Article:
Maher, M.J. (2001) Stochastic user equilibrium assignment with elastic demand. Traffic, Engineering and Control, 42 (5). pp. 163-167. ISSN 0041-0683

Published paper

Maher, M.J. (2001) Stochastic user equilibrium assignment with elastic demand. Traffic Engineering & Control, 42(5), pp.163-167.

SUEED - Stochastic User Equilibrium Assignment with Elastic Demand

Mike Maher
School of the Built Environment, Napier University, 10 Colinton Road, Edinburgh EH10 5DT

Abstract
It is well-known that, in deterministic user equilibrium assignment, it is straightforward to allow for elastic demand. The research described in this paper sets out to show that the same is true for stochastic user equilibrium assignment with elastic demand (SUEED). It presents a new objective function for SUEED, and a simple solution algorithm that can be implemented by only a minor modification to existing assignment software. A numerical example illustrates the working of the algorithm

1. Introduction

Conventional traffic assignment techniques assume that the demand matrix is fixed. However, it is now generally accepted that this may not always be accurate. A reduction in travel costs through, for example, the construction of a new highway scheme, may lead to the release of previously suppressed trips. The SACTRA Report (1994) concluded that there was overwhelming evidence to show that the fixed demand assumption was unsafe and that there are a variety of possible behavioural responses to increased congestion. As well as re-routeing, there may be changes in mode choice, trip frequency, destination choice and time of travel. Further evidence was gathered (Coombe *et al,* 1998) that the impact of highway capacity *reduction* (through, for example, the implementation of a bus priority scheme) was likely again to result in changes in mode choice, time of travel and destination choice. In both cases, it was recommended that, in the short term at least, this complex set of responses should be approximated by an *elastic demand* assignment model. Demand is said to be elastic, if the level of demand between an O-D pair is a function simply of the travel cost between that O-D pair.

The purpose of this paper is to show that it is as easy to solve the assignment problem with elastic demand as it is for fixed demand. This has been known for some considerable time for deterministic user equilibrium (DUE) assignment; here we show that the same is true for stochastic user equilibrium (SUE) assignment. The aim is to show that, with very little modification, existing assignment software can be used to solve for Stochastic User Equilibrium with Elastic Demand (SUEED).

We start, in section 2, by reviewing briefly the techniques adopted for the solution of DUE assignment with elastic demand. Then we move on to consider previous work on SUE with fixed demand in section 3, types of stochastic loading methods in section 4, and the determination of step size in section 5. In section 6, we consider the formulation of the SUEED problem, with a new solution algorithm described in section 7, and results from its application in section 8. We then see how the approach can be extended to take account of multiple user classes in section 9 before summarising the main findings in section 10.

2. Elastic demand in (deterministic) user equilibrium

Techniques for allowing for elastic demand in deterministic user equilibrium (DUE) assignment models have been established for many years and have been implemented in commercial software: as in the SATEASY module within the SATURN suite (Van Vliet and Hall, 1993), for example. The original formulation of the DUE model as an optimisation problem (Beckmann *et al*, 1956) also included the formulation of the DUE model with elastic demand:

$$z = \sum_a \int_0^{x_a} c_a(x)\,dx - \sum_{rs} D_{rs}^{-1}(C_{rs}) \tag{1}$$

where x_a and c_a are respectively the flow and cost on link a, and C_{rs} is the travel cost between O-D pair rs. It has been shown that a convenient way to solve the elastic demand problem is to treat it as a fixed demand problem and to create, for each O-D pair, an extra imaginary link directly connecting the origin and destination to carry the "excess" demand (that is, that part of the fixed total demand that does not travel through the real network). The cost-flow function for the imaginary link is constructed from the inverse demand function $D^{-1}{}_{rs}(q_{rs})$..

Therefore, for DUE assignment with elastic demand:

- The problem has been formulated as a function minimisation problem; and
- The problem can be readily solved as a fixed demand problem by the creation of extra links.

Hence, the DUE problem with elastic demand is in effect no more difficult to solve than the fixed demand problem.

