Sensitivity analysis of the variable demand probit stochastic user equilibrium with multiple user-classes

Richard D. Connors *, Agachai Sumalee 1, David P. Watling

Institute for Transport Studies, University of Leeds, 36-40 University Road, Leeds LS2 9JT, UK

Received 2 February 2006; received in revised form 13 November 2006; accepted 13 November 2006

Abstract

This paper presents a formulation of the multiple user class, variable demand, probit stochastic user equilibrium model (SUE). Sufficient conditions are stated for differentiability of the equilibrium flows of this model. This justifies the derivation of sensitivity expressions for the equilibrium flows. This paper then considers the network design problem (NDP), assuming that users’ responses to changes in the design variables follow the probit SUE. With probit SUE stated as a fixed-point condition, the NDP is a mathematical program with equilibrium constraints (MPEC). The probit SUE sensitivity expressions provide the information necessary to adopt a gradient-based approach to solving the probit SUE NDP. Numerical examples verify the sensitivity expressions, and the NDP solution.

Keywords: Sensitivity analysis; Probit stochastic user equilibrium; Network design problem; Optimal toll

1. Introduction

In many network-based techniques for transport planning, design and estimation, there is a key role played by the implicit relationship between the data input to a traffic assignment model and the predictions of equilibrium network flows based on those data. For example, in equilibrium-based trip matrix estimation, the ‘data’ in question are the unknown trip matrix elements, which when assigned according to an equilibrium model are required to give predicted link flows that reproduce—to some given level, according to some distance metric—flows that have actually been observed on a subset of links (Yang et al., 1992). A second common application arises in the equilibrium-based network design problem, the ‘data’ reflecting some policy measure under the control of the planner (e.g. link capacities, road tolls, signal timings), where the objective is to optimize some measure of system/economic performance while anticipating the equilibrium response of travellers on the network (Magnanti and Wong, 1984; Patriksson and Rockafellar, 2002; Yang and Bell, 1992).
A third application is in the field of network reliability assessment, where the vulnerability of origin–destination/system performance to unreliable capacity or demand conditions can be imputed from the impact such input changes may have on the equilibrium state (Bell et al., 1999; Chen et al., 2002; Du and Nicholson, 1997). A final application is in the area of error estimation, where the impact of sampling errors in the estimated input data (e.g. trip matrix elements, parameters of the link travel time functions) on errors in the forecast network evaluation measures can be deduced from the implicit equilibrium relationship (Bell and Iida, 1997; Leurent, 1998).

In such applications, it is commonly necessary to deal with the implicit equilibrium relationship (mapping) as a sub-problem during the course of some overall, master solution algorithm, and it is therefore only natural to consider ways of either approximating this relationship and/or of computing its gradients or sub-gradients, should they exist. This is the role of sensitivity analysis, a technique with a substantial history both in non-linear programming generally and in the transportation network field specifically. As illustration of its significance, all of the references cited in the paragraph above were chosen not only to illustrate the application areas in which implicit equilibrium problems arise, but also because they all propose algorithms based on sensitivity analysis. Aside from its importance for applied problems, the prominence of sensitivity analysis has been magnified in recent years from a technical perspective, due to the work of Patriksson and Rockafellar (2002, 2003), who brought into question the whole basis and validity of the seminal transportation paper by Tobin and Friesz (1988) upon which many of the subsequent applications were based. The technical problems and their ramifications continue to be debated (Robinson, 2006), yet it is important to appreciate that these are problem-specific in the sense that they relate crucially to the choice of Wardrop deterministic user equilibrium (DUE) as the network flow model. While these difficulties may be described in a number of different guises, on a simple level there are two main facets of this model that require careful handling. These are namely: (i) non-uniqueness of the equilibrium path flows in general networks even for fixed input data; and (ii) problems of ‘complementarity’ due to the active equilibrium path set (even if it were unique!) changing as the input data to the model are changed, the latter meaning that even directional derivatives of the equilibrium flows may not exist at certain points.

The nature of these difficulties has meant that it has been only natural to consider embedding alternative network flow models in the application problems mentioned above. Specifically, Davis (1994) and Ying and Miyagi (2001) describe the computation of sensitivity analysis for the logit SUE model, which may be observed to exist everywhere (and be efficiently computable) under mild conditions. The disadvantage of this approach is that one is then left with a question of plausibility of the adopted model, in the light of the well-known deficiencies of the logit model in being unable to represent correlated alternatives, of which the routes in a network are one of the most natural examples. The SUE approach is, however, sufficiently general to admit a range of alternative behavioural assumptions, through the form of joint distribution assumed for the stochastic path error terms. Examples of such models include the C-logit (Cascetta et al., 1996), nested logit (Gentile and Papola, 2001), cross-nested logit (Vovsha and Bekhor, 1998), paired combinatorial logit (Gliebe et al., 1999; Prashker and Bekhor, 1999), mixed logit (Nielsen et al., 2002), and probit (Daganzo and Sheffi, 1977). To this end, Clark and Watling (2002) describe a computational procedure for sensitivity analysis of the probit SUE. In the probit case, given that the choice proportions are not expressible in closed form but are rather the result of a multidimensional integral, a key practical factor in this latter work is seen to be deducing the sensitivity analysis expressions in such a way that the relevant Jacobian matrices are computable by analytic means, without resort to the vagaries and errors of finite difference approximation.

Part of the analyst’s role is to determine the appropriate balance between generality, accuracy and efficiency when choosing a modelling approach. In the light of the recent work on sensitivity analysis reported above, it is our contention that probit SUE increasingly affords the best compromise for network modelling, particularly within the wider context of network design optimisation problems. Our aim in this paper is to justify this claim by presenting a formulation of the probit SUE that admits individuals of different classes, and allows them to choose not to travel (elastic demand). We derive gradient information for this general formulation of the probit SUE, showing that many of the analytical pitfalls detailed by Patriksson & Rockafellar are avoided, with the equilibrium flows varying smoothly with the design parameters. This may be contrasted with the DUE model, in which the systematically non-smooth variation of the equilibrium flows in the design parameters makes problems such as equilibrium-based network design extremely difficult. Even if one is
careful to follow the techniques described by Patriksson and Rockafellar (2002, 2003), one must still face the prospect that even directional derivatives will not exist at some points in the design space. In any case, it is not difficult to make a case that drivers do not know precisely, nor perceive identically, the travel costs they will experience on any journey, implying that some form of stochastic model would be more appropriate. Furthermore, in the whole family of equilibrium models mentioned above, the probit SUE has a claim to maximum generality in being able to approximate all such models by appropriate choice of the error distribution. These appealing features of the probit SUE are achieved at a cost; computation and analysis of the probit SUE flows are comparatively difficult and time consuming. Part of the purpose of this paper is to show that these obstacles are diminishing.

Specifically, in the present paper, our original contributions are to extend the work of Clark and Watling (2002) in several key ways:

- The underlying probit SUE model is generalised from the single user class, fixed demand case to a case with multiple user classes and elastic demand.
- A formal proof is provided of the existence of the sensitivity analysis for this model.
- An improved computational procedure is described (even for the single user class, fixed demand case) for computing the base equilibrium solution and the choice probability Jacobian, which both improves the efficiency of the method and avoids the difficulties in interpreting Monte Carlo error (Monte Carlo techniques are not used).
- Explicit formulae are presented to allow the straightforward implementation of the method in widely available matrix-based mathematical languages, such as MATLAB.
- Practical applications of the methods for solving toll-pricing NDPs are reported for realistic-sized networks.

The structure of the paper is as follows. In Section 2 the necessary notation is introduced, and our particular formulation of elastic demand probit SUE presented. Differentiability of the equilibrium flows is established in Section 3, providing conditions to ensure existence of the sensitivity analysis. In Section 4 sensitivity expressions are derived for the equilibrium flows. Implementation issues and the network design problem are discussed in Section 5 with numerical experiments reported in Section 6, before presenting the conclusions in Section 7.

