An Algorithm for the Stochastic User Equilibrium Problem

Olof Damberg
Jan T. Lundgren*
Michael Patriksson

Department of Mathematics
Linköping Institute of Technology
S-581 83 Linköping
Sweden

Report LiTH-MAT-R-1992-48

January 9, 1995

Abstract

In this paper we present a new algorithm for the approximate solution of the logit-based stochastic user equilibrium problem. The main advantage of this algorithm is that it provides route flows explicitly, of particular interest in the evaluation of route guidance and information systems; in previously proposed methods for the stochastic user equilibrium problem, only link flows are provided. The proposed algorithm alternates between two main phases. In the subproblem phase, profitable routes are generated. In a restricted master problem, a descent method is used to solve the restriction to the original problem to the subset of the total set of routes generated so far. We present and evaluate alternative strategies for generating routes algorithmically, and discuss the possibility of utilizing such strategies for reducing the inherent problem of overlapping route flows in logit-based traffic models.

1 Introduction

The *traffic assignment problem* is fundamental in transportation analysis, and concerns the distribution of travel demands to routes in a traffic network. In mathematical models, the problem is commonly represented by a directed graph where each link is associated with a travel cost function, and where the demands, in terms of number of trips, are given by an origin-destination (O-D) matrix.

The classical assumption made in the models regarding the distribution is the *User Equilibrium* (UE) principle. This principle, which is due to Wardrop (1952), is formulated as follows.

> The journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle on any unused route.

*Author to whom all correspondence should be addressed

Characteristic features of the situation described by the UE principle are that all travelers have perfect information about travel costs, and are uniform, in the sense that they have the same travel cost perception. The user equilibrium state may thus be characterized as a situation in which no traveler can reduce his own journey cost by unilaterally changing routes. Models based on this behavioral assumption may give a reasonable approximation of the actual traffic situation, and efficient optimization algorithms can be applied to generate equilibrium flows. However, if there is a lack of information among the travelers about which routes are the shortest, or if travelers have different preferences and perceive travel costs differently, it is then natural to assume that traffic flows do not satisfy the user equilibrium conditions.

A consequence of the difference in travel cost perception is that routes with a higher cost than the least-cost routes will also be utilized. (It is still natural to assume that a more costly route is less probable to be chosen by a traveler than a less costly one.) In order to allow for variations in the travelers' perception of travel cost, one could extend the basic UE model to include a random component in the travel cost function. The probability of choosing a specific route, given actual costs, is then dependent on the probability distribution of this random component.

Daganzo and Sheffi (1977) extend the user equilibrium principle to the principle of *Stochastic User Equilibrium* (SUE), which is stated as follows.

> In a stochastic user equilibrium network no user believes he can improve his travel time by unilaterally changing routes.

An equivalent notion of SUE is that no traveler can reduce his own perceived travel cost by unilaterally changing routes.

Assignment models which do not assume that flows satisfy the user equilibrium conditions have attracted an increased interest during the last few years, especially in the context of route (or vehicle) guidance and information (RGI) systems. In a simulation tool for testing and evaluating the potential benefits of an RGI system, there is a need for models that can describe situations where drivers have less than perfect information about the network conditions. The situation that can be observed in the absence of an RGI system is one such example. The initial conditions should then be compared with the traffic conditions that can be achieved if an RGI system is applied. The SUE condition is the most commonly utilized initial condition in models for simulations of RGI systems; see Bonsall (1992) and Van Vliet (1990). In SUE models, the behavior of different classes of drivers, e.g., driver with or without access to information about the traffic conditions given by an RGI system, can be modeled through appropriate choices of parameters related to the variance in travel cost perceptions.

In algorithms for SUE models presented to date, only link flows are provided. In the evaluation of an RGI system optimal route flows are of interest to obtain, since route flows contain more detailed information about the travel pattern. The purpose of this paper is to present a new algorithm for the approximate solution of the logit-based stochastic user equilibrium problem of Fisk (1980), which provides route flows explicitly. The iterative algorithm combines a scheme for generating routes with a descent method for a restriction to the original problem to a subset of the routes.

The outline of the rest of this paper is as follows. In the following section we introduce the necessary notation, formulate the stochastic user equilibrium model of Fisk, and discuss its most important properties. In Section 3 we present the new algorithm. In Section 4 we evaluate the algorithm and analyze the solutions obtained, and in Section 5 we make some concluding remarks.

2 The stochastic user equilibrium problem

2.1 The basic user equilibrium model

Consider a transportation network $\mathcal{G}$=($\mathcal{N}$,$\mathcal{A}$), where $\mathcal{N}$ denotes the set of nodes, and where each (directed) link $a \in \mathcal{A}$ is associated with a generalized travel cost, $t_a(f_a)$, representing the disutility of using link a as a function of its flow f_a. This cost may include several additive components, perhaps most importantly the travel time on the link, and it is assumed that, as a result of congestion, t_a is a positive and strictly increasing function of the flow on link a.

For certain pairs of origins and destinations, $(p, q) \in \mathcal{C}$, where $\mathcal{C} \subset \mathcal{N} \times \mathcal{N}$, there is a given positive demand d_{pq} of flow. For each O-D pair (p, q), we denote the set of simple (cycle-free) routes from p to q by $\mathcal{R}_{pq}$ — a set which in general is not known explicitly — and the flow on route r from p to q by h_{pqr}. By defining a link-route incidence matrix (δ_{pqra}) for $\mathcal{G}$, i.e., $\delta_{pqra} = 1$ if route $r \in \mathcal{R}_{pq}$ contains link a, and 0 otherwise, link flows may be calculated from route flows according to the relation

$$f_a = \sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}} \delta_{pqra} h_{pqr}, \qquad \forall a \in \mathcal{A}. \tag{1}$$

The user equilibrium traffic assignment problem may be formulated as the following convex program; see e.g., Beckmann et al. (1956, Chapter 3) and Dafermos (1972). Here, f denotes the vector of link flows f_a.