3. Previous work on SUE assignment

Before dealing with elastic demand, we give a brief summary of the "state of play" of fixed demand SUE assignment modelling. This will provide the basis for describing the extension to elastic demand.

The SUE assignment problem, in which individual drivers minimise their own *perceived* travel costs was formulated by Sheffi and Powell (1982) as an unconstrained optimisation problem in the flows $\boldsymbol{x}$, with objective function:

$$z_{SUE} = \sum_a x_a\, c_a(x_a) - \sum_a \int_0^{x_a} c_a(x)\,dx - \sum_{rs} q_{rs} S_{rs}(\mathbf{x}) \tag{2}$$

in which q_{rs} is the demand and S_{rs} is expected minimum perceived travel cost (sometimes referred to as the "satisfaction" function) between O-D pair rs. All terms, including the S_{rs}, in the expression above for z are routinely calculated within the stochastic loading

The solution to this problem is obtained by the simple iterative technique shown in schematic form in Figure 1. The current solution $\boldsymbol{x}$, consisting of the vector of link flows, is input into the link cost-flow functions to provide current costs $\boldsymbol{c}$. These are then used as the mean link costs in the input to a stochastic loading routine (of which we shall say more in a moment). The output is the "auxiliary" solution consisting of the vector $\boldsymbol{y}$ of link flows. The updated current solution is then a linear combination (or weighted average) of the old current solution $\boldsymbol{x}$ and the auxiliary solution $\boldsymbol{y}$, with a step length λ (about which we shall also say more in a moment): that is, $\boldsymbol{x}^{(n+1)} = (1-\lambda)\boldsymbol{x}^{(n)} + \lambda \boldsymbol{y}^{(n)}$ where n is the iteration number. The process is repeated until convergence: that is, until $\boldsymbol{x} = \boldsymbol{y}$ or, more precisely, until some measure of the difference between them is sufficiently small.

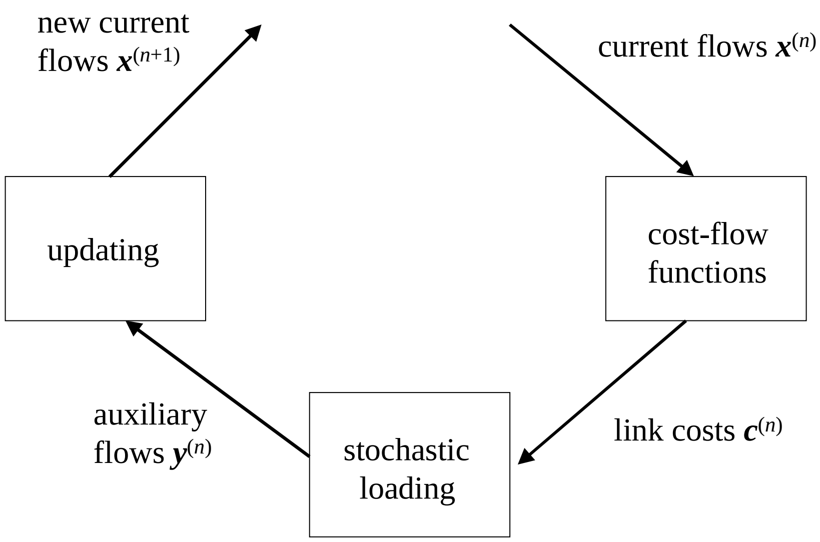

Figure 1: standard scheme for the SUE iterative process

4. Types of stochastic loading

The stochastic loading process may be any of a number of forms, including the Monte Carlo methods of Burrell (1968), with uniformly distributed link costs, or Daganzo and Sheffi (1977) with Normally distributed costs: both are available in SATURN. Alternatively, logit loading may be used in the form either of the STOCH algorithm (Dial, 1971) which, by omitting any “inefficient” links avoids any loops, or that of Bell (1995) that allows for *all* possible loops. A further alternative is the SAM probit loading method first described in Maher (1992) and more fully in Maher and Hughes (1997a) that takes proper account of overlapping paths, unlike logit. Our principle emphasis here will be on the numerically-based methods of logit and probit loading rather than the simulation-based methods.