2. Definitions, notation and assumptions

We represent the road network by a graph, with directed links labelled \( a = 1, \ldots, N \) connecting the nodes. The origin-to-destination (OD) movements on the network are labelled \( r = 1, \ldots, R \) and the user classes \( m = 1, \ldots, M \). The \([NM \times 1]\) vector of disaggregate link flows is ordered by class

\[
x = [x_1, \ldots, x^M]^T = [x^1_1, x^1_N, x^2_1, \ldots, x^M_N]^T;
\]

\( x_a^m \) is the flow of class \( m \) on link \( a \). Following Daganzo (1983) we denote the \([N \times 1]\) total standardized link flow vector

\[
z = [z_1, \ldots, z_N]^T = z \cdot x,
\]

where the \([N \times NM]\) matrix \( z \) converts the disaggregate flows into standardized flow of equivalent passenger car units and sums over all user classes on each link. In this paper we will restrict attention to car-only networks with user classes distinguished by their value of time; in (2), \( z = [I_N] \cdots [I_N] \), comprising \( M \) identity matrices \( I_N \), and simply sums the disaggregate flows on each link.

The set of simple paths available to class \( m \) on the \( r \)th OD movement is of size \( K^m_r \); each class may have a different path-set. The total number of paths is \( K = \sum_m \sum_r K^m_r \). An assignment of flows to all paths is denoted by the \([K \times 1]\) vector \( f \), whose elements are ordered by class and sub-ordered by OD movement

\[
f = [f^{1,1}_1, \ldots, f^{1,1}_k, f^{1,2}_1, \ldots, f^{M,R}_k]^T.
\]
with \( f^m_{kr} \geq 0 \ \forall m, r, k \). The \([MR \times 1]\) OD demand vector, \( \mathbf{q} \), is similarly ordered by class and OD movement, with entry \( q^m_r \) representing the total potential travel demand by user-class \( m \) for the \( r \)th OD movement. The path flow assignment \( \mathbf{f} \) is feasible for demand \( \mathbf{q} \) if and only if, for each constituent class and OD movement, \( \sum_{k=K^m} f^m_{kr} = q^m_r \ \forall m, r \). \( \Psi \) is the \([MR \times K]\) demand-path incidence matrix, such that

\[
\mathbf{q} = \Psi \cdot \mathbf{f}.
\]

(4)

The set of feasible path flows is bounded, closed and convex.

The \([MN \times K]\) block-diagonal link path incidence matrix \( A \), whose elements are Kronecker delta functions \( \delta^m_{ak} \), denotes the links comprising each path, for every class and OD pair. The vector of disaggregate link flows, (1), is therefore given by

\[
\mathbf{x} = \begin{bmatrix}
[A^{1,1}; A^{1,2}; \ldots; A^{1,R}] & 0 & \ldots & 0 \\
0 & [A^{2,1}; A^{2,2}; \ldots; A^{2,R}] & \ldots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \ldots & 0 & [A^{M,1}; A^{M,2}; \ldots; A^{M,R}]
\end{bmatrix} \cdot \begin{bmatrix}
f^{1,1} \\
f^{1,2} \\
\vdots \\
f^{1,R} \\
\vdots \\
f^{M,1} \\
f^{M,2} \\
\vdots \\
f^{M,R}
\end{bmatrix} = A \cdot \mathbf{f},
\]

(5)

where \( A^{m,r} \) is the \([N \times K^m_r]\) component link-path incidence matrix for class \( m \) on the \( r \)th OD movement. The standardised link flows are

\[
\mathbf{z} = \mathbf{a} \cdot \mathbf{x} = \mathbf{a} \cdot A \cdot \mathbf{f} = \begin{bmatrix}
A^{1,1}; A^{1,2}; \ldots; A^{1,R}; A^{2,1}; A^{2,2}; \ldots; A^{M,R}
\end{bmatrix} \cdot \mathbf{f}.
\]

(6)

Following Daganzo (1983), the standard link travel time on link \( a \) is denoted \( b_a(\mathbf{z}) \), and users on link \( a \) of class \( m \) experience generalized cost

\[
t^m_a(\mathbf{z}; \tau, \beta) = \tau^m_a + \beta^m b_a(\mathbf{z}).
\]

(7)

The parameter \( \beta^m \) is the value of time for user-class \( m \) and \( \tau^m_a \) is a constant cost (e.g. a toll) specific to link \( a \) and user-class \( m \). We assume that

**A1.** The link cost functions, \( b_a(\mathbf{z}) \), are single valued and continuously differentiable.

**A2.** The Jacobian \( \nabla_a \mathbf{b}(\mathbf{z}) \) is positive definite.

This does not require the link travel time functions to be separable nor that the Jacobian, \( \nabla_a \mathbf{b} \), be symmetric, although A2 does imply that the standard link travel time Jacobian is invertible, and that the standard link travel time functions are monotonic.

The corresponding disaggregate path costs are the summed constituent link costs: \( \mathbf{c} = A^T \cdot \mathbf{t} \) so that

\[
c^m_k(\mathbf{z}; \tau, \beta) = \sum_a t^m_a(\mathbf{z}; \tau, \beta) \delta^m_{ak},
\]

(8)

and hence the disaggregate path cost vector, \( \mathbf{c} \), has entries ordered as for \( \mathbf{f} \). For ease of notation, we will often omit explicit reference to the dependence on tolls and values of time in the link and path cost functions.

### 2.1. The variable demand and choice models

Individuals are motivated to travel by the utility they gain from getting to their desired destination; they choose between travelling to their desired destination on one of the available routes, or not travelling (or going later or by a different mode that is outside of the road network). For each individual, the benefit of travelling, as they perceive it, is weighed against the perceived cost of making their journey, specifically, the (perceived)
minimum route cost. We assume that if the individual perceives no overall gain in utility from making their desired journey, they will choose not to travel.

For each OD pair, the option of “no travel” is represented in the network by a pseudo-link that provides drivers with another choice of OD route; this link thus comprises a pseudo-path. We note that although our terminology is similar, the method we describe should not be confused with the conventional way of implementing elastic demand DUE problems by the excess demand formulation (Gartner, 1980), which requires a separate demand function to be specified and inverted. In our case, the specification of the demand function is integral to the probit model as an additional choice alternative. Without loss of generality, we can assume that for each class and for each OD movement, the flow on the pseudo-path is \( f_k^{m,r} \). With this formulation of variable demand, every driver is assigned to the network; those choosing not to travel are assigned to the pseudo-path connecting the relevant OD pair. We will assume that congestion on a “real” network does not depend on the number of potential travellers staying at home i.e. costs on the “real” links depend only on the flows on the “real” links, and not on the pseudo-link flows.

The motivation to travel is not the same for all users, even within a single class. For users of class \( m \) let the mean utility of travelling (across the population of class \( m \)) on the \( r \)th OD movement be \( v_k^{m,r} \). The initial motivation to make a particular OD movement does not depend on the state of the network; the cost of travel (due to the network flows) determines the net gain in utility and hence whether or not an individual decides to travel. Therefore \( v_k^{m,r} \) is a constant.

Following discrete choice theory we assume that, for users of class \( m \), the perceived utility of the \( r \)th OD movement on route \( k \) is the random variable, dependent on \( z \)

\[
U_k^{m,r}(z) = v_k^{m,r} - c_k^{m,r}(z) + e_k^{m,r},
\]

where \( c_k^{m,r}(z) \) are the deterministic, flow-dependent path costs defined in (8) above. We assume that

A3. The stochastic terms \( e_k^{m,r} \) have a non-degenerate joint probability density function that is continuous, strictly positive, and independent of the deterministic path costs.