$$\text{[TAP-UE]} \qquad \begin{aligned} \min \quad & T(f) = \sum_{a\in\mathcal{A}} \int_0^{f_a} t_a(s)ds \\ \text{s.t.} \quad & \begin{array}{rcll} \displaystyle\sum_{r\in\mathcal{R}_{pq}} h_{pqr} & = & d_{pq} & \forall (p,q) \in \mathcal{C} \\ h_{pqr} & \geq & 0 \quad \forall r \in \mathcal{R}_{pq} & \forall (p,q) \in \mathcal{C} \\ \displaystyle\sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}} \delta_{pqra} h_{pqr} & = & f_a \quad \forall a \in \mathcal{A} & \end{array} \end{aligned}$$

This is a well studied model, and a number of algorithms exists for its solution. One of the most commonly applied algorithms is the Frank–Wolfe algorithm; see e.g., LeBlanc et al. (1975). Extensions to the Frank–Wolfe method include PARTAN directions (LeBlanc et al., 1985) and the simplicial decomposition methods of Hearn et al. (1987) and Larsson and Patriksson (1992). For a survey of algorithms for the [TAP-UE] problem and their relationships, see Patriksson (1994).

2.2 Stochastic user equilibrium

In stochastic user equilibrium models the deterministic traffic model is extended by including random components in the travel cost functions to account for the variations in the travelers' perception of travel cost. The travelers' choice of route can then be modeled using discrete choice theory, which describe individuals' choice between competing alternatives. The perceived travel cost on route r in O-D pair (p,q) can be expressed as $C_{pqr} = c_{pqr} + \varepsilon_{pqr}$, where c_{pqr} is the actual travel cost at a given flow, and where ε_{pqr} is the random component. Depending on the probability distributions chosen for the random components, different models are obtained, see e.g., Sheffi (1985, Part IV), Sheffi and Powell (1981, 1982) and Daganzo (1982).

If we assume that the random terms ε_{pqr} are independently and identically distributed Gumbel variates, we obtain the widely used multinomial *logit* model. Given actual travel costs, the route choice probabilities can then be given in closed form as (e.g., Sheffi, 1985, Section 11.1)

$$P_{pqr} = \frac{\mathrm{e}^{-\Theta c_{pqr}}}{\sum_{l \in \mathcal{R}_{pq}} \mathrm{e}^{-\Theta c_{pql}}}, \qquad \forall r \in \mathcal{R}_{pq},\ \forall (p,q) \in \mathcal{C}, \tag{2}$$

where Θ is a positive scaling factor related to the variances in the perceived travel costs.

Another model is the multinomial *probit* model, which results from assuming that the random terms are normally distributed. In the probit model, either an analytical approximation or a Monte Carlo simulation must be used to estimate the route choice probabilities, see e.g., Sheffi (1985, Section 11.2).

Among the properties of the logit formula (2) is that of every route receiving a positive flow in the equilibrium state, regardless of its travel cost. Since the number of routes is very large in general, a route-flow based algorithm would be prohibitive unless the number of routes could be limited. By combining it with a scheme in which routes are generated iteratively, one may, however, limit the number of routes within the method. This approach is used in the algorithm proposed in this paper. Since only a subset of the routes is generated by the algorithm, the solution obtained is an approximate stochastic user equilibrium. For the subset of the routes generated, however, the algorithm gives the optimal route flows.

2.3 A stochastic user equilibrium model

We consider the logit-based stochastic user equilibrium model of Fisk (1980).

$$\text{[TAP-SUE]} \qquad \begin{array}{lrcll} \min & Z(h) & = & \frac{1}{\Theta} \sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}} h_{pqr} \log h_{pqr} + \sum_{a\in\mathcal{A}} \int_0^{f_a} t_a(s)ds & \\ \text{s.t.} & \sum_{r\in\mathcal{R}_{pq}} h_{pqr} & = & d_{pq} & \forall (p,q) \in \mathcal{C} \\ & h_{pqr} & \geq & 0 \quad \forall r \in \mathcal{R}_{pq} & \forall (p,q) \in \mathcal{C} \\ & \sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}} \delta_{pqra} h_{pqr} & = & f_a \quad \forall a \in \mathcal{A} & \end{array}$$

Here h denotes the vector of route flows h_{pqr}. The parameter Θ is assumed non-negative, and $x \log x$ is defined as zero at $x = 0$. For $\Theta < \infty$, Z is strictly convex; hence, equilibrium route flows are unique. This is an advantage compared to the case when $\Theta = \infty$, (i.e., for [TAP-UE]) where equilibrium route flows are not unique in general.

To show that the minimum of Z satisfies the logit choice formula (2), Fisk considers the Lagrangean function

$$L(h,\pi) = Z(h) + \sum_{(p,q)\in\mathcal{C}} \pi_{pq} \left(\sum_{r\in\mathcal{R}_{pq}} h_{pqr} - d_{pq} \right),$$

where π is the vector of multipliers π_{pq} corresponding to the demand feasibility constraints. The partial derivative $\frac{\partial L}{\partial h_{pqr}}$ only exists where $h_{pqr} > 0$. Setting $\frac{\partial L}{\partial h_{pqr}} = 0$, we obtain

$$\frac{1}{\Theta}(\log h_{pqr} + 1) + \sum_{a\in\mathcal{A}} \delta_{pqra} t_a(f_a) + \pi_{pq} = 0.$$

This expression yields

$$h_{pqr}(\pi_{pq}) = \mathrm{e}^{-\Theta c_{pqr} - 1 - \Theta\pi_{pq}},$$

where, from (1),

$$c_{pqr} = c_{pqr}(h) = \sum_{a\in\mathcal{A}} \delta_{pqra} t_a(f_a) = \sum_{a\in\mathcal{A}} \delta_{pqra} t_a \left(\sum_{(p,q)\in\mathcal{C}} \sum_{r\in\mathcal{R}_{pq}} \delta_{pqra} h_{pqr} \right) \tag{3}$$

is the travel cost on route r in O-D pair (p,q) at the flow h.