The overall structure of a stochastic loading (whether probit or logit) is illustrated in Figure 2. In the Forward Pass (FP) through the network the link costs $\boldsymbol{c}(\boldsymbol{x})$ are input and, for each origin in turn, the splits $\boldsymbol{p}$ (the proportions of traffic entering each node via each of the “before” links) and the satisfactions (OD travel costs) $\boldsymbol{S}$ are found. In the Backward Pass (BP) the demands $\boldsymbol{q}$ are used, together with the splits $\boldsymbol{p}$, to calculate the auxiliary link flows $\boldsymbol{y}$. Note particularly (i) that the demands are not used in the Forward Pass and (ii) that the flows $\boldsymbol{x}$ and the demands $\boldsymbol{q}$ need not be consistent (that is, at an intermediate stage of the iterative process, the sum of flows on links emerging from a zone centroid need not be equal to the sum of demands originating at that zone). The significance of these points will be seen later.

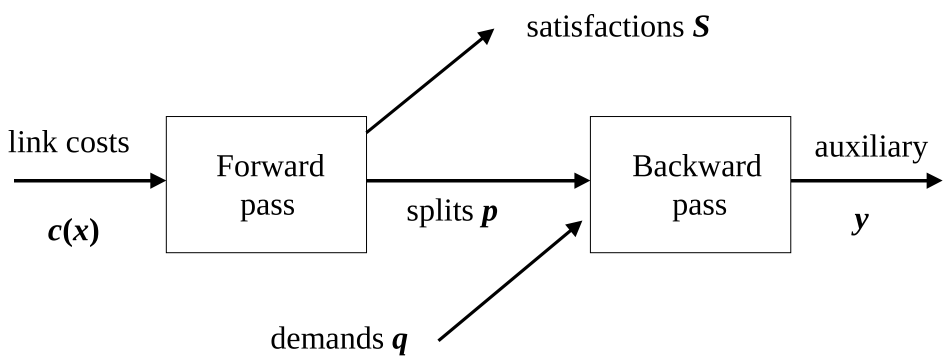

Figure 2: structure of a stochastic loading

5. Step length determination

The step length to be taken from current towards auxiliary may be pre-set (as in the familiar Method of Successive Averages (MSA), with steadily reducing values at each iteration, such as the typical $\lambda = 1/(n+1)$), or may be optimised in a manner similar to that of the Frank-Wolfe (1956) method for DUE. Figure 3 shows a typical plot of the SUE objective function (2) along a search direction: hence, the left hand end, where $\lambda = 0$, corresponds to the current solution $\boldsymbol{x}$, whilst the right hand end, where $\lambda = 1$, corresponds to the auxiliary solution $\boldsymbol{y}$. The optimal step length is that which gives the minimum value of z or equivalently where the gradient g (= $dz/d\lambda$) is zero. A simple scheme to estimate the optimal λ is provided by calculating g at the two end points and using linear interpolation to estimate where $g = 0$, at $\lambda^* = -g_0 / (-g_0 + g_1)$.

