19}	5.3813	5.3667	1.9807	5.3002
x_{20}	2.0455	2.0294	1.4532	2.0369
x_{25}	2.4603	2.5174	1.5643	2.7670
x_{26}	2.5102	2.5493	3.0987	2.8222
x_{29}	2.7715	2.7673	4.9876	3.0124
x_{39}	4.8032	4.8057	5.1095	4.7348
x_{48}	2.8279	2.8256	4.7896	2.9746
x_{74}	4.7888	4.7728	4.1287	4.7511
F (reported)	79.8999	79.9003	81.04	80.0043
F (recalc.)	79.8999	79.9003	82.2552	79.9237
Eq. sol. count	512	30	?	?
Time (s.)	85.9	5.3	35 (in 2005)	?

Table 2: Numerical results for Sioux Falls capacity expansion problem, initial guess 6.25.

In the tables, first the ten variable values (rounded to four decimals) at the solution are presented. For the proposed method (columns SNOPT and PBUN), "F (reported)" presents the objective function value evaluated at the solution before rounding, while "F (recalc.)" presents the objective function value evaluated at the rounded solution variable values. For the remaining methods (columns PIPA, SBD, CSP), "F

Method		SNOPT		PBUN	
Initial guess		0	6.25	0	6.25
Removed PASs	0	69	493	47	29
	1	2	17	5	1
	2	0	2	2	0
	3	0	0	0	0
	4	0	0	1	0
	Σ	71	512	55	30

Table 3: Frequencies of number of removed PASs for the equilibria obtained during local search using local search methods SNOPT, PBUN and initial guesses 0 and 6.25.

(reported)" presents the optimal objective function value presented in the source paper, while "F (recalc.)" is the objective function value recalculated using the proposed method for the given solution variable values.

When available, "Eq. sol. count" is the number equilibrium solutions required to obtain the design solution. The last row specifies the following: in case of the proposed method, the time required to compute the design solution, using a single core on a mid-end personal computer in 2012; while for CSP, the time given in the source paper. No time was presented for the other methods. During traffic assignment we let all user equilibrium solutions converge to an average excess cost of less than 10^{-15}.

The total travel cost of the unexpanded network is 99.9413, while the total travel cost and expansion cost for the best solution found (SNOPT with $x^0 = 6.25$) is 79.8999, of which 4.4437 is investment cost for the expansion, and the remainder is the total travel cost.

For each equilibrium computed during the local search in the proposed method, the PAS vectors with at least one zero-flow segment are discarded before constructing the PAS matrix. In Table 3 are recorded the frequencies of the number of removed PASs for the equilibria obtained during the local search for all four combinations of method and initial guess. Each column adds up to the total number of equilibria computed before termination of the method. For the majority of equilibria, no PAS vectors are removed. Additionally, there was no reduction of the rank of the PAS matrix due to the removal of the PAS vectors for any of the four setups, i.e., the removal of the PAS vectors didn't affect the set of feasible link flow perturbations for the sensitivity analysis problem. This means that K and $\bar{K}$ were the same at all visited points, as far as we can tell from the TAPAS output.

It is interesting to compare the number of equilibrium solutions required to acquire the design solution. SNOPT with $x^0 = 0$ required 72 equilibrium solutions, and terminated with a higher objective function value of $F = 80.1461$. SNOPT with $x^0 = 6.25$ required 512 equilibrium solutions, and found a solution with $F = 79.8999$. In the former case, SNOPT terminated with the optimality conditions satisfied, while it in the latter case terminated with numerical difficulties, i.e., without satisfying the optimality conditions. In Figure 8 are plotted two graphs corresponding to the objective function values and directional derivative values along the line between the two solutions above. At $\alpha = 0$ on the α-axis we have the solution for SNOPT with $x^0 = 0$, while at $\alpha = 1$ we have the solution for SNOPT with $x^0 = 6.25$. We can see that the former solution may be a smooth local optimum, while the second solution has a discontinuity in the derivative, causing the numerical difficulties for SNOPT.

The proposed method when using PBUN as upper-level solver was profiled over ten consecutive runs in order to investigate the workload distribution. Total time was 86 s., out of which traffic assignment with TAPAS spent 56 s., the sensitivity problem and MINRES used 13 s., and the remaining 17 s. were spent on the upper-level problem, such as evaluating the investment cost function and executing PBUN.