We assign the opportunity cost of not travelling (the utility gained by travelling) to the pseudo-link: \( e_1^{m,r} = v_k^{m,r} \forall m, r \), so that the utility of not travelling has zero mean. The constant cost on the pseudo-path also represents the fact that there is no congestion effect on the “no travel” alternative (this assumption can be relaxed and the results to be derived still follow—indeed, the constant cost assumption is more problematic, as seen in Section 3, but is chosen so as to be more consistent with conventional implementations of elastic demand problems). It will be convenient to define the vector of deterministic path utilities, \( u(z) \), with elements \( v_k^{m,r}(z) = v_k^{m,r} - c_k^{m,r}(z) \). This gives \( U_k^{m,r}(z) = u_k^{m,r}(z) + e_k^{m,r} \), and on the pseudo-paths, \( u_1^{m,r}(z) = 0 \forall m, r \).

Drivers from class \( m \), on OD movement \( r \) choosing the \( k \)th path are those who perceive this to maximise their utility (given the current mean path costs for their class: \( e_m \)). The corresponding choice probability is defined to be

\[
P_k^{m,r}(u) = \Pr(U_k^{m,r} \geq U_j^{m,r} \forall j \mid u) .
\]

The \([K \times 1]\) vector of path choice probabilities is denoted \( P \). Note that since \( P(u) = P(v - c) \), then \( V_u P = -V_c P \) and so we can readily work in terms of either utility or cost derivatives. The choice model thus defined is single-valued and continuously differentiable (Gentile and Papola, 2001) in the deterministic path utilities (equivalently in the path costs), and hence (using A1) in the path and link flows.

We assume no inter-class nor inter-OD dependence of the path choice probabilities: \( \partial P_k^{m,r} / \partial u_j^{r,s} = 0 \) unless \( n = m \) and \( r = s \). This assumes that the path choice proportions for a given class on some OD movement, only depend on the path utilities for this class on this OD; they do not rely on the costs experienced by users of other classes, or on other ODs. The impact of other user-classes and OD flows occurs through the path utilities’ dependence, \( u^{m,r}(z) \), on the flows of all user classes on all links. Consequently, the path-choice probability Jacobian \( V_u P \) has a block diagonal structure by class, within which it is block diagonal by OD movement.
The variable demand model presented here is a natural extension of the fixed demand case (with no pseudo-links) and is based on the same underlying principles of discrete choice theory used to model drivers’ route choice behaviour. A demand function is implicitly defined by the choice model; for the \( m \)th class on the \( r \)th OD movement the number choosing to travel, with link flows \( z \), is

\[
q_{r \text{travel}}(z) = q_{r}^{m} (1 - \Pr[U_{r}^{m} \geq U_{r}^{m} (z) \forall k = 1, \ldots, K^{m,r}]).
\]

In this formulation of variable demand, the demand function is determined by an integrated demand and path-choice probability function, and a different demand function prototype cannot be freely chosen (in contrast, for example, with the conventional manner of defining elastic demand DUE problems). While for a given joint probability distribution the demand function is of a fixed type, the variance–covariance matrix allows the cross elasticities to be defined, in particular for the no travel option.

2.2. The stochastic user equilibrium

The Stochastic User Equilibrium (SUE) is defined to be a feasible set of flows such that:

At SUE, no driver can improve his or her perceived utility by unilaterally changing route.

An SUE, \( \hat{z} \), is a solution to the fixed-point problem (see Sheffi, 1985)

\[
\hat{z} = z \cdot A \cdot q \cdot P(u(\hat{z}))
\]

for feasible flows, path choice probabilities, and OD demands. Here \( q = \text{diag}(\nu^{T}q) \) is the matrix-expanded version of the potential demand vector, such that each constituent class-specific OD movement comprises

\[
\begin{bmatrix}
    f_{1}^{m,r} \\
    f_{2}^{m,r} \\
    \vdots \\
    f_{K^{m,r}}^{m,r}
\end{bmatrix}
= \begin{bmatrix}
    q_{r}^{m} & 0 & \cdots & 0 \\
    0 & q_{r}^{m} & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & q_{r}^{m}
\end{bmatrix}
\begin{bmatrix}
    P_{1}^{m,r} \\
    P_{2}^{m,r} \\
    \vdots \\
    P_{K^{m,r}}^{m,r}
\end{bmatrix}
\]

Note that we choose to write the equilibrium condition in terms of the aggregate link flows, \( z \). Due to the properties of the random utility model, by assumption (A1) and following Rosa (2003), the SUE can also be written as an equivalent fixed-point problem in terms of the disaggregate link or path flows: at equilibrium, there is a one-to-one mapping between corresponding aggregate and disaggregate path and link flows. The aggregate link flow SUE, \( \hat{z} \), gives disaggregate link costs as in (7), \( t(\hat{z}) \), and corresponding disaggregate path costs, \( c(\hat{z}) = A^{T} \cdot t \). The choice model then uniquely determines the disaggregate path choice proportions \( \hat{P} = P(u(\hat{z})) \) and hence the disaggregate path flows, \( \hat{f} = q \cdot \hat{P} \), and link flows, \( \hat{x} = A \cdot \hat{f} \), that are consistent with the aggregate link flows \( \hat{z} = z \cdot \hat{x} \).

For the case of UE there would not be a unique mapping between the aggregate link flows, \( z \), and the disaggregate link flows \( x \). It is a property of SUE that gives this one-to-one correspondence.

It remains to solve the fixed-point problem (12) in order to calculate the equilibrium flows for a given network. Fortunately, with this variable demand model, it is no more difficult than calculating the fixed demand SUE flows, methods for which have been extensively discussed elsewhere (Fisk, 1980; Maher and Hughes, 1997; Rosa, 2003; Sheffi, 1985; Sheffi and Powell, 1981).
2.3. The probit model

The probit model is a particular instance of the formulation described above, in which for each user-class \( m \) and O-D movement \( r \) the joint distribution of the vector of path error terms \( e^{m,r} \), which has elements \( e_{k}^{m,r} \) for \( k = 1, 2, \ldots, K^{m,r} \), follows a Multivariate Normal distribution with zero mean and covariance matrix \( \Sigma^{m,r} \). Many structures are possible for this covariance matrix; for example, a diagonal matrix with identical diagonal entries would allow us to approximate the i.i.d. assumption of the multinomial logit. Alternatively, as the magnitude of the terms in \( \Sigma^{m,r} \) becomes small, it is well known that the SUE model increasingly approximates the DUE. As a further alternative, Yai et al. (1997) propose a structure incorporating path-specific error terms. However, the most commonly adopted, and arguably most natural, assumption is to impute \( \Sigma^{m,r} \) from constituent link cost components, with the joint distribution of the link cost error components itself Multivariate Normal. Given that the Multivariate Normal assumption is preserved under linear transformation (from the link to path cost domain), such link error component models do indeed imply a probit path choice model, provided that the implied path cost error covariance matrix is well-conditioned (an important point to which we return in Section 5). While the link error component assumption also permits link error correlations to be specified, a simplification is to neglect these correlations and assume independent Normal link cost error distributions for each link. In this latter, simplest case, if the link cost error distribution for link \( a \) as perceived by user-class \( m \) on O-D movement \( r \) is Normal with zero mean and variance \( \sigma_{amr}^2 \), then the network structure gives the required components of \( \Sigma^{m,r} \) as

\[
\sum_{k,l} = \sum_{a=1}^{N} \delta_{a,k} \delta_{a,l} \sigma_{amr}^2 \quad (k = 1, 2, \ldots, K^{m,r}; l = 1, 2, \ldots, K^{m,r}).
\]

Hence the joint distribution of error terms \( e^{m,r} \) is known in order to apply (10), and which (it is supposed) satisfies assumption A3. Note that since the perceived utilities of the travel alternatives (9) are MVN distributed, the path choice probabilities, (10), and hence the demand function, (11), cannot be written in closed form.

2.4. Properties of the SUE flows

Existence of a solution to the SUE problem defined is guaranteed by Brouwer’s theorem since, by A1 and A3, the fixed-point condition (12) is a continuous function of the flows, being a continuous composition of continuous mappings, and the feasible region is closed and convex (in \( \mathbb{R}^n \), hence bounded). A sufficient condition for the uniqueness of this solution (Cantarella, 1997) is that the link cost function, \( b(z) \), is monotone, non-decreasing, which is given by A2.