Inserting $h_{pqr}(\pi_{pq})$ in the relaxed constraint yields

$$h_{pqr} = d_{pq} \frac{\mathrm{e}^{-\Theta c_{pqr}}}{\sum_{l\in\mathcal{R}_{pq}} \mathrm{e}^{-\Theta c_{pql}}}, \tag{4}$$

which is equivalent to the conditions (2) of route choice probability.

The parameter Θ may be interpreted as a measure of the travelers' sensitivity to the route costs. It can also be interpreted as the degree of information that is available to a traveler; by varying this parameter it is possible to represent different traveler behavior. In the limit when $\Theta \to \infty$ the user equilibrium solution is obtained. This corresponds to a situation where the travelers are extremely sensitive to an increased route cost (or, equivalently, that they have perfect information about the actual travel costs). As pointed out by Fisk (1980), the value of the parameter Θ is uniquely determined by the optimal flows, and may therefore be calibrated in a practical application, for instance from observed flows.

For all finite values of Θ, the solution to [TAP-SUE] will yield a nonzero flow for each route in the network. Since the number of routes is very large in general, and therefore cannot be stored explicitly, algorithms proposed for this problem have mainly been based on generating the optimal *link* flows instead of the optimal *route* flows.

In the Method of Successive Averages (MSA) [e.g., Fisk, 1980, Powell and Sheffi, 1982, Sheffi and Powell, 1982 and Sheffi, 1985, Section 12.2], simulations of the link costs are performed iteratively, and the resulting shortest route flows are weighted together with the current flow solution using predetermined weighting factors. An advantage of this method is that it may be used also for other probability distributions of travel costs. Chen and Alfa (1991) suggest improvements to the MSA algorithm, where the predetermined weighting is replaced with a line search with respect to Z, or to its deterministic part T (as in the Frank–Wolfe algorithm). Bell et al. (1993) observe that the algorithm of Chen and Alfa (1991) may yield inconsistent flows, and propose an iterative balancing procedure, embedded in a route generation scheme, to enforce consistency.

If the congestion effects are ignored (i.e., link costs are fixed), the resulting logit-based *stochastic network loading problem* can be efficiently solved in terms of link flows. Dial (1971) introduces the notion of an *efficient* route to (implicitly) define a subnetwork ($\mathcal{G}' = (\mathcal{N}, \mathcal{A}')$ with $\mathcal{A}' \subset \mathcal{A}$), in which optimal link flows can be found through the determination of two shortest route patterns. In the congested case, this strategy could be used iteratively to find an approximate stochastic user equilibrium solution, by, for instance, replacing the simulation part of MSA, see Fisk (1980) and Sheffi (1985, Page 327).

3 An algorithm for the stochastic user equilibrium problem

In this section we present a new algorithm for approximately solving [TAP-SUE], which provides optimal route flows, given a generated subset of the complete set of routes. The algorithm is an extension to the disaggregate simplicial decomposition algorithm of Larsson and Patriksson (1992), originally proposed for [TAP-UE]. For the readers' convenience, we provide a brief description of this method.

3.1 Disaggregate simplicial decomposition

Simplicial decomposition methods for [TAP-UE] utilize that any point in the feasible set (which is a bounded polyhedron) can be described as a convex combination of its extreme points (e.g., Bazaraa et al., 1993, Theorem 2.6.7). The extreme points are defined by shortest route patterns, and are generated iteratively through linearized subproblems (the same as in the Frank–Wolfe algorithm). Alternately, the original objective function is minimized over the convex hull of all the shortest route patterns generated so far. This *restricted master problem* generalizes the line search step of the Frank–Wolfe method, in which the current flow is combined only with the latest generated shortest route pattern (e.g., Hearn et al., 1987 and Patriksson, 1994, Page 120).

In the problem [TAP-UE], disregarding the link flow definitional constraints, the feasible set is actually a Cartesian product of feasible sets for the different O-D flows. Larsson and Patriksson (1992) observe that simplicial decomposition methods may then be used in a *disaggregate* fashion, i.e., extreme points may be generated independently, as single route flows, for the different O-D pairs. As a consequence, the restricted master problem will now have one convexity constraint for each O-D pair, instead of one in total as in ordinary simplicial decomposition methods. Since extreme points can be identified as single route flows, the restricted master problem is actually equivalent to the original problem [TAP-UE], with the exception that the route set $\mathcal{R}_{pq}$ is replaced by the subsets $\hat{\mathcal{R}}_{pq}$ for all $(p,q) \in \mathcal{C}$ hitherto generated.

The *disaggregate simplicial decomposition* (DSD) algorithm works as follows. Suppose that a restricted master problem, defined by the sets $\hat{\mathcal{R}}_{pq}$ of routes, has been solved. Given its solution, the corresponding link flows f_a and link costs $t_a(f_a)$ are calculated. The shortest routes are computed for all O-D pairs, and each set $\hat{\mathcal{R}}_{pq}$ is augmented by the shortest route if it is not already contained in the set $\hat{\mathcal{R}}_{pq}$. The procedure is then repeated, with the solution of a new restricted master problem. Convergence of this scheme is finite, and the output is a set of routes with flows satisfying the user equilibrium condition. See Larsson and Patriksson (1992) for further details.

We shall now extend the DSD algorithm to (approximately) solve the logit-based stochastic user equilibrium problem [TAP-SUE].

3.2 The restricted master problem phase

Since the feasible sets of [TAP-UE] and [TAP-SUE] are equal, when using a route generation strategy for [TAP-SUE] corresponding to that outlined above for [TAP-UE], a restricted master problem of [TAP-SUE] is defined by replacing the sets $\mathcal{R}_{pq}$ with the subsets $\hat{\mathcal{R}}_{pq}$ currently known in the formulation of [TAP-SUE]. Strategies for algorithmically generating routes in the solution of [TAP-SUE] are given in Section 3.3.