This experiment shows that the proposed method is a competitive method for design problems both in terms of efficiency and solution quality.

7. Conclusions and future research

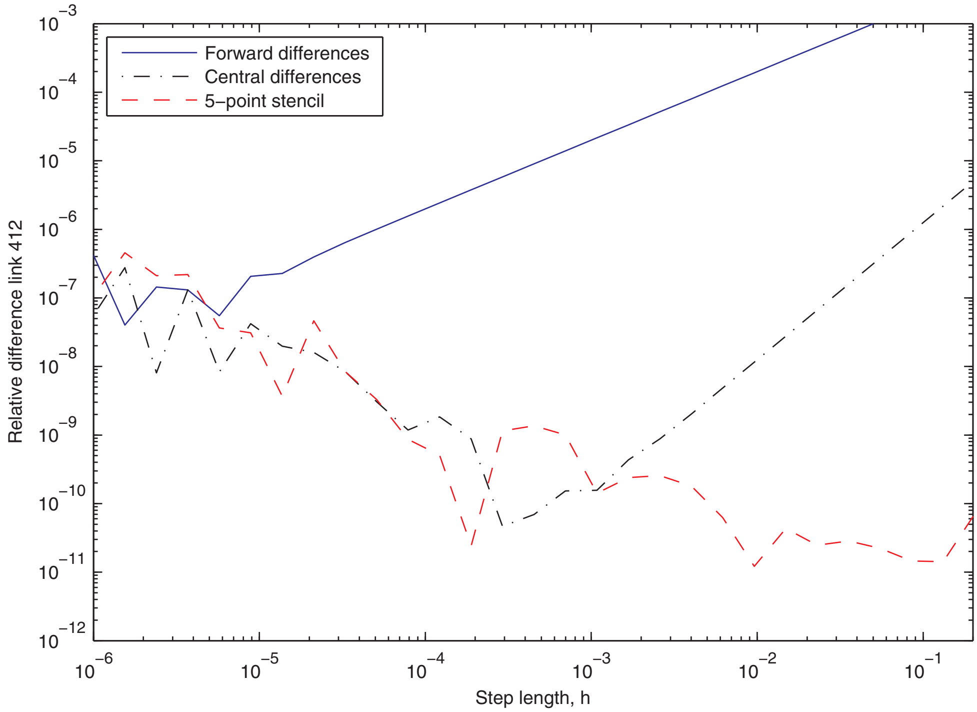
This paper has two purposes. The first (see Section 6.2) is to illustrate the importance of very efficiently, and very precisely, calculating equilibria and their sensitivities to parameter changes, when they are utilized in the search for link tolls and network designs in traffic networks. The second purpose (see Sections 5 and 6.1) is to present and in detail numerically test one such methodology in the case of a fixed demand setting. Our main computational tool is the very efficient and very precise traffic equilibrium code TAPAS (Bar-Gera (2010)), an overview of which is provided in Section 4.

Using highly precise equilibrium solutions (average excess cost 10^{-15} or better) we show that the gradient evaluations computed by the proposed method are very precise as well (relative error of 10^{-9} or better). In contrast, a re-evaluation of past experiments reveal imprecise computations of equilibria, and hence design objective values.