2.5. Network design parameters

We wish to consider changes to design parameters, \( s = [s_1, \ldots, s_S]^T \), that define the network; namely, the OD demands and the parameters in the link cost functions. Davis (1994) remarked that

\[ \ldots \text{“if the link use probabilities can be expressed as differentiable functions of the capacity increases and of the link volumes [the design parameters], which is the case for both logit and probit models, the NDP with SUE assignment becomes a differentiable optimization problem with a manageable number of differentiable constraints.”} \]

This implies that the probit SUE flows are differentiable in the design parameters, though Davis provides no formal proof of this statement. In the next section, we determine sufficient conditions for the equilibrium flows to be differentiable in the design parameters, in the case of our variable demand multiple user class network model. Differentiability of the equilibrium flows is required to justify the concomitant derivation of sensitivity expressions for these flows as the design parameters are perturbed.

3. Differentiability of the equilibrium flows

In this section, we establish that the disaggregate equilibrium link flows, \( \hat{x}(s) \), vary smoothly as the design parameters, \( s \), are changed. We shall achieve this in several steps:
• First, we introduce a gap function for the SUE problem, written only in terms of the standardised (aggregate) link flows.
• This gap function formulation is subsequently modified to be written entirely in a new set of variables (the reduced standardised link flows), obtained by eliminating those components of the standardised link flows that correspond to the pseudo-links.
• It is then established that the Jacobian of the gap function with respect to the reduced standardised link flows exists and is invertible.
• This thus allows the implicit function theorem to be invoked, establishing that the reduced standardised link flows are differentiable in the design parameters.
• Finally, it is shown that this is sufficient to guarantee differentiability of the disaggregate equilibrium link flows in the design parameters.

Before proceeding, it is worth noting that there is a particular reason for seeking to establish differentiability in this way. It may seem that a more straightforward approach would be to first introduce an SUE gap function explicitly in the disaggregate link flows, such that the implicit function theorem may be directly applied to these flows. The problem with such an approach is that a key sufficient condition used in the proof below (namely, assumption A2) does not transfer analogously to the space of disaggregate link costs written as a function of disaggregate link flows; under the model assumed, the Jacobian of disaggregate link costs with respect to flows on the pseudo-links. The pseudo links are not deleted and their flows are not set to zero. This approach forces the introduction of more notation, however it is necessary in order to establish differentiability for this formulation of variable demand.

Thus, consider \( \mathbf{d}(\mathbf{z}; \mathbf{s}) = \mathbf{d}(z_1, \ldots, z_N, s_1, \ldots, s_S) \), a continuously differentiable, vector valued function mapping on an open set \( E \subseteq \mathbb{R}^{N+5} \) into \( \mathbb{R}^N \). Let \((\mathbf{z}_0, \mathbf{s}_0)\) be a point in \( E \) for which \( \mathbf{d}(\mathbf{z}_0, \mathbf{s}_0) = 0 \) and for which the \([N \times N]\) Jacobian determinant, evaluated at this point, is non-zero: \( \nabla_\mathbf{z} \mathbf{d} |_{(\mathbf{z}_0, \mathbf{s}_0)} \neq 0 \). Then there exists an \( S \)-dimensional neighbourhood \( \mathcal{W} \) of \( \mathbf{s}_0 \) and a unique, continuously differentiable function \( \mathbf{g} : \mathcal{W} \rightarrow \mathbb{R}^N \) such that \( \mathbf{g}(\mathbf{s}_0) = \mathbf{z}_0 \) and \( \mathbf{d}(\mathbf{g}(\mathbf{t}); \mathbf{t}) = 0 \) for all \( \mathbf{t} \in \mathcal{W} \).

In the light of the SUE fixed-point condition (12), it is natural to consider the gap function for the standardised link flows

\[
\mathbf{d}(\mathbf{z}; \mathbf{s}) = \mathbf{z} - \mathbf{z} \cdot \mathbf{A} \cdot \mathbf{q}(\mathbf{s}) \cdot \mathbf{P}[\mathbf{u}(\mathbf{z}; \mathbf{s})].
\]  

For design parameters, \( \mathbf{s} \), the link flows \( \mathbf{z} \) are a solution to the SUE if and only if \( \mathbf{d}(\mathbf{z}; \mathbf{s}) = 0 \). For every point, \( \mathbf{s}_0 \), where the Jacobian is non-singular, \( |\nabla_\mathbf{z} \mathbf{d}(\mathbf{z}(\mathbf{s}_0); \mathbf{s}_0)| \neq 0 \), the implicit function theorem therefore states that the equilibrium flows are continuously differentiable as a function of the design parameters (in some neighbourhood of \( \mathbf{s}_0 \)). Note that, by assumptions (A1) and (A3), this gap function is differentiable with respect to the standardised link flows, giving

\[
\nabla_\mathbf{z} \mathbf{d} = I - \mathbf{z} \cdot \mathbf{A} \cdot \mathbf{q}(\mathbf{s}) \cdot \nabla_\mathbf{u} \mathbf{P} \cdot \nabla_\mathbf{u} \mathbf{q}(\mathbf{z}; \mathbf{s}).
\]

However, it is difficult to guarantee that the Jacobian in (16) is invertible; in particular, the constant cost pseudo-paths arising from our variable demand model contribute zero rows to \( \nabla_\mathbf{u} \mathbf{u} \), which is therefore rank deficient and hence non-invertible.

To proceed, we consider a reduced formulation of the network flows, by removing variables corresponding to flows on the pseudo-links. The pseudo links are not deleted and their flows are not set to zero. This approach forces the introduction of more notation, however it is necessary in order to establish differentiability for this formulation of variable demand.

Let \( \mathbf{A}^{m,r} \) be the \([N - R \times R^{m,r}]\) reduced link-path incidence matrix for class \( m \) on the \( r \)th OD movement; this is \( \mathbf{A}^{m,r} \) with rows corresponding to pseudo-links removed. The reduced link-path incidence matrix, \( \mathbf{A}^- \), constructed from the \( \mathbf{A}^{m,r} \) structured as in (5), maps the full vector of disaggregate path flows to the vector of disaggregate real link flows. Let \( \mathbf{x}^- \) be the \([N - R \times M(N - R)]\) matrix of concatenated identity matrices: \( \mathbf{x}^- = [I_{N-R} \cdots I_{N-R}] \) such that the vector of reduced standardised link flows is

\[
\mathbf{z}^- = \mathbf{x}^- \mathbf{A}^- \cdot \mathbf{f}.
\]
So $\Delta^{-1} = [\Delta^{-1} \cdots \Delta^{-R}]$ maps the full vector of path flows to the $[N - R \times 1]$ vector of standardised real link flows, wherein the pseudo-links are omitted for each OD. The SUE fixed point condition for the standardised real link flows is

$$z^{-} = \Delta^{-1} \cdot q \cdot P(u(z^{-))). \quad (18)$$

Note that the argument of the choice model is the full disaggregate vector of path utilities, so the correct proportions of flow are assigned to all paths. The full demand matrix is used. Here the full vector of path utilities is reconstructed easily from the vector of real flows, $z^{-}$, since the deterministic utility on all pseudo-paths is zero; in fact for each class and OD movement the full vector of path utilities is given by

$$u^{m,r} = r^{m,r} - \Delta^{m,rT}[\tau^{m} + \beta^{m}b^{-}(z^{-})], \quad (19)$$

here $\tau^{m}$ is the vector of constant link costs $\tau^{m}_{a}$ for the real links and $b^{-}$ is the reduced vector of link cost functions. The fixed-point condition (18) is equivalent to (12); at equilibrium, the flows on the real links uniquely determine the flows on each of the pseudo-paths, via the choice model that itself imposes the demand constraints since the choice probabilities sum to unity.