The numbers of variables (i.e., the number of routes) in the restricted master problems vary (and are in general much lower than in the original problem). When referring to a flow h, we shall implicitly always assume that the vector h is consistent with the sets $\hat{\mathcal{R}}_{pq}$, i.e., that only the routes in $\hat{\mathcal{R}}_{pq}$ are represented by variable components in h. Consequently, when the set $\hat{\mathcal{R}}_{pq}$ is augmented by a route, the variable vector h is augmented with an element $r \in \mathcal{R}_{pq}$ (and the corresponding variable h_{pqr} is initially set to zero).

Consider a restricted master problem of [TAP-SUE], where the subsets $\hat{\mathcal{R}}_{pq}$ of $\mathcal{R}_{pq}$, $(p,q) \in \mathcal{C}$, are known. This problem is solved using the following descent approach.

Let $\overline{h}^k$ be a given feasible route flow at some iteration k, and let $c^k_{pqr} = c_{pqr}(\overline{h}^k)$, for all $r \in \hat{\mathcal{R}}_{pq}$ and $(p,q) \in \mathcal{C}$, denote the corresponding route costs, given by (3). Define an auxiliary route flow $\underline{h}^k$ by

$$\underline{h}^k_{pqr} = d_{pq} \, \frac{e^{-\Theta c^k_{pqr}}}{\sum_{l \in \hat{\mathcal{R}}_{pq}} e^{-\Theta c^k_{pql}}}, \qquad \forall r \in \hat{\mathcal{R}}_{pq},\ \forall (p,q) \in \mathcal{C}. \tag{5}$$

If the vector $\underline{h}^k - \overline{h}^k$ is nonzero, then it defines a descent direction with respect to Z. A line search is made in the direction of $\underline{h}^k - \overline{h}^k$, resulting in a new solution $\overline{h}^{k+1}$, and the process is repeated. If $\underline{h}^k = \overline{h}^k$, then the flow $\overline{h}^k$ solves the restricted master problem, i.e., the stochastic user equilibrium conditions (4) are satisfied for the restriction of the original problem to the subsets $\hat{\mathcal{R}}_{pq}$ of the total set of routes.

The motivation behind this simple scheme is the following. The auxiliary flow $\underline{h}^k$, defined in (5), is the result of the solution of an approximation of the restricted master problem, where the link costs are fixed at the flow $\overline{h}^k$, i.e., the deterministic part T of Z is linearized at $\overline{h}^k$. In the terminology of Patriksson (1993), the flow $\underline{h}^k$ is therefore the result of solving a *partially linearized* problem, and the vector $\underline{h}^k - \overline{h}^k$, if nonzero, hence defines a descent direction with respect to Z by the result of Patriksson (1993, Theorem 2.1). Since Z is strictly convex, the sequence $\{\overline{h}^k\}$ of route flows converges to the unique solution of the restricted master problem from any feasible initial flow $\overline{h}^0$; see Patriksson (1993, Theorem 2.2).

Moreover, the optimal value of each partially linearized problem, which can be computed given (5), is a lower bound of that of the restriction of [TAP-SUE]; a criterion based on the relative objective error is therefore available for the termination of the descent method for each restricted master problem.

The restricted master problem is combined with a column generation scheme, i.e., a method for augmenting the sets $\hat{\mathcal{R}}_{pq}$. Strategies for generating routes are discussed in the next section.

3.3 The column generation phase

The descent approach described in the previous section solves the restricted master problem given subsets $\hat{\mathcal{R}}_{pq}$ of the complete sets of routes. How well this solution describes the real travel pattern depends on how well the subsets $\hat{\mathcal{R}}_{pq}$ approximate the sets of routes that are actually used. The choice of strategy by which the routes (or columns) are generated is obviously of crucial importance.

In theory, the total number of routes in $\mathcal{R}_{pq}$ can be very large. However, it is reasonable to assume that in practice a small portion of them will carry most of the demand. The column generation strategy should, ideally, be such that only the routes that are most utilized will be generated, and that the error made by assuming that the remaining routes have zero flow is small.

There exist many ways of generating routes; in fact, in the proposed algorithm the definition and generation of the elements of the sets $\hat{\mathcal{R}}_{pq}$ are arbitrary and may, for instance, be based on experience from the properties of the actual travel pattern. In the following, we describe two strategies for iteratively generating routes; they are both based on the solution of shortest route problems given the solution to a restricted master problem of [TAP-SUE].

The first approach is to generate routes through the solution of shortest route problems based on the *actual* (i.e., deterministic) travel costs, thereby temporarily ignoring the random components of the travel costs. This is the familiar approach used in column generation algorithms for [TAP-UE].

This is a valid approach especially when the random components are less significant; this applies to the situation where the value of the parameter Θ is large and travelers are sensitive to the travel costs (e.g., when the congestion level is high). Moreover, from the formula (2), the probability of utilizing a particular route is a decreasing function of its actual cost. The use of the described route generation scheme could therefore be expected to yield the routes that have the highest probability of being utilized, i.e., the routes in a SUE solution with the largest flows.

For networks with low congestion levels or small values of Θ, the perception error may, however, be more significant, and the strategy of generating routes based on actual costs may result in very few routes being generated. For these cases, this strategy of generating routes is therefore not recommended.

The second and perhaps most natural approach is to generate routes based on *randomized* link costs, as in the MSA algorithm, in this way incorporating the differences in the travelers' perceptions of travel costs. The resulting method then is a natural extension of MSA where, instead of weighting link flows together in a predetermined fashion, all the available route information is utilized in the restricted master problem.

Given a feasible link flow and a corresponding (fixed) actual travel cost, an auxiliary link flow solution is in the MSA algorithm generated as an average of all-or-nothing solutions given by sampled realizations of the perceived travel cost. A predetermined step length is then taken towards the auxiliary flow, defining the next iterate. (This approach is valid for many choices of distributions of the random cost components; see Sheffi, 1985, Chapter 12.)