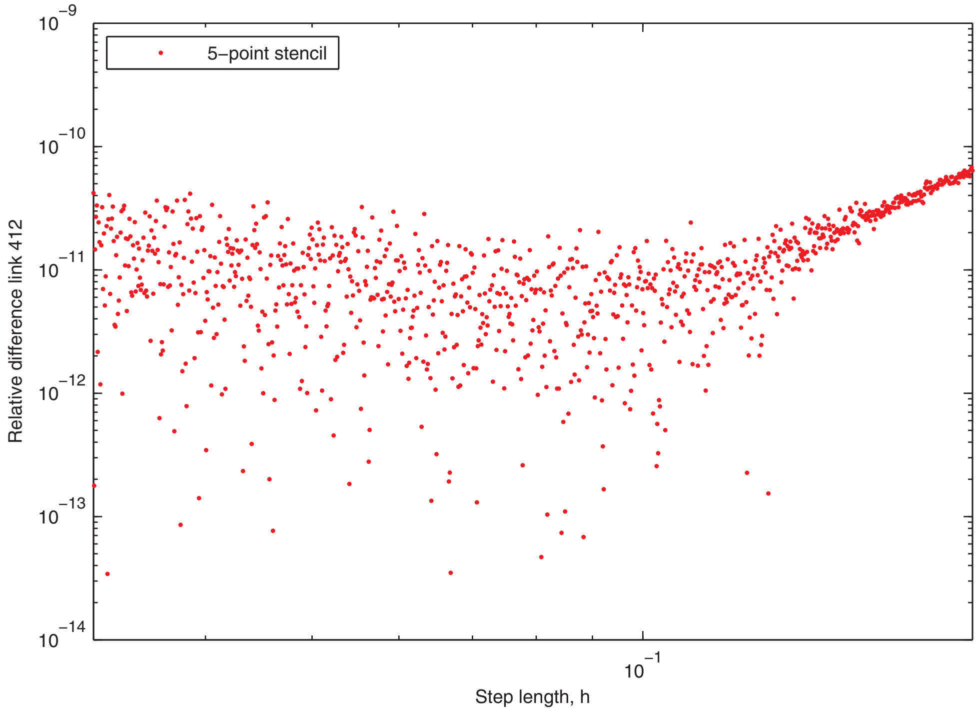
Our main plan for the future is to numerically analyze the new approach on much larger networks. We also aim to extend the analysis and methodology to a framework which includes an elastic demand traffic equilibrium model. While the sensitivity analysis of traffic equilibria in the elastic demand case is already in place (see Qiu & Magnanti (1989); Patriksson & Rockafellar (2003); Patriksson (2004)), it is the subject of future study to extend in the best possible manner the effective utilization of TAPAS, or other efficient methodologies for fixed demand traffic equilibrium and sensitivity computations, to the elastic demand case.

Acknowledgments

The authors wish to acknowledge the many pertinent remarks made by four referees as well as by prof. David Boyce.



(a) Overview plot, 30 step lengths (connected). The range of the h-axis is $\left[10^{-6}, 10^{-0.7}\right]$.



(b) Zoom in at south east corner, 1000 step lengths (dots). The range of the h-axis is $\left[10^{-1.5}, 10^{-0.7}\right]$.

Fig. 4: Relative difference between sensitivity analysis based derivative and finite difference derivative of flow on link 412 at the point $x^* = 0$ for different finite difference step lengths. Experiment is described in Section 6.1.2.

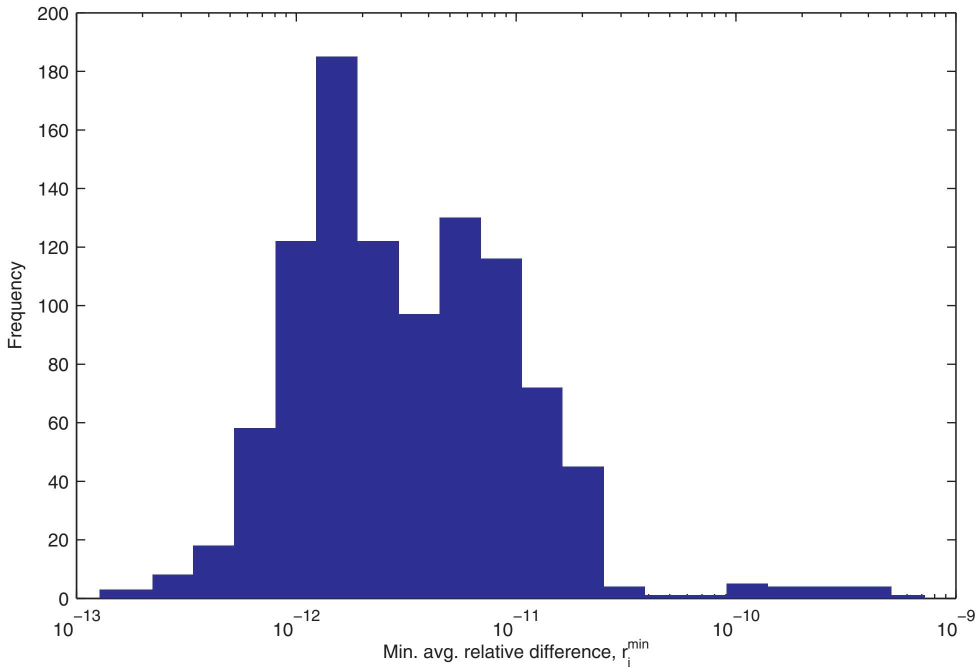


Fig. 5: Histogram of $r_i^{\min}$ for the 1000 points selected in the experiment in Section 6.1.3.