To establish differentiability of the SUE flows, we apply the implicit function theorem to the gap function

$$d^{-}(z^{-}; s) = z^{-} - \sum_{m=1}^{M} \sum_{r=1}^{R} q^{m,r}(s)\Delta^{m,r} \cdot P^{m,r}[u^{m,r}(z^{-}; s)], \quad (20)$$

$d^{-}(\cdot)$ is single-valued and $C^{1}$ because (by A1 and A3) the link cost functions and the choice model are single-valued and $C^{1}$. To apply the implicit function theorem, it remains to show that the Jacobian Determinant (with respect to $z^{-}$) is invertible

$$\nabla_{z} d^{-}(z^{-}; s) = I_{N - R} - \sum_{m=1}^{M} \sum_{r=1}^{R} q^{m,r}(s)\Delta^{m,r} \cdot \nabla_{z} P^{m,r}[u^{m,r}(z^{-}; s)]. \quad (21)$$

By the chain rule

$$\nabla_{z} d^{-}(z^{-}; s) = I_{N - R} + \sum_{m=1}^{M} \sum_{r=1}^{R} q^{m,r}(s)\Delta^{m,r} \cdot \nabla_{w^{m,r}} P^{m,r} \cdot \nabla_{z} [\Delta^{m,rT}\beta^{m}b^{-}(z^{-}; s)]. \quad (22)$$

As noted above, the matrix $\Delta^{m,r}$ reconstructs the full vector of path utilities from the reduced vector of constituent link costs, with the pseudo-path being constructed with zero mean utility. The additive constants in (7) and (9) do not contribute. Simplifying (22) gives

$$\nabla_{z} d^{-}(z^{-}; s) = I_{N - R} + \sum_{m=1}^{M} \sum_{r=1}^{R} q^{m,r}(s)\Delta^{m,r} \cdot \nabla_{w^{m,r}} P^{m,r} \Delta^{m,rT} \nabla_{z} b^{-}. \quad (23)$$

The choice probability Jacobian $V_{w}P$ and its constituent blocks $\nabla_{w^{m,r}} P^{m,r}$ are positive semi-definite (Daganzo, 1979). Pre-multiplying Eq. (23) by the Jacobian of standardised real link cost functions gives

$$\nabla_{z} b^{-T} \nabla_{z} d^{-}(z^{-}; s) = \nabla_{z} b^{-T} + \sum_{m=1}^{M} \sum_{r=1}^{R} q^{m,r}(s)\beta^{m}[\Delta^{m,rT} \nabla_{z} b^{-}]^{T} \cdot \nabla_{w^{m,r}} P^{m,r} \cdot [\Delta^{m,rT} \nabla_{z} b^{-}]T. \quad (24)$$

The first term on the right hand side is positive definite by (A2). The terms inside the double sum are positive semi-definite since each is a quadratic form applied to the path choice probability Jacobian, itself a positive semi-definite matrix. The demands and values of time are assumed positive. The double sum is therefore positive semi-definite as a sum of positive semi-definite matrices. The right hand side is therefore the sum of a positive definite matrix and a positive semi-definite matrix and is hence positive definite. On the left hand side, $\nabla_{z} b^{-T} \nabla_{z} d^{-}$ is therefore positive definite and hence invertible. In addition, $\nabla_{z} b^{-T}$ is invertible (by A2). Finally then, $\nabla_{z} d^{-}$ is invertible, with inverse

$$(\nabla_{z} d^{-})^{-1} = (\nabla_{z} b^{-T} \nabla_{z} d^{-})^{-1} \nabla_{z} b^{-T}. \quad (24)$$

The implicit function theorem can therefore be applied to the gap function (20).
Consider a setting of the design parameters $s = s_0$ and the corresponding SUE reduced link flow vector, $\tilde{z}^-(s_0)$. There is an open neighbourhood $W$ of $s_0$ and a continuously differentiable function $g$ such that $g(s_0) = \tilde{z}^-_0$ and $g(s) = \tilde{z}^-$ for all $s \in W$, that is $d^- (g(s),s) = 0$. The reduced standardised equilibrium link flows are a continuously differentiable function of the design parameters.

It follows that the full vector of disaggregate SUE link flows is also a continuously differentiable function of the design parameters, since the disaggregate link flows are uniquely specified by a continuously differentiable function of the standardised real link flows, namely

$$\tilde{x} = A^{-1} \cdot q \cdot P(\Psi^T v - A^{-T} [\tau + \beta \cdot \tilde{b}^-(\tilde{z}^-)])$$

(25)

where $A^{-}$ is the $[M(N - R) \times K]$ reduced link path incidence matrix introduced above, and the vector $\tau$ and the matrix $\beta$ follow from the definitions in (7). This equation highlights the key role of the choice model in uniquely specifying the disaggregate flows from the (reduced) aggregate flows at equilibrium; this one-to-one correspondence does not occur in the case of UE.

4. Link based sensitivity analysis of the equilibrium flows

Standard optimisation algorithms need gradient information to find minima/maxima. For the SUE flows, computation of the relevant Jacobian matrices by numerical differencing requires many SUE evaluations and is prone to error due to the amplification of inaccuracies in the calculation of the SUE flows (see Connors et al., 2003). A sensitivity analysis of the SUE flows provides analytical expressions for the Jacobians of link flows that, in turn, are required to derive Jacobian matrices describing the gradients of any objective function in a network design optimisation (see Magnanti and Wong, 1984).

Sensitivity analysis for the elastic demand UE case can be found in Yang (1997) and Josefsson and Patriksson (2007), for logit SUE in Davis (1994) and for single user class, fixed demand probit SUE in Clark and Watling (2002). In this section, we calculate the sensitivity expressions for the multiple user-class variable demand probit SUE disaggregate link flows.

An expression for the gradient of the reduced standardised link flows can be obtained via the Taylor series expansion of the gap function (20),

$$d^- (\tilde{z}^-(s); s) \approx d^- (\tilde{z}^-(s_0); s_0) + \nabla_x d^- \cdot [\tilde{z}^- (s) - \tilde{z}^- (s_0)] + \nabla_x d^- \cdot [s - s_0].$$

The Jacobians are evaluated at $\tilde{z}^-(s_0)$. This approximation is accurate in the neighbourhood of $s_0$ where the neglected (curvature) terms in the Taylor expansion of $d^- (\tilde{z}^-; s)$ remain small. This approximation should be expected to break down where the variation in flows displays high curvature. Since the gap function is zero at equilibrium, at both $\tilde{z}^-(s_0)$ and $\tilde{z}^-(s)$, this can be rearranged to give

$$\tilde{z}^-(s) \approx \tilde{z}^-(s_0) - (\nabla_x d^-)^{-1} (\nabla_x d^-) [s - s_0],$$

(26)

and hence the gradient of the reduced standardised equilibrium link flows is

$$\nabla_x \tilde{z}^-(s) \approx - (\nabla_x d^-)^{-1} (\nabla_x d^-).$$

(27)

For convenience (and to reduce the matrix sizes for numerical computation), we order the design parameters into various types: demand, class specific fixed link costs (tolls), value of time parameters and parameters in the standard link cost functions, so that $s = [s_q, s_r, s_p, s_b]$, so that

$$\nabla_s d^- \cdot (s - s_0) = \nabla_s d^- \cdot (s_q - s_{q0}) + \nabla_s d^- \cdot (s_r - s_{r0}) + \nabla_s d^- \cdot (s_p - s_{p0}) + \nabla_s d^- \cdot (s_b - s_{b0}).$$

It follows, from (25), that the disaggregate flows for class $m$ can be obtained from any vector of reduced standardised link flows as follows:

$$x^m (\tilde{z}^-, s) = \sum_{r=1}^R q^m (s) A^m r \cdot P^m r (\Psi^T v - A^m r^T [\tau^m (s) + \beta^m (s) \cdot \tilde{b}^- (\tilde{z}^-; s)])$$

(28)
Evaluating this formula at equilibrium allows us to derive the gradient of the disaggregate equilibrium link flows

$$\nabla_s \tilde{x} = \begin{bmatrix} \vdots \\ \nabla_s x^m(\tilde{z}(s), s) = \nabla_s x^m(\tilde{z}, s) + \nabla_{\tilde{z}} x^m(\tilde{z}, s) \nabla_s \tilde{z}(s). \end{bmatrix}$$  \hspace{1cm} (29)

Eq. (27) gives an expression for $\nabla_s \tilde{z}$. The form of constituent Jacobians $${\phi^m}(M) = \sum_{r=1}^{R} q^m \Delta^m r \cdot \nabla_{\tilde{w}} P^m r \cdot \Delta^m r^T \cdot M,$$  \hspace{1cm} (30)

$$\phi^m(M) = \sum_{r=1}^{R} q^m \Delta^m r \cdot \nabla_{\tilde{w}} P^m r \cdot \Delta^m r^T \cdot M.$$  \hspace{1cm} (31)

Note that $\tilde{\phi}^m(M)$ has $N-R$ rows whereas $\phi^m(M)$ has $N$ rows.