Hence, the auxiliary flow is the result of possibly many shortest route solutions. Since we are interested in utilizing all the route information available (through the formulation and solution of a restricted master problem), it seems unnatural to generate routes (in the simulation subalgorithm) that are not stored.

Mimis (1984) (see also Sheffi, 1985, Section 12.3) concludes from numerical experiments on large-scale probit models that the best convergence results for MSA are obtained by using a single drawing of perceived travel costs, coupled with a single shortest route solution, at every main iteration. Using such a strategy in the solution of [TAP-SUE] would correspond to generating the routes based on one sampling of the travel costs at the solution of the previous restricted master problem; in this way, all the route information gained may be utilized.

In both approaches, initial sets of routes may be defined by a rough approximation of the solution to [TAP-UE], for instance by a few steps of an *incremental assignment* algorithm (see Sheffi, 1985, Pages 115–116), or by an early truncation of the DSD algorithm.

To comply with the independence assumption on the random cost terms of the logit model, each set $\hat{\mathcal{R}}_{pq}$ of routes should consist only of routes that have no links in common, i.e., the routes should not overlap each other. (The less they overlap, the better does the solution comply with the independence assumption.) Algorithms for logit-based traffic models tend to generate solutions with overestimates of flows on overlapping routes; see Burrell (1976), Florian and Fox (1976), Daganzo and Sheffi (1977), Ben-Akiva et al. (1984) and Sheffi (1985, Section 11.1). In the algorithm proposed in this paper it is, however, possible to avoid generating flows on overlapping routes by deleting (or suitably modifying) any route generated that overlaps with a previously generated one more than a maximal allowed measure of overlapping; depending on the overlap measure, this may be easily performed by augmenting the route generation phase with a suitable check.

The application of the logit model gives unreasonable results if the routes overlap heavily. On the other hand, if overlapping routes are avoided completely, the resulting flows will not necessarily be in accordance with route flows which can be observed in practice. The amount of overlapping that is acceptable is mainly an empirical question. Since, in the proposed algorithm, route flows are computed explicitly, it is for the first time possible to analyze the presence of overlapping in stochastic traffic models; an initial investigation is made in Section 4.4.

The algorithm terminates when no new routes are obtained from the column generation phase. In neither of the two above column generation approaches the algorithm is guaranteed to converge to the unique optimal solution of [TAP-SUE], since it, in general, terminates before all routes are generated. It is, however, guaranteed to solve the restriction of [TAP-SUE] to any set of routes generated.

3.4 A summary of the proposed algorithm

0. **Initialization**
   Compute an initial set of routes $\hat{\mathcal{R}}_{pq}$ for all $(p,q) \in \mathcal{C}$, and an initial route flow h^0.
   Let $i = 0$.

1. **Restricted master problem phase**
   Let $k = 0$.
   Let $\overline{h}^k = h^i$.
   Repeat the following until convergence.[1]

[1] Termination criteria may be based on the relative objective error, the step length t_k, the difference between $\underline{h}^k$ and $\overline{h}^k$, or on the relative decrease of Z or change in the flows between two consecutive line searches.

Compute route costs $c_{pqr}^k = c_{pqr}(\overline{h}^k)$ for all $r \in \hat{\mathcal{R}}_{pq}$ and $(p, q) \in \mathcal{C}$.
Compute the auxiliary route flow $\underline{h}^k$ according to the logit formula (5).
Terminate if $\underline{h}^k = \overline{h}^k$.
Solve the line search problem

$$t_k = \arg\min_{t \in [0,1]} Z(t) \stackrel{\text{def}}{=} Z\left(\overline{h}^k + t\left(\underline{h}^k - \overline{h}^k\right)\right). \tag{6}$$

Let the new point be

$$\overline{h}^{k+1} = \overline{h}^k + t_k\left(\underline{h}^k - \overline{h}^k\right).$$

Let $k := k + 1$.
Let $\overline{h} = \overline{h}^k$ be the output.

2. **Column generation phase**
Generate a set of routes based on one of the strategies described in Section 3.3, and augment the sets $\hat{\mathcal{R}}_{pq}$.
Terminate if no new routes are found.[2]
Let $h^{i+1} = \overline{h}$.
Let $i := i + 1$.
Go to Step 1.

4 Numerical experiments

In this section we present an initial evaluation of the proposed method through numerical experiments on a version of the well-known network model of Winnipeg. The network has 1052 nodes, 2836 links and 4344 O-D pairs (see Florian and Nguyen, 1976, for a description).

4.1 Implementational issues

We implemented the algorithm in FORTRAN-77 and performed the tests on a Sun SPARCstation IPX. The implementation was based on the DSD code developed by Larsson and Patriksson (1992) for the solution of [TAP-UE].

We implemented both column generation strategies described in Section 3.3. In the strategy where actual (i.e., deterministic) travel costs are used, the initial sets $\hat{\mathcal{R}}_{pq}$ of routes are generated by performing two incremental assignments initiated at zero flow.

The strategy of generating routes based on randomized travel costs is based on a single drawing of perceived link travel costs, using a Gumbel distribution of the random cost components. Since the values of the error terms in this cost are independent of the flows and their actual costs, the link travel costs used in the shortest route calculations may be very far from the actual ones, possibly leading either to the generation of negative cycles in the shortest route algorithm, or to the generation of routes with unreasonably high actual costs. In initial experiments, both these situations did indeed occur; the latter one resulted in numerical difficulties in the solution of the restricted master problems, in particular for large values of Θ. In order to circumvent

[2] Alternative global termination criteria may be defined by the relative decrease of the objective or change in the flows between two consecutive restricted master problems.

the generation of routes of unrealistically high costs, we therefore introduce some amount of dependency in the link cost error terms of the actual costs, by forcing the absolute value of each link error term to be at most 25 % of that of the actual link cost; we further reject any route generated from entering a set $\hat{\mathcal{R}}_{pq}$ that has an actual cost exceeding the average cost of the routes in $\hat{\mathcal{R}}_{pq}$ by more than 50 %. The initial route sets $\hat{\mathcal{R}}_{pq}$ are obtained by performing two incremental assignments based on the link costs obtained from a rough solution to [TAP-UE]. By truncating the random cost components and rejecting very expensive routes we hope to generate approximate solutions to [TAP-SUE] where routes with insignificant flows are avoided.