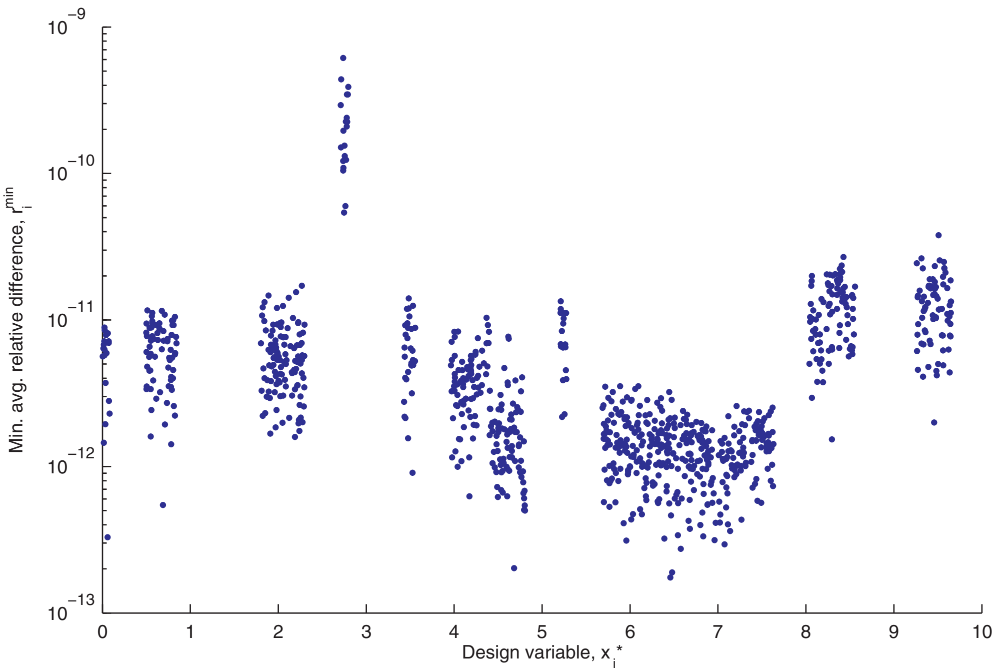


Fig. 6: Scatter plot of $r_i^{\min}$ versus x_i^* for the 1000 points selected in the experiment in Section 6.1.3.