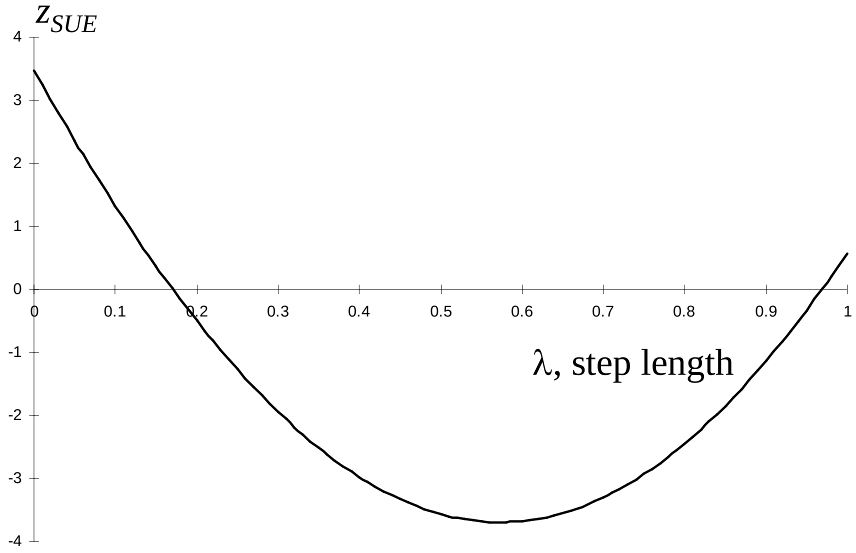

Figure 3: SUE objective function along a search direction

These values of g can be easily calculated, using quantities automatically found in the loading process.

$$g_0 = -\sum_a (y_a - x_a)^2 \frac{dc_a}{dx_a} \qquad g_1 = -\sum_a (y_a - x_a)(w_a - y_a)\frac{dc_a}{dx_a} \tag{3}$$

where $\boldsymbol{w}$ is the "auxiliary of the auxiliary": that is, in addition to the stochastic loading based on the current solution $\boldsymbol{x}$, we carry out a *second* stochastic loading based on the auxiliary solution $\boldsymbol{y}$. The disadvantage, then, of this approach over MSA is that *two* stochastic loadings have to be carried out for each iteration. The advantage, though, is that the algorithm converges *very much* faster than MSA, even allowing for the fact that, because of the extra loading, each iteration takes twice as long as one in MSA. A sensible hybrid approach is to use the pre-set MSA step lengths for the first few iterations and then switch to the use of optimised step lengths for subsequent iterations.

6. An objective function for SUEED

Having reviewed the position for SUE with fixed demand, we turn now to SUE assignment with *elastic* demand, which we shall refer to as SUEED. At the SUEED solution, we have simultaneous equilibria in the flows ($\boldsymbol{x} = \boldsymbol{y}$) and the demands ($q_{rs} = D_{rs}(S_{rs})$ for each OD pair, rs). Just as (2) formulates SUE assignment as an optimisation problem, so it can be shown that the objective function in (4) below formulates SUEED as an optimisation problem (again *unconstrained*) in $\boldsymbol{x}$ and $\boldsymbol{q}$. More technical details of equivalence and convexity can be found in Maher *et al* (1999) and Maher and Zhang (2000):

$$
\begin{aligned}
z_{SUEED} &= \sum_a x_a\, c_a(x_a) \;-\sum_a \int_0^{x_a} c_a(x)\,dx \\
&+ \sum_{rs} D_{rs}^{-1}(q_{rs}) D_{rs}(S_{rs}(\mathbf{x})) - \sum_{rs} \;(S_{rs}(\mathbf{x})) D_{rs}(S_{rs}(\mathbf{x})) \qquad (4) \\
&+ \sum_{rs} \int_0^{q_{rs}} D_{rs}^{-1}\;(q)\,dq \;-\; \sum_{rs} q_{rs}\, D_{rs}^{-1}\;(q_{rs})
\end{aligned}
$$

The benefit of having an objective function is that it enables a sound theoretical basis for the SUEED model. For example, it can be shown that when the derivatives of (4) with respect to link flows and demands are all zero, the twin sets of conditions for SUEED are satisfied. It is also provides direct evidence of convergence as its value steadily decreases through an iterative process. Furthermore, closer inspection and manipulation of the objective function leads to the formulation of a solution algorithm, referred to as the Balanced Demand Algorithm (BDA) which turns out to be a very natural and efficient method of solution. Other approaches are possible (see, for example, Maher and Hughes, 1997b, 1998) but the BDA is the simplest.