We also implemented a rule by which routes with large amounts of overlapping can be rejected from entering the sets $\hat{\mathcal{R}}_{pq}$. The rule is as follows. Assume that in O-D pair (p,q) a new route has been generated. Calculate the portion of the total number of links in the new route that is already included in any of the routes in $\hat{\mathcal{R}}_{pq}$. If this portion is higher than a (predetermined) maximal portion of overlap, $MO \in [0,1]$, then the route is rejected. By varying the parameter MO between zero and one, the amount of overlap can be kept below any level desired. (With MO=0, no overlapping will be present in the solution generated, while, with MO=1, no route is rejected.) Note that there are many other ways of defining overlap measures; such measures may, for example, be defined by taking the costs of the overlapping links into account.

Each new restricted master problem is solved using the solution to the previous restricted master problem as initial solution. The line searches (6) are performed using an Armijo-type (Armijo, 1966) step length rule. (The initial line search is, however, performed using simultaneous search, since directional derivatives of Z are unbounded at a zero route flow.) Each restricted master problem is terminated when the relative improvement in the objective value between two consecutive line searches is below 10^{-5}, and the overall algorithm is terminated when either no routes are added to a set $\hat{\mathcal{R}}_{pq}$ in Step 2 or the number of main iterations exceeds 100.

When using actual travel costs in the route generation process, the execution time of the algorithm is roughly comparable to that of the DSD algorithm for [TAP-UE], and was found to decrease with a decreasing value of Θ. An explanation for this is that for a small value of Θ the restricted master problems are easily solved, since the auxiliary flow $\underline{h}$ obtained from the logit formula (5) is near-optimal, regardless of the route costs. (In the limit when $\Theta \to 0$, the auxiliary flow $\underline{h}_{pqr} = d_{pq}/\,|\mathcal{R}_{pq}|$, $r \in \mathcal{R}_{pq}$, $(p,q) \in \mathcal{C}$, is indeed optimal.) When using randomized travel costs, however, the total number of restricted master problems (and as a result their final sizes) grows rapidly with a decreasing value of Θ; the resulting CPU time is for this reason increasing with decreasing values of Θ. Further details on the computational efficiency of the algorithm are found in Damberg et al. (1992).

4.2 Number of routes generated

In the first set of experiments we investigated the number of routes generated by the algorithm for different route generation strategies and for values of the parameter Θ in the interval $[0.05, 5]$. (For any finite value of Θ the numbers of routes generated and used are equal.) For comparison, the number of routes used in the corresponding UE solution (taken from an execution of the DSD algorithm) is also computed.

In Figure 1, the average number of routes used in an O-D pair, as a function of Θ, is given for the case where the route generation is based on actual costs. (The total number of routes in each O-D pair is weighted by the corresponding demand.) The increase in the number of routes used when the value of Θ decreases is expected, since a smaller value of Θ corresponds to a situation where the travelers have less knowledge of (or are less sensitive to) the actual route costs, and therefore a larger number of routes are perceived to be the shortest by at least some travelers.

[Place of Figure 1]

In the UE solution about 43% of the O-D pairs have positive flows on *one* route only, mainly because the origin and destination are situated close to each other so that only one realistic route exists. This explains the relatively small average number of routes used in the UE solution (and, indeed, in the SUE solution as well).

Figure 2 shows the average number of routes used in an O-D pair for the case where route costs are simulated. Results from tests with the values $MO = 0, 0.3, 0.75$, and 1.0 of the overlap parameter are given.

[Place of Figure 2]

The introduction of the link cost error terms results in more routes being generated. For all values of the parameter MO, the shape of the function relating the value of Θ to the average number of routes generated is the same.

When the value of Θ decreases, the average number of routes generated (and used) increases, until a plateau is reached. The reason for reaching a plateau is twofold. First, the absolute values of the link cost error terms are truncated when becoming too great compared to the actual costs. (The lower the value of Θ, the more often the truncation is performed.) Second, the number of main iterations is limited to 100, which prevents, for small values of Θ, some of the O-D pairs from receiving flows on unrealistically large numbers of routes.

The number of routes generated of course depends on the value of the parameter MO. For $MO = 1.0$, on average 20 routes are generated. For $MO = 0$, on the other hand, very few routes are generated. The large difference between the average number of routes used when $MO = 1.0$ and $MO = 0.75$ is accounted for by the vast number of very similar routes that is generated for $MO = 1.0$ for spatially separated pairs of origin and destination. (Between spatially separated origins and destinations in the large network model of Winnipeg, there are many alternative routes which are similar with respect to both their travel costs and the links included.)

With the aid of the parameter MO one is thus able to simultaneously restrict the number of routes that is used in a solution as well as the routes' topological similarities.

Note that the lower the value of MO, the more sensitive the output is to the order in which the routes are generated, and in particular to the choice of the initial sets $\hat{\mathcal{R}}_{pq}$. Also, the network topology has a higher influence on the routes generated by the algorithm for smaller values of MO.

4.3 Route cost distribution

In the solution to a SUE model the costs on the routes used in an O-D pair will be different. In the second set of experiments we investigated the distribution of the route costs for the two route generation strategies and different values of Θ, by computing an aggregated measure of the minimum, the average, and the maximum route cost among the O-D pairs. (The average route cost was for each O-D pair obtained by weighting the cost of each route with the corresponding route flow.) Each aggregated measure was obtained by weighting the corresponding measure in each O-D pair by the corresponding demand.