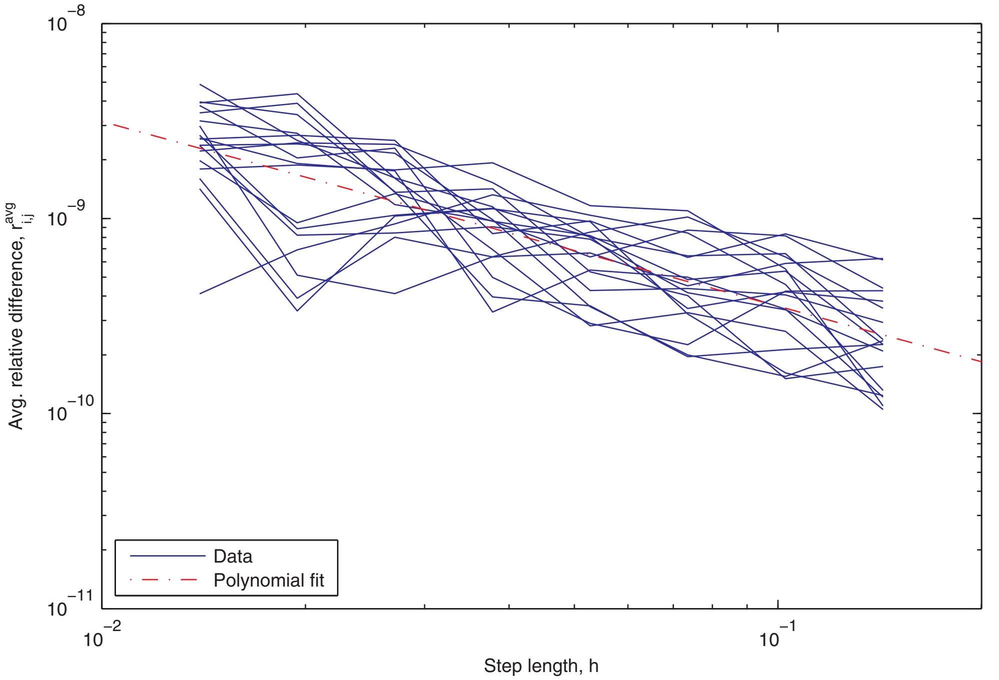


Fig. 7: Average relative differences for 18 points showing large relative differences in the experiment in Section 6.1.3. The slope of the fitted line is −0.9466 and indicates that the error stems from the finite difference approximation.

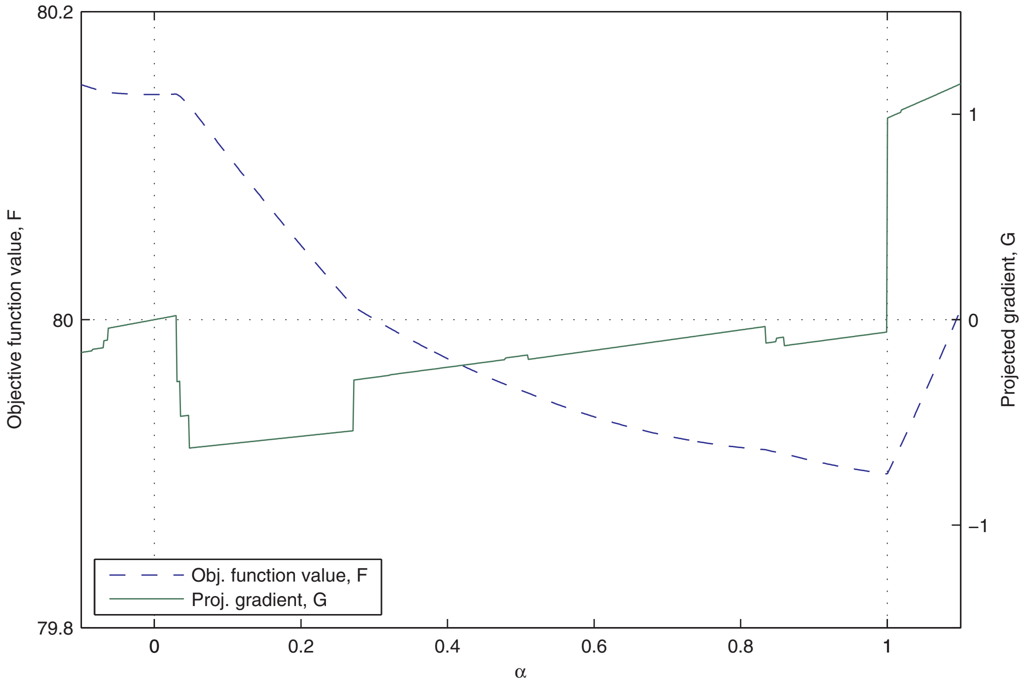


Fig. 8: Objective function and directional derivative along a line between two solutions from experiment in Section 6.2.

References

Babonneau, B., Du Merle, O., & Vial, J. P. 2006. Solving large scale linear multicommodity flow problems with an active set strategy and proximal-ACCPM. *Operations Research*, **54**, 184–197.

Bar-Gera, H. 2001. *Transportation Network Test Problems*. Web page at http://www.bgu.ac.il/~bargera/tntp/, accessed July 10, 2012.

Bar-Gera, H. 2010. Traffic assignment by paired alternative segments. *Transportation Research*, **44B**, 1022–1046.

Boyce, D. E., Ralevich-Dekic, B., & Bar-Gera, H. 2004. Convergence of traffic assignment: how much is enough? *Journal of Transportation Engineering*, **130**, 49–55.