4.1. Link flow Jacobians

The Jacobian of the gap function with respect to the reduced standardised link flows is then

$$\nabla_s d^-(\tilde{z}; s) = I_{N+R} + \sum_{m} \beta^m \tilde{\phi}^m(\nabla_{\tilde{z}} b^-).$$  \hspace{1cm} (32)

For the disaggregate equilibrium link flows we get similarly,

$$\nabla_{\tilde{z}} x^m(\tilde{z}, s) = -\beta^m \phi^m(\nabla_{\tilde{z}} b^-).$$  \hspace{1cm} (33)

4.2. Demand Jacobian

On differentiating with respect to parameters changing the OD demands, $s_q = [s_1, \ldots, s_L]$, the Jacobian of the reduced gap function is

$$\nabla_s d^- = -\sum_{m} \sum_{r} A^m r \cdot \begin{bmatrix} P^m r \\ P^m r \\ \vdots \\ P^m r \end{bmatrix} \cdot \nabla_s q^m r.$$  \hspace{1cm} (34)

The design parameters designate those OD/class demands that are being perturbed and so the last matrix comprises zeros and ones if the design parameters represent only additive perturbations to OD/class demands, wherein the Jacobian has columns $-A^m r P^m r$, for each OD/class designated by the design parameters.

The Jacobian of disaggregate link flows for class $m$ is

$$\nabla_s x^m = \sum_{r=1}^{R} A^m r P^m r \cdot \nabla_s q^m r.$$  \hspace{1cm} (35)

Substituting (34) and (32) into (27) gives $\nabla_s \tilde{z}^-(s)$ and using this along with (35) in (29) gives $\nabla_s \tilde{x}(s)$. Similar substitutions are required to obtain formulae for $\nabla_s \tilde{x}(s)$, for other types of design parameters below.

4.3. Class-specific link constant (link toll) Jacobian

Consider the design parameters that alter the tolls, $s_c = [s_1, \ldots, s_L]$; each design parameter denotes a class and a (real) link on which the toll is being perturbed.
\[
\n\nabla_s \mathbf{d}^- = \sum_m \sum_r q^{m,r} A^{m,r} \cdot \nabla w^{m,r} \cdot P^{m,r} \cdot A^{m,r\top} \nabla_s \mathbf{c}^m_r = \sum_m \bar{\phi}^m(\nabla_s \mathbf{c}^m_r). \tag{36}
\]

Each design parameter \( s_j \) generates a column in the Jacobian corresponding to some class-specific link constant, \( \tau^m_s \). Similarly, \( \nabla_s \mathbf{x}^m = -\phi^m(\nabla_s \mathbf{c}^m_r) \).

### 4.4. Value of time Jacobian

Those design parameters that alter the values of time, \( s_b = [s_1, \ldots, s_L] \), give the Jacobian

\[
\nabla_{s_b} \mathbf{d}^- = \sum_{m=1}^M \sum_{r=1}^R q^{m,r} A^{m,r} \cdot \nabla w^{m,r} \cdot P^{m,r} \cdot A^{m,r\top} \cdot b^-(z^-) \cdot \nabla_{s_b} \beta^m = \sum_m \bar{\phi}^m(b^-(z^-) \cdot \nabla_{s_b} \beta^m). \tag{37}
\]

Each design parameter contributes a column to the Jacobian corresponding to the class whose value of time is being perturbed. In the same way, if a design parameter designates the value of time for class \( m \) the Jacobian of flows for this class has the column \( \nabla_{s_b} \mathbf{x}^m = -\phi^m(b^-(z^-) \cdot \nabla_{s_b} \beta^m) \).

### 4.5. Standard link cost Jacobian

Those design parameters, \( s_h = [s_1, \ldots, s_L] \), that alter parameters in the standard link cost functions, such as capacities, free flow times etc, give the Jacobian

\[
\nabla_{s_h} \mathbf{d}^- = \sum_m \sum_r \beta^m q^{m,r} A^{m,r} \cdot \nabla w^{m,r} \cdot P^{m,r} \cdot A^{m,r\top} \cdot b^- = \sum_m \bar{\phi}^m(\nabla_{s_h} \mathbf{c}^m_r). \tag{38}
\]

For such design parameters, the Jacobian of disaggregate flows for class \( m \) is \( \nabla_{s_h} \mathbf{x}^m = -\beta^m \phi^m(\nabla_{s_h} \mathbf{c}^m_r) \).

### 5. Solving the network design problem

#### 5.1. Computing the SUE flows

Numerical computation of the SUE flows, and their sensitivities to changes in the design parameters, comprises several stages, each of which can be accomplished using a variety of methods. Given current path costs, the probit path choice probabilities need to be calculated. These probabilities are used within an iterative optimisation algorithm to determine the equilibrium flows; at each iteration, a search direction and step length are required. Once the equilibrium flows are known, the Jacobians must be computed to obtain the sensitivity expressions derived in the previous section.

There is no closed form expression for the probit path choice probabilities, which must therefore be computed either by numerical integration (e.g. Genz, 1992), simulation (e.g. Monte Carlo as in Sheffi and Powell, 1981), or by analytic approximation (review in Rosa, 2003). Approximation by simulation (Monte Carlo or otherwise) introduces non-smooth variation in the equilibrium flows (as the design parameters are smoothly varied) due to random sampling of the multivariate probability distribution; estimating the path choice probabilities via analytic approximation does not introduce such artefacts. In this paper, we follow the recommendation of Rosa (2003) and use the Mendell–Elston analytic approximation method (Mendell and Elston, 1974). Monte Carlo simulation is not used anywhere in this paper.

To calculate the equilibrium flows, we use an algorithm based on the ‘traditional’ method of successive averages (MSA), (see Wilde, 1964). We seek to iteratively reduce the objective function of the SUE equivalent optimisation (Sheffi, 1985) using the search direction from the MSA algorithm (moving toward the auxiliary flows). However, rather than using the prescribed, \( 1/n \), step length of MSA at each iteration, we calculate an optimised step length using a quadratic approximation (Maher and Hughes, 1997), augmenting this with a bisecting line search when necessary, to ensure improvement at every iteration.

Once the equilibrium flows have been determined, computation of their sensitivities to changes in the design parameters requires calculation of all the Jacobian matrices detailed in the previous section. One component in
these formulae requires special effort: the Jacobian of path-choice probabilities with respect to path costs, $\nabla_c \mathbf{P}$. A method to compute this Jacobian, which Daganzo (1979) attributes to McFadden, is described in Clark and Watling (2002). The choice probabilities required within the calculation of $\nabla_c \mathbf{P}$ are calculated using the method of Mendell–Elston.