In Figure 3, the increase (in percent) in the minimum, average and maximum SUE route costs compared to the average UE route cost are given, as functions of the value of the parameter Θ, for the case where the route generation is based on actual costs. (The average UE route cost was obtained by weighting the equilibrium route costs among the O-D pairs by their respective demands.) The average SUE route cost (as well as the minimum and maximum route cost) is increasing with a decreasing value of Θ, since less sensitivity to (or less information about) actual travel costs causes a larger number of travelers to choose more expensive routes. The gap between the maximum and minimum route costs is also increasing with a decreasing value of Θ.

[Place of Figure 3]

The minimum travel cost in the SUE solution is roughly the same as the average UE route cost. This is due to the fact that the routes with the lowest actual costs in the SUE and UE solutions tend to be similar, since there will always be some travelers choosing a least-cost route regardless of the value of Θ. A larger value of Θ corresponds to a situation where travelers are more sensitive to the actual costs, and hence a vast majority choose least-cost routes. However, a few travelers will choose more expensive routes. For large values of Θ, the minimum travel cost in the SUE solution is lower than the UE travel cost, since the least-cost routes are almost identical and slightly fewer are using them in the SUE solution.

In Figure 4, the increase (in percent) in the minimum, average and maximum SUE route costs compared to the average UE route cost are given for the case where the route generation is based on simulated link costs. Figures for the values $MO = 0, 0.3, 0.75$, and 1.0 on the overlap parameter are given, as functions of the value of the parameter Θ.

[Place of Figure 4]

From Figure 4, we may conclude that the variations in the travel costs are slightly more pronounced when using randomized link travel costs in the route generation process. The main reason for the overall relatively small variations is the use of the error term truncation strategy.

A comment is needed on the results for the case where no overlapping is allowed ($MO = 0$). As noted earlier, the smaller the value of MO, the more the output depends on the order in which the routes are generated, since routes generated late in the process will be rejected with high probability. Since the routes generated initially often are not among the least-cost ones, and few more routes are subsequently allowed to enter into the sets $\hat{\mathcal{R}}_{pq}$ (as seen from Figure 2), the resulting route costs will be comparatively high. The small number of routes generated also explains the very limited travel cost variation. This result shows that the initial sets of routes should be chosen with some care, at least for very small values of MO.

The average SUE route cost (or the corresponding system travel cost) can be used to calibrate the value of the parameter Θ. In a numerical study of the networks of Weetwood, U.K., and Barcelona, Spain, reported by Van Vliet (1990), the value of Θ was chosen such that the average travel cost was of the order of 6% greater than in the user equilibrium solution. (The reason for this choice was that Jeffery (1987) found — from analyzing times and distances corresponding to a sample of journeys made in the U.K. — that this difference was about 6%.) According to Figure 3, a value of Θ around 0.08 would correspond to this cost increase, while from Figure 4 ($MO = 1.0$) the corresponding value is about 0.3. (For the networks of Weetwood and Barcelona, Van Vliet (1990) chooses Θ equal to 0.3 and 0.4, respectively.)

4.4 Route overlapping

In the final set of experiments we investigated the amount of overlap that is present in the routes generated for the two route generation strategies implemented and the values $MO = 0, 0.25, 0.5, 0.75$, and 1. For comparison, we here introduced the rule for rejecting overlapping routes in both route generation strategies, and also computed the overlap measures for the corresponding UE solution.

Two aggregated measures of the amount of overlapping were computed. The first overlap measure gives an estimate of the number of routes in an O-D pair that traverses a given link. This measure was computed as follows. For each O-D pair, we computed the number of links that is traversed by a given number of routes. The resulting portions of the total number of links were then weighted with the corresponding demands to define aggregated measures. The aggregated data is the average portions of the links which are traversed by 1, 2 or more routes in an O-D pair. The resulting portions are given in Table 1, for the value $\Theta = 0.08$.

[Place of Table 1]

The second overlap measure is complementary to the first one, and gives an estimate of the portion of the average total route travel cost that originates from links traversed by a given number of routes. (This is a natural measure, since the independence assumption in the logit model is expressed in terms of travel costs.) Aggregated measures were computed similarly to that of Table 1, with the portion of the total number of links replaced by the portion of the total travel cost, thereby obtaining additional information of the properties of the overlapping routes. The results are shown in Table 2, for the value $\Theta = 0.08$.

[Place of Table 2]

A general observation is that the amount of overlap present in the routes generated is fairly limited, both in terms of the portion of the total number of links that are included in overlapping routes and in terms of the portion of the total route costs that originates from those links. (The amount of overlap is, in fact, lower than in the UE solution.) This observation can be made regardless of the amount of overlap allowed; moreover, the amount of overlap present is always much lower than the upper limit specified by the parameter MO.

In Appendix A, we illustrate graphically the overlapping present in two O-D pairs in the solutions obtained for the values 0.75 and 0.25 of MO.

5 Discussion

In this paper we propose a new algorithm for the logit-based stochastic user equilibrium problem. The algorithm provides optimal route flows given a set of routes generated. There is a large amount of freedom in the choice of strategies for generating routes. Different strategies are described and evaluated, and we introduce a scheme by which flows on overlapping routes may be avoided to any degree specified.

The algorithm is able to solve large-scale logit-based SUE models provided that the necessary route storage requirements are met, and provides an approximate solution within the same order of magnitude of execution time as state-of-the-art route generation algorithms for the corresponding UE models.

The route flow solutions obtained were found to have reasonable properties in terms of the number of routes used and their travel cost variances. We also found the the solutions obtained do not include many overlapping route alternatives; they may, however, be avoided completely by setting the value of the parameter $MO = 0$.

We finally remark that while, in the formulation of the mathematical program [TAP-SUE], the route cost c_{pqr} is necessarily additive [cf. (3)] and each travel cost function t_a is separable (and hence symmetric), the logit formula (4) is valid also for asymmetric and non-additive travel costs. This means that using (4) instead of [TAP-SUE] as the definition of the logit route-choice problem, it is possible to incorporate both asymmetric link interactions and generalized (non-separable) costs including distance traveled, monetary costs and pollutant emissions, relevant to traffic management policies. The algorithm proposed above is easily extended to the solution of such a general model, by replacing the objective Z in the line search (6) with another merit function measuring the violation of (4).