Braess, D. 1968. Über ein Paradoxen aus der Verkehrsplanung. *Unternehmensforschung*, **12**, 258–268.

Chiou, S. W. 2008. A non-smooth model for signalized road network design problems. *Applied Mathematical Modelling*, **32**, 1179–1190.

Choi, S.-C. T., Paige, C. C., & Saunders, M. A. 2011. MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems. *SIAM Journal of Scientific Computing*, **33**, 1810–1836.

Dial, R. B. 2006. A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research*, **40B**, 917–936.

Ekström, J., Engelson, L., & Rydergren, C. 2009. Heuristic algorithms for a second-best congestion pricing problem. *Netnomics*, **10**, 85–102.

Friesz, T. L., Tobin, R. L., Cho, H.-J., & Metha, N. J. 1990. Sensitivity analysis based heuristic algorithms for mathematical programs with variational inequality constraints. *Mathematical Programming*, **48**, 265–284.

Gill, P. E., Murray, W., & Saunders, M. A. 2005. SNOPT: An SQP Algorithm for large-scale constrained optimization. *SIAM Review*, **47**, 99–131.

Hellman, F. 2010. *Towards the solution of large-scale and stochastic traffic network design problems*. M.Phil. thesis, Department of Mathematical Sciences, Chalmers University of Technology, Gothenburg, Sweden.

Josefsson, M., & Patriksson, M. 2007. Sensitivity analysis of separable traffic equilibria, with application to bilevel optimization in network design. *Transportation Research*, **41B**, 4–31.

Kristoffersson, I. 2011. *Congestion Charging in Urban Networks: Modelling Issues and Simulated Effects*. Ph.D. thesis, Department of Transport Science, Traffic and Logistics, Royal Institute of Technology, Stockholm, Sweden.

Lim, A. C. 2002. *Transportation Network Design Problems: An MPEC Approach.* Ph.D. thesis, Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD, USA.

Lukŭ, L., & Vlček, J. 2001. Algorithm 811: NDA: Algorithms for nondifferentiable optimization. *ACM Transcations on Mathematical Software*, **27**, 193–213.

Luo, Z.-Q., Pang, J.-S., & Ralph, D. 1996. *Mathematical Programs with Equilibrium Constraints*. Cambridge, UK: Cambridge University Press.

Marcotte, P. 1986. Network design problem with congestion effects: A case of bilevel programming. *Mathematical Programming*, **34**, 142–162.

Paige, C. C., & Saunders, M. A. 1975. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, **12**, 617–629.

Patriksson, M. 2004. Sensitivity analysis of traffic equilibria. *Transportation Science*, **38**, 258–281.

Patriksson, M., & Rockafellar, R. T. 2003. Sensitivity analysis of variational inequalities over aggregated polyhedra, with application to traffic equilibria. *Transportation Science*, **37**, 56–68.

Qiu, Y. P., & Magnanti, T. L. 1989. Sensitivity analysis for variational inequalities defined on polyhedral sets. *Mathematics of Operations Research*, **14**, 410–432.

Slavin, H., Brandon, J., Rabinowicz, A., & Sundaram, S. 2009. Application of accelerated user equilibrium traffic assignment to regional planning models. *In: Paper presented at the 12th Transportation Research Board National Transportation Planning Applications Conference, Houston, TX, USA*.

Suh, S., & Kim, T. J. 1992. Solving nonlinear bilevel programming models of the equilibrium network design problem: A comparative review. *Pages 203–218 of:* Anandalingam, G., & Friesz, T. L. (eds), *Hierarchical Optimization*. Annals of Operations Research, vol. 34. Basel, Switzerland: J. C. Baltzer AG.

Suwansirikul, C., Friesz, T. L., & Tobin, R. L. 1987. Equilibrium decomposed optimization: A heuristic for the continuous equilibrium network design problem. *Transportation Science*, **21**, 254–260.

Vitins, B. J., Schüssler, N., & Axhausen, K. W. 2012. Comparison of hierarchical network design shape grammars for roads and intersections. *In: paper presented at the 91st Annual Meeting of the Transportation Research Board, Washington, D.C.*

Wardrop, J. G. 1952. Some theoretical aspects of road traffic research. *Institution of Civil Engineers Proceedings, Part II*, **1**(2), 325–378.