7. The Balanced Demand Algorithm

The form of the BDA is very simple and draws on the fact, mentioned earlier, that the forward pass of a stochastic loading is carried out without any reference to the demands but uses only the link flows $\boldsymbol{x}$ from which the splits $\boldsymbol{p}$ and the satisfactions $\boldsymbol{S}$ are found. In the BDA, the demands are only specified at the end of the forward pass (FP) and before the start of the backward pass (BP), by applying the demand function (DF) for each O-D pair, to give a set of "balanced" demands $q_{rs} = D_{rs}(S_{rs})$ that therefore satisfy the elastic demand equilibrium conditions. The whole scheme of the algorithm is shown in Figure 4, with the first loading using the current flows $\boldsymbol{x}$ as input and producing as output the auxiliary pattern $\boldsymbol{y}$, and the second loading using $\boldsymbol{y}$ as input and producing the "auxiliary of the auxiliary" $\boldsymbol{w}$.

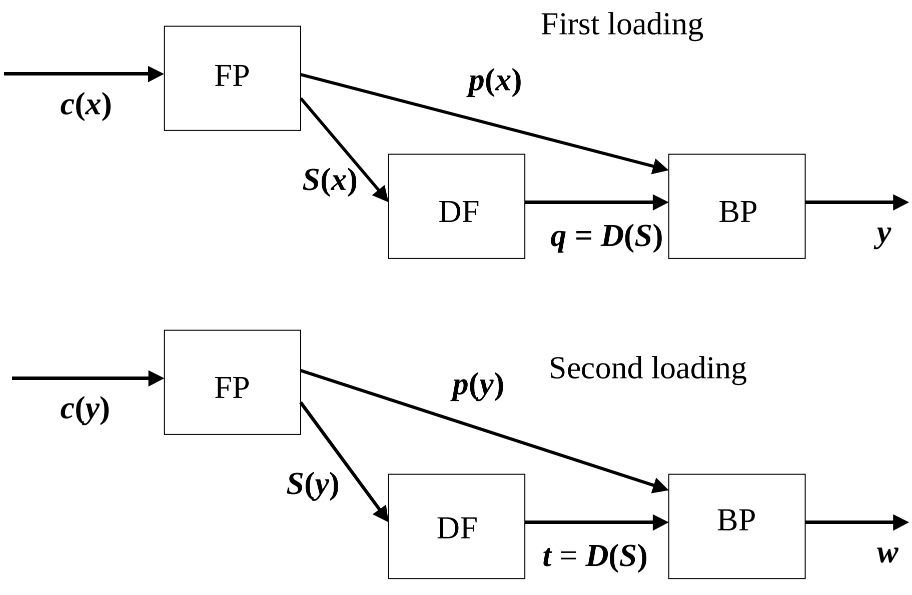

Figure 4: Schematic description of the Balanced Demand Algorithm

The BDA for the solution of the SUEED problem therefore requires no more computational effort than that for SUE with fixed demand. Again, two stochastic loadings are carried out in each iteration: one at the current solution $\boldsymbol{x}$ and the other at the auxiliary solution $\boldsymbol{y}$. The gradients g are calculated in exactly the same manner at each point and, by linear interpolation of the gradient, the step length at which $g = 0$ can be estimated.

8. Test results

We illustrate the application of the Balanced Demand Algorithm firstly to a small test network where the calculations can be fully set out and followed. The network, consisting of five links, three paths and a single OD pair is shown in Figure 5. The link cost-flow functions are each of the simple BPR form: $c(x) = c_0(1 + x/X)$ where c_0 is the free-flow cost and X is the notional capacity. The free-flow costs are $\boldsymbol{c}_0 = (10, 13, 5, 15, 10)$ and the capacities are $\boldsymbol{X} = (600, 800, 400, 900, 700)$. We shall apply logit loading with a value of the sensitivity parameter $\theta = 0.05$ and assume a demand function of the "constant elasticity", power law form $D = D_0\ (S/S_0)^{-e}$ with $e = 0.7$, base values of $D_0 = 1000$ and $S_0 = 20$. Starting with an arbitrary initial flow pattern of $\boldsymbol{x} = (400, 800, 100, 300, 900)$, the detailed calculations for the first iteration are set out in Table 1.