The inverse of the path-covariance matrix appears in the definition of the probit path choice probabilities, and in the calculation of the path-choice probability Jacobian. Although the path-covariance matrix may be singular (see Clark and Watling, 2002) as it is, for example, in the figure-of-eight network, the network equilibrium flows are well-behaved and vary smoothly with the design parameters. Moreover, the probit model, that is to say the multivariate normal probability density function, can accommodate a singular variance–covariance matrix. However, our methods for computing the SUE flows and the sensitivity expressions presented in this paper do run into difficulties when the path-covariance matrix is not invertible. We therefore adopt a mechanism for avoiding this scenario, and construct the path set to avoid any rank deficiencies in the link-path incidence matrix for each OD pair that would result in such a non-invertible path-covariance matrix. In the standard MSA algorithm (Sheffi, 1985), all paths are implicitly available. The active paths are generated incrementally, at each iteration, using auxiliary solutions generated by a stochastic shortest path method. While in an infinite number of iterations this algorithm would generate all conceivable paths, in practice (at the end of a finite number of iterations) the active path set simply comprises those paths generated thus far during the procedure.

The alternative method for calculating probit SUE, as used in this paper, is to define the active path set upfront. The path set is generated heuristically in an attempt to include all paths that carry “significant” quantities of flow at equilibrium: this process begins with inclusion of the pseudo-links as the no-travel paths, and the shortest free flow path for each OD pair. With this initial path set, we iterate as follows: given the current path set, the equilibrium flows and resulting path costs are computed under ten times the normal demand, then new shortest and “non-degenerate” paths are added to the path set (as in the Sheffi MSA). Such iterations continue until no new paths are generated under the inflated demand and the resulting path set is then fixed. This fixed path set is then used throughout the sensitivity and NDP calculations presented below.

5.2. The probit SUE network design problem as an MPEC

If we choose to evaluate the performance of the network at some equilibrium condition, say probit SUE, the NDP naturally presents itself as a mathematical program with equilibrium constraints (MPEC). The mathematical program is an optimisation problem: adjusting the network design parameters (link capacities, tolls etc) in order to maximise the network’s performance as measured by our objective function, while the network flows are constrained to satisfy probit SUE.

With objective function $g(\cdot)$, and network design parameters $\mathbf{s}$, the NDP in terms of the link flows is

$$\max_{\mathbf{s}} g(\mathbf{x}, \mathbf{s}) \quad \text{subject to } \mathbf{x} = \tilde{\mathbf{x}}(\mathbf{s})$$

(39)

For our probit SUE formulation the lower level equilibrium has a unique solution. Solving the NDP typically involves a numerical search across values of the network design parameters to optimise the upper level objective function, whilst evaluating the lower level equilibrium flows at each iteration.

Since the probit SUE flows vary smoothly with the design parameters, with a smooth objective function, $g(\cdot)$, the probit SUE NDP presents a smooth objective function ‘surface’, though typically with multiple optima in a high dimensional space.

The computational effort required to solve the probit-based MPEC is concentrated in certain necessary steps. Although multiple equilibrium assignments may not be needed by all solution strategies, all algorithms will require multiple probit-assignments. Central to each probit assignment is the evaluation of the path choice proportions for every OD movement and every class at given path costs; the number of these evaluations therefore provides a measure of the efficiency of the algorithm. Since the method used to calculate the Jacobian (Clark and Watling, 2002) is somewhat equivalent to a probit assignment, those algorithms using such gradient information will be judged fairly by this metric.
5.3. Implicit programming formulation

Uniqueness of path flows (and hence link flows) and the smoothness of the relationship between the path flows and design variables allow us to reformulate (39) as the implicit function

\[
\max_s \tilde{g}(s),
\]

(40)

where \( \tilde{g}(s) = g(\tilde{x}(s), s) \). The Jacobian for this implicit objective function

\[
\nabla_s \tilde{g} = \nabla_s g + \nabla_s g \cdot \nabla_s \tilde{x}
\]

(41)

can be derived using the sensitivity analysis of the previous section. This Jacobian information allows the application of gradient-based methods to the smooth optimisation problem (40), one example is the Sequential Quadratic Programming (SQP) algorithm.

The basic form of SQP methods were popularized by Han (1977) and Powell (1978). In its basic form, SQP solves a non-linear programming (NLP) problem via a sequence of quadratic programming (QP) approximations obtained by replacing the non-linear constraints by a linear first order Taylor series approximation and the non-linear objective by a second order Taylor series approximation augmented by second order information from the constraints. It is well known that the SQP method may fail to converge if it is started far from a local solution. In order to induce convergence, many popular methods use a penalty function that is a linear combination of the objective function and some measure of the constraint violation. A related idea is an augmented Lagrangian function in which a weighted penalty term is added to a Lagrangian function. A step in an SQP method is then accepted when it produces a sufficient decrease in the penalty function. Two frameworks exist which enforce sufficient decrease, namely line-search in the direction of the QP solution or a trust-region that limits the QP step that is computed. The size of the trust-region is reduced if the step is rejected and increased if it is accepted. The particular SQP algorithm implemented in the following section is documented online (Mathworks, 2006).

Given a vector of design variable \( (s_k) \) at iteration \( k \), the SQP algorithm re-calculates the SUE flows, evaluates the objective function at these new SUE flows, and calculates the Jacobian of the objective function with respect to the design variables. The SQP algorithm then uses this information to determine the predicted optimal design variable \( (s_{k+1}) \) for the next iteration. The SUE flow constraint is incorporated into the objective function leaving only the bounds on the design parameters as constraints.

6. Numerical experiments

6.1. Comparison with UE

To understand the influence of the choice of equilibrium condition (UE or SUE), we begin by examining a very simple NDP, but one that presents multiple optima of the objective function.

Consider the variable demand two-link network with total demand \( Q = 11 \), with a single network design parameter representing a toll, \( \tau \), imposed on Link 1, and link cost functions

\[
C_0(x_0) = c_0 + \epsilon_0 \quad \text{with} \quad \epsilon_0 \sim N(0, \sigma^2),
\]

\[
C_1(x_1) = 10 + \tau + x_1 + \epsilon_1 \quad \text{with} \quad \epsilon_1 \sim N(0, \sigma^2),
\]

\[
C_2(x_2) = 60 + x_2^2 + \epsilon_2 \quad \text{with} \quad \epsilon_2 \sim N(0, \sigma^2).
\]

The demand variation is modelled by a ‘pseudo-link’ with cost \( C_0 \), and \( x_0 \) denoting the number of drivers not travelling. For simplicity, all links have the same variance in their perceived costs. We explore the behaviour of this network, that is to say the SUE link flows, as demand (via \( c_0 \)) and toll (\( \tau \)) are varied. Different values for the variance of the choice probabilities are calculated. Fig. 1 depicts the SUE flows on the network for various settings of \( c_0 \) and toll \( \tau \) and for different variances, including \( \sigma^2 = 0.2 \) which approaches the UE case. When the cost of not travelling is low, all the potential demand is taken by the pseudo-link and no one travels. As the cost of staying at home, \( c_0 \) increases, more people are forced onto the network. At low values of the link 1 toll, drivers choose to travel on link 1, since the free flow cost on link 2 is comparatively high. As link 1 becomes
congested through either more travellers (as $c_0$ increases), or becomes less attractive (as $\tau$ increases), so more traffic moves onto link 2. Note that the transition of flow between the links as the network parameters change occurs smoothly for SUE. This may be contrasted with equivalent UE-like flows that change between routes abruptly as the network parameters are changed. This can be seen as the ‘sharp corners’ of the flows in the figures on the left hand side.

Regardless of the toll level, as the cost of not travelling ($c_0$) increases, more people elect to travel on the network, that is, on links 1 and 2. The toll determines the relative attractiveness of link 1 or link 2, as can be seen in Fig. 1 by the shape of the curves when $\tau = 60, 90$; for these cases the uncongested cost of link 1 is higher than that on link 2, hence the flow is initially assigned to link 2 and only when the congestion on this link increases does link 1 begin to be used. As the variance increases, so the transitions in flow occur smoothly.

In addition to examining the link flows and their dependence on the underlying parameters of the network, other quantities can also be measured. For example, the revenue generated by the toll on link 1 is simply $R = x_1 \tau$. The dependence of revenue on the toll level is shown in Fig. 2, with $c_0 = 100$. For the low variance
case (solid line) the flow on link 1, and hence the revenue, drops to zero when the fixed cost on link 1 (free flow cost plus toll) increases beyond the cost of no travel. Notice that the revenue surface has two local maxima, whereas with increased variance SUE flows there is a single global maximum for the revenue surface. This is not to imply that SUE will always result in a convex objective function. However, this illustrates the non-smoothness of UE and that one aspect of the SUE smoothing can be to remove “sharp” local optima.