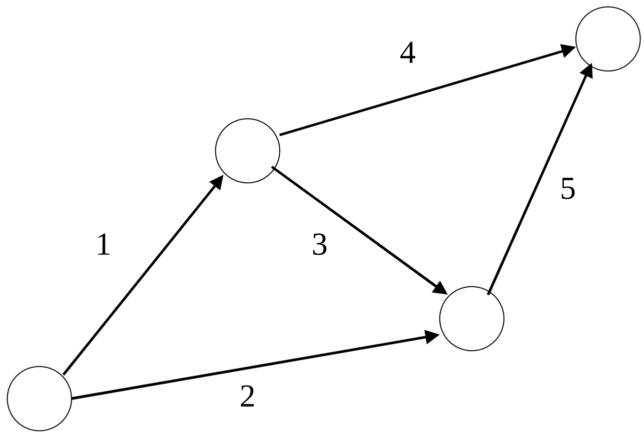

Figure 5: simple five-link network

In the upper part of Table 1, the first loading is carried out, starting from the flow pattern $\boldsymbol{x}$ above. This consists of calculating, in turn, the link costs $\boldsymbol{c}$, the three path costs $\boldsymbol{C}$, the values of $\exp(-\theta C)$ and, from the multinomial logit model, the proportions $\boldsymbol{p}$ using each path and the satisfaction value S:

$$p_j = \frac{\exp(-\theta C_j)}{\sum_k \exp(\theta C_k)} \qquad S = -\frac{1}{\theta}\log_e\left(\sum_k \exp(-\theta C_k\right)$$

[It should be noted that although here, for simplicity, the loading is carried out on the basis of *paths*, it is normally envisaged that a *link-based* loading method would be used, especially when the network is of a more realistic size].

Then, the value of S = 21.10073 is input to the demand function to calculate the "balanced" demand q = 963.2121 which is then used together with the splits $\boldsymbol{p}$ to produce the auxiliary $\boldsymbol{y}$. Finally, the value of the gradient g_0 is found to be –9622.129. The whole set of calculations is then repeated in the lower half of the table, for the second loading, starting with the auxiliary pattern $\boldsymbol{y}$ this time, and producing at the right hand side the "auxiliary of the auxiliary" $\boldsymbol{w}$ and the gradient g_1 = 4773.066. The estimate of the optimal step length θ = 9622.129/(9622.129 + 4733.006) = 0.668. To complete the iteration, the new current solution is found by the linear combination 0.332$\boldsymbol{x}$ + 0.668$\boldsymbol{y}$.

This iterative process is continued until the difference between $\boldsymbol{x}$ and $\boldsymbol{y}$ is acceptably small. In this example, after five iterations, the current and auxiliary link flows are identical to two decimal places and are (604.55, 393.92, 253.19, 351.36, 647.11) and the values of the two gradients gradient g_0 and g_1 are both reduced to values of the order of 10^{-6}.

The BDA has also been applied to many other test networks using both logit and probit loading. A set of typical results is shown in Figure 6 (these being for the Winnipeg network with over 4,000 links and 2,500 nodes). The plot compares the rates of convergence of BDA and MSA. (The measure of convergence J is the *logarithm* of the difference between the current and auxiliary solutions: at full convergence this equals -∞ and hence the rate of convergence is indicated by the steepness of the graph. It can be seen that the convergence of MSA is initially good but becomes steadily slower, whereas the convergence of BDA is uniformly fast throughout.) Note that the plot is against the number of *loadings* and not the number of iterations, and so takes account of the fact that the BDA requires *two* loadings per iteration instead of the one in MSA. Generally, a good policy is to use MSA for the early iterations and then to switch to the BDA for subsequent iterations.

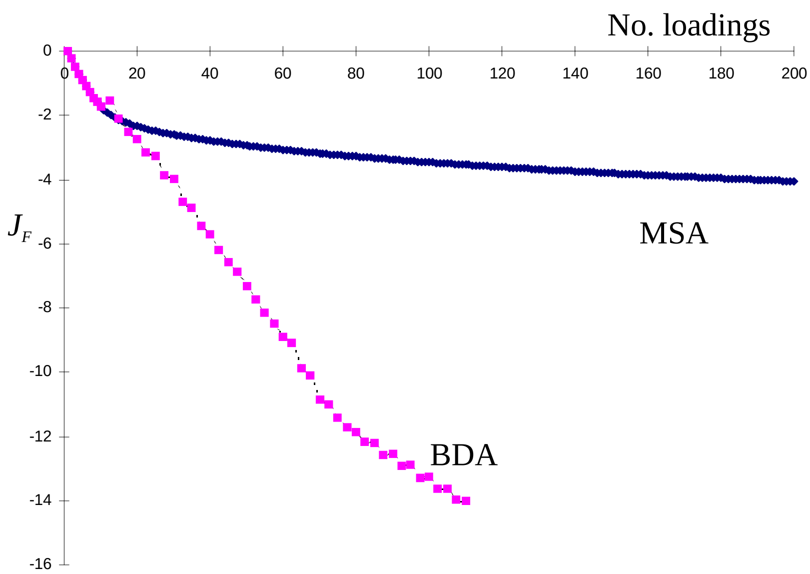

Figure 6: Plot of convergence measure against number of loadings, for the BDA and MSA.

9. Extension to allow for multiple user classes

For certain applications it is desirable to distinguish between different user classes. For example, to distinguish between those whose vehicles are equipped with an in-car driver information system and those whose vehicles are unequipped, one might attribute the former with a lower value of the variability parameter in the stochastic loading process (to reflect the better and more reliable information they had on network conditions). See Maher and Hughes (1996). Alternatively, one might wish to distinguish between different vehicle types in respect of their speeds and hence in their travel costs. In fact, it can be shown that both are equivalent, if the ratio of costs between any pair of user classes is the same for all links in the network.

One might therefore wish to formulate what could be referred to as a MUCSUEED problem (Multiple User Class Stochastic User Equilibrium with Elastic Demand). At the MUCSUEED solution, the demand is in equilibrium for each separate user class (that is, $q_{rsu} = D_{rsu}(S_{rsu})$ for each O-D pair rs and for each user class u), as well the flows being a SUE solution for that set of demands. It is, in fact, quite easy to show that the MUCSUEED problem can be formulated as an unconstrained function minimisation problem, with an objective function that is a natural extension of that for SUEED:

$$
\begin{aligned}
z_{MUCSUEED} &= \sum_a x_a\, c_a(x_a) \; - \sum_a \int_0^{x_a} c_a(x)\, dx \\
&+ \sum_u \sum_{rs} D_{rsu}^{-1}(q_{rsu}) D_{rsu}(S_{rsu}(\mathbf{x})) - \sum_u \sum_{rs} (S_{rsu}(\mathbf{x})) D_{rsu}(S_{rsu}(\mathbf{x})) \\
&+ \sum_u \sum_{rs} \int_0^{q_{rsu}} D_{rsu}^{-1}(q)\, dq \; - \sum_u \sum_{rs} q_{rsu}\, D_{rsu}^{-1}(q_{rsu})
\end{aligned}
\tag{5}
$$

It also follows from this that the BDA can be readily extended to deal with multiple user classes. In fact, for each origin, a forward pass is carried out *for each user class* based on the current flows $\boldsymbol{x}$ so that user-specific splits $\boldsymbol{p}_u$ and satisfactions $\boldsymbol{S}_u$ are determined. Then, the balanced demands $q_{rsu} = D_{rsu}(S_{rsu})$ are calculated and applied, together with the splits, in the backward pass for each class, to determine the auxiliary flows $\boldsymbol{y}_u$ which, when summed over all user classes, give the total link flows $\boldsymbol{y}$. The calculation for optimal step length is identical to that in SUEED and, in particular, uses only the *total* link flows $\boldsymbol{x}$ and $\boldsymbol{y}$.