To illustrate the application of sensitivity analysis to a network design problem, we consider this two link variable demand network with two user classes. With link cost functions as above, and pseudo-link cost

![Fig. 2. Revenue at fixed demand for SUE with several variances.](image1)

![Fig. 3. Objective function surface for the NDP.](image2)
For the sake of illustration, we consider the revenue earned (tolling each class independently) as the NDP objective function. Evaluating the revenue at equilibrium for each combination of tolls generates the objective function surface in Fig. 3. Solving the NDP by the implicit function method (40) is shown in Fig. 4; numbered points mark the SQP iterations with a triangle marking the final solution, gradients calculated as described in Sections 4 and 5 are listed in Table 1 for each SQP iteration.

Convergence was assumed when the absolute change in the tolls and objective function was less than $10^{-10}$; the norm of the gradient was then $1.5 \times 10^{-4}$. The method therefore successfully finds a local optimum (in this particular case the global optimum) requiring only five SUE assignments.

### 6.2. NDP for the Sioux Falls network (single user class)

The implicit function method described above was applied to the network of Sioux Falls that has 528 OD movements, 604 links (76 road links and one pseudo-link for each OD movement), and a total of 2806 paths. The OD movements and link cost functions are as defined in (Bar-Gera, 2001; Suwansirikul et al., 1987) (see Fig. 5).

A uniform toll was applied on both directions to links joining node pairs (6,8), (7,8), (9,10), (10,16), (13,24).

![NDP Maximising Revenue](image)

Fig. 4. Gradient-based solution of the SUE NDP.

<table>
<thead>
<tr>
<th>SQP iteration</th>
<th>Toll on class 1</th>
<th>Toll on class 2</th>
<th>Revenue</th>
<th>Revenue gradient (increasing class 1 toll)</th>
<th>Revenue gradient (increasing class 2 toll)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9.9976</td>
<td>4.9999</td>
</tr>
<tr>
<td>2</td>
<td>65.0040</td>
<td>92.1300</td>
<td>606.4825</td>
<td>-11.4927</td>
<td>4.8298</td>
</tr>
<tr>
<td>3</td>
<td>54.0993</td>
<td>97.1904</td>
<td>714.8674</td>
<td>-2.37621</td>
<td>0.7832</td>
</tr>
<tr>
<td>4</td>
<td>52.2367</td>
<td>96.9992</td>
<td>717.0622</td>
<td>-0.03444</td>
<td>-0.0083</td>
</tr>
<tr>
<td>5</td>
<td>52.1889</td>
<td>96.9392</td>
<td>717.0638</td>
<td>0.000073</td>
<td>-0.0001</td>
</tr>
</tbody>
</table>

Table 1

Iterations of SQP in solving the network design problem
The result of the implicit function method (Section 5.3) is displayed in Fig. 6; each iteration moves closer to the optimum. After 4 SQP iterations (and 332,456 probit choice evaluations) it gives an optimal toll of 0.442. Note that a probit route choice proportion is calculated (and counted) for each OD movement, so this measure of performance increases rapidly with iterations of the optimisation algorithm.

Fig. 5. The Sioux Falls network map.

Fig. 6. The Sioux Falls NDP.
6.3. Sensitivity analysis of the Headingley network

We consider the Headingley network: 73 nodes, 240 OD movements (hence 240 pseudo-links) and 188 road links, with elastic demand and two user classes that have independent values of time, set to be 5.6 pence/minute and 10.5 p/m (see Department for Transport, 2004). The path set is generated up front as described in the previous section, resulting in 1463 paths.

Appropriate costs for the pseudo-paths are calculated from a fixed demand assignment to the real links. For each OD movement, the pseudo-link cost is set equal to the maximum path cost. The probit variance for each link is set to be a multiple of the free flow time on that link. Since the pseudo-links have constant cost, they tend to have higher “free flow” times than other links in the network and hence are accorded relatively high variances, representing systematically higher variation in the travel/no travel decision than the perceived cost variation between alternative routes. Clearly, any other variance structure could be adopted with this model.

The test network is shown in Fig. 7. Origins and destinations are marked with triangles; the corresponding pseudo-links are not displayed in this figure. One scenario is investigated in this paper: a flat toll is imposed on all links within the marked area of the network. Sensitivity analysis of the equilibrium link flows is calculated at zero toll, and the new equilibrium flows are computed for various toll levels from £5 to +£5.

In Figs. 8 and 9 the solid lines represent the equilibrium state of the network, with the SUE flows recomputed at each toll level. The dashed line is evaluated using the equilibrium flows that are predicted by the sensitivity analysis for this toll level. The approximations derived from the sensitivity expressions are tangential to the “true” (recomputed) equilibrium behaviour at zero toll, where the sensitivity analysis was conducted.
Fig. 8. Variation in demand with toll, by user class.

Fig. 9. Variation in travel time with toll, by user class.

Fig. 10. Variation in total revenue with toll and numbered iterations of the implicit programming solution to NDP.
The sensitivity analysis gives a linear approximation to the SUE flows and hence to the total number travelling on the real network, since this is simply a sum of certain flows. Therefore, the dashed lines in Fig. 8 are straight. However, travel time is not a linear function of flow, so the sensitivity analysis predictions in Fig. 9 are not necessarily straight-line approximations.

The two user classes are distinguished by their value of time, and the toll imposed in this example is the same for all users. Fig. 8 shows that demand for travel decreases with toll for users with a low value of time, removing congestion on the network, provoking an increase in the number of high value of time users choosing to travel. Fig. 9 displays the total travel time by user class, calculated from those travelling on the real links in the network. As the toll increases, there is a decrease in total travel time for low value of time users; this is due to the decrease in number travelling. Similarly the increase in high value of time users choosing to travel results in an increase total travel time for this class.

To show implementation of the implicit programming method, using the analytic sensitivity expressions (Section 4) to provide gradient information, we consider the NDP of maximising revenue. The results are displayed in Fig. 10. After 865,944 probit choice evaluations, the revenue maximising toll is found to be £18.41. The solid dots mark each iteration of the SQP algorithm (iterations numbered), which terminate at the crossed-circle. To verify the progress of the algorithm, the solid line shows the underlying behaviour of the network, plotted by computing the probit SUE flows at each of 100 tolls between £0 and £50.

In comparison, with the same total demand but with only one user class having value of time = 5.6 ppm, we get the revenue maximising NDP shown in Fig. 11. The optimal toll of £9.61 is reached after 459,000 probit choice evaluations.

7. Conclusions

In this paper we present a formulation of probit SUE with multiple user classes and elastic demand, and establish conditions under which the equilibrium flows are differentiable with respect to the design parameters. Sensitivity expressions are derived for the equilibrium flows with respect to network design parameters that can relate to perturbations in OD demands, values of time, tolls and capacities. The constituent Jacobian matrices provide gradient information for the flows resulting from changes to these parameters. The method of implementation described, particularly the avoidance of simulation methods, avoids the difficulties in interpreting Monte Carlo error and is shown to be applicable to realistic-sized networks.

With these analytic expressions for the gradient of the equilibrium flows, we considered the Network Design Problem in the case where the user and network response is governed by the elastic demand, probit SUE model presented. Such a model has claims to greater behavioural realism than the commonly used
UE model, and in addition the perceptual variance has desirable consequences in terms of the smoothness of the resulting NDP. A gradient-based method was used to solve illustrative NDPs.

Acknowledgements

The research described in this paper was conducted under the financial support of the U.K. Engineering & Physical Sciences Research Council. Thanks to Mike Maher and Andrea Rosa for valuable discussions regarding the calculation of the probit SUE flows, and to the referees for their comments.

References