Hence, the extension to multiple user classes is straightforward both in terms of the formulation of the problem and its solution.

10. Summary and conclusions

The main achievement of the research has been to demonstrate that it is no more difficult to solve the SUEED problem than the SUE problem with fixed demand. In particular

- A new objective function for SUEED has been formulated whose properties (equivalence, convexity etc) have been formally established;
- An efficient algorithm (the Balanced Demand Algorithm) has been developed which converges much faster than the previous conventional method of MSA;
- The algorithm can be applied with any form of logit or probit stochastic loading process.
- The calculations to perform the BDA require only quantities that are readily available in any existing assignment software. Therefore, only minimal changes need to be made to such software to enable it to find the SUEED solution using the BDA.

- The objective function and the solution algorithm can both be readily extended to allow for multiple user classes (MUCSUEED).

Acknowledgements
The research reported here was funded by the Engineering and Physical Sciences Research Council (research grant GR/L91399) and their support is gratefully acknowledged.

References
Beckmann M., McGuire C.B. and Winsten C.B. (1956) *Studies in the Economics of Transportation*. Cowles Commission Monograph, Yale University Press, New Haven.

Bell M.G.H. (1995) Alternatives to Dial's logit assignment algorithm. *Transportation Research B*, **29B**(4), 287-295.

Burrell J.E. (1968) Multipath route assignment and its application to capacity restraint. Proc. 4th International Symposium on the Theory of Road Traffic Flow, Karlsruhe.

Coombe D., Bates J. and Dale M. (1998) Modelling the traffic impacts of highway capacity reductions. *Traffic Engineering and Control*, **39**(7/8), 430-433.

Daganzo C. and Sheffi Y. (1977) On stochastic models of traffic assignment. *Transportation Science* **11**(3), 253-274.

Dial R.B. (1971) A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research* **5**, 83-111.

Frank M. and Wolfe P. (1956) An algorithm for quadratic programming. *Naval Research Logistics Quarterly* **3**(1-2), 95-110.

Maher M.J. (1992) SAM - A stochastic assignment model. In: *Mathematics in Transport Planning and Control* (ed. J. Griffiths), Oxford University Press, 121-133.

Maher M.J. and Hughes P.C. (1996) Estimation of the potential benefit from an ATT system using a multiple user class stochastic user equilibrium assignment model. In: *Applications of Advanced Technologies in Transportation Engineering* (eds: Stephanedes Y.J. and Filippi F.), American Society of Civil Engineers, 700-704.

Maher M.J. and Hughes P.C. (1997a) A probit-based stochastic user equilibrium assignment model. *Transportation Research* B, **31**(4), 341-355.

Maher M.J. and Hughes P.C. (1997b) An Algorithm for SUEED - Stochastic User Equilibrium with Elastic Demand. 8th IFAC Symposium on Transportation Systems, Chania, Crete.

Maher M.J. and Hughes P.C. (1998) The stochastic user equilibrium assignment problem with elastic demand - a comparison of new approaches. 6th Meeting of the EURO Working Group on Transportation, 9-11 September 1998, Gothenberg, Sweden

Maher M.J., Hughes P.C. and Kim K-S. (1999) New algorithms for the solution of the stochastic user equilibrium assignment problem with elastic demand. 14th International Symposium on Transportation and Traffic Theory, Jerusalem, July 1999.

Maher M.J. and Zhang X. (2000) Formulation and algorithms for the problem of stochastic user equilibrium assignment with elastic demand. 8th EURO Working Group meeting on Transportation, Rome, September 2000.

Sheffi Y. and Powell W.B. (1982) An algorithm for the equilibrium assignment problem with random link times. *Networks* **12**(2), 191-207.

Standing Advisory Committee on Trunk Road Assessment (SACTRA) (1994) *Trunk Roads and the Generation of Traffic*. HMSO, London.

Van Vliet D. and Hall M. (1993) *SATURN Version 8.4 User Manual*. Institute of Transport Studies, University of Leeds.