A Graphical illustration of the route overlapping

We illustrate the overlapping present in the solutions obtained by the algorithm through graphically presenting the routes generated in two O-D pairs (with $\Theta = 2.0$ and $\Theta = 0.2$ respectively), for two values of the parameter MO (0.75 and 0.25) and for both route generations strategies.

The links in the below graph are aggregates of sequences of links in the original network of Winnipeg. The width of a link is proportional to the number of routes traversing it. Each link is labeled with the number of links to which it corresponds in the original network, and with its actual travel cost. Each graph is supplemented with a list of the routes generated in the corresponding O-D pair, their actual travel costs and flows (in portion of the demand).

[Place of Figure 5]

When using actual link costs in the route generation strategy, only the first route from Figure 5 is generated.

[Place of Figure 6]

When using actual link costs in the route generation strategy, only the first route from Figure 6 is generated.

[Place of Figure 7]

When using actual link costs in the route generation strategy, only the routes 1, 2 and 4 from Figure 7 are generated.

[Place of Figure 8]

When using actual link costs in the route generation strategy, the same routes as in Figure 8 are generated.

Acknowledgments

We thank two anonymous referees for helpful remarks. Louise Lychou provided the numerical results presented in Section 4.4. The research leading to this paper was supported by the Swedish Transport and Communications Research Board (KFB) (Dnr 16/88-62 and Dnr 92-128-63).

References

Armijo L. (1966) Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics,* **16**, 1–3.

Bazaraa M. S., Sherali H. D. and Shetty C. M. (1993) *Nonlinear Programming. Theory and Algorithms* (Second edition). John Wiley & Sons, New York, NY.

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

Bell M. G. H., Lam W. H. K., Ploss G. and Inaudi D. (1993) Stochastic user equilibrium assignment and iterative balancing. In Daganzo C. F. (Ed.), *Transportation and Traffic Theory, Proceedings of the 12th International Symposium on the Theory of Traffic Flow and Transportation, Berkeley, CA, 21–23 July, 1993,*, pp. 427–439 Elsevier, Amsterdam.

Ben-Akiva M., Bergman M. J., Daly A. J. and Ramaswamy R. (1984) Modelling inter urban route choice behaviour. In Volmuller J. and Hamerslag R. (Eds.), *Proceedings of the Ninth International Symposium on Transportation and Traffic Theory, Delft, The Netherlands, 11–13 July 1984,*, pp. 299–330 VNU Science Press, Utrecht, The Netherlands.

Bonsall P. (1992) The influence of route guidance advice on route choice in urban networks. *Transportation,* **19**, 1–23.

Burrell J. E. (1976) Multiple route assignment: a comparison of two methods. In Florian M. A. (Ed.), *Traffic Equilibrium Methods, Proceedings of the International Symposium Held at the Université de Montréal, November 21–23, 1974,*, Vol. 118 of *Lecture Notes in Economics and Mathematical Systems,*, pp. 229–239 Springer-Verlag, Berlin.

Chen M. and Alfa A. S. (1991) Algorithms for solving Fisk's stochastic traffic assignment model. *Transportation Research,* **25B**, 405–412.

Dafermos S. C. (1972) The traffic assignment problem for multiclass-user transportation networks. *Transportation Science,* **6**, 73–87.

Daganzo C. F. (1982) Unconstrained extremal formulation of some transportation equilibrium problems. *Transportation Science,* **16**, 332–360.

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

Damberg O., Lundgren J. T. and Patriksson M. (1992) An algorithm for the stochastic user equilibrium problem. In Baños J. C. M., Friedrich B., Papageorgiou M. and Keller H. (Eds.), *Proceedings of the First Meeting of the EURO Working Group on Urban Traffic and Transportation, Landshut, Germany, 1–3 October, 1992,* Technical University of Munich.

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

Fisk C. (1980) Some developments in equilibrium traffic assignment. *Transportation Research,* **14B**, 243–255.

Florian M. and Fox B. (1976) On the probabilistic origin of Dial's multipath traffic assignment model. *Transportation Research,* **10**, 339–341.

Florian M. and Nguyen S. (1976) An application and validation of equilibrium trip assignment methods. *Transportation Science,* **10**, 374–390.

Hearn D. W., Lawphongpanich S. and Ventura J. A. (1987) Restricted simplicial decomposition: computation and extensions. *Mathematical Programming Study,* **31**, 99–118.

Jeffery D. (1987) Route guidance and in-vehicle information systems. In *Information Technology Applications in Transport,*, VNU Science Press, Utrecht, pp. 319–351.

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

LeBlanc L. J., Helgason R. V. and Boyce D. E. (1985) Improved efficiency of the Frank–Wolfe algorithm for convex network programs. *Transportation Science,* **19**, 445–462.

LeBlanc L. J., Morlok E. K. and Pierskalla W. P. (1975) An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research,* **9**, 309–318.

Mimis S. (1984) Equilibrium traffic assignment models for urban networks. Master's thesis, Center for Transportation Studies, Massachusetts Institute of Technology, Cambridge, MA.

Patriksson M. (1993) Partial linearization methods in nonlinear programming. *Journal of Optimization Theory and Applications,* **78**, 227–246.

Patriksson M. (1994) *The Traffic Assignment Problem — Models and Methods,*. VSP, Utrecht, The Netherlands.

Powell W. B. and Sheffi Y. (1982) The convergence of equilibrium algorithms with predetermined step sizes. *Transportation Science,* **16**, 45–55.

Sheffi Y. (1985) *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods,*. Prentice-Hall, Englewood Cliffs, NJ.

Sheffi Y. and Powell W. B. (1981) A comparison of stochastic and deterministic traffic assignment over congested networks. *Transportation Research,* **15B**, 53–64.

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

Van Vliet D. (1990) Modelling route guidance systems using SATURN. In *DRIVE Project V1054 (ASTERIX) Deliverable 4,*, pp. B–83–B–105.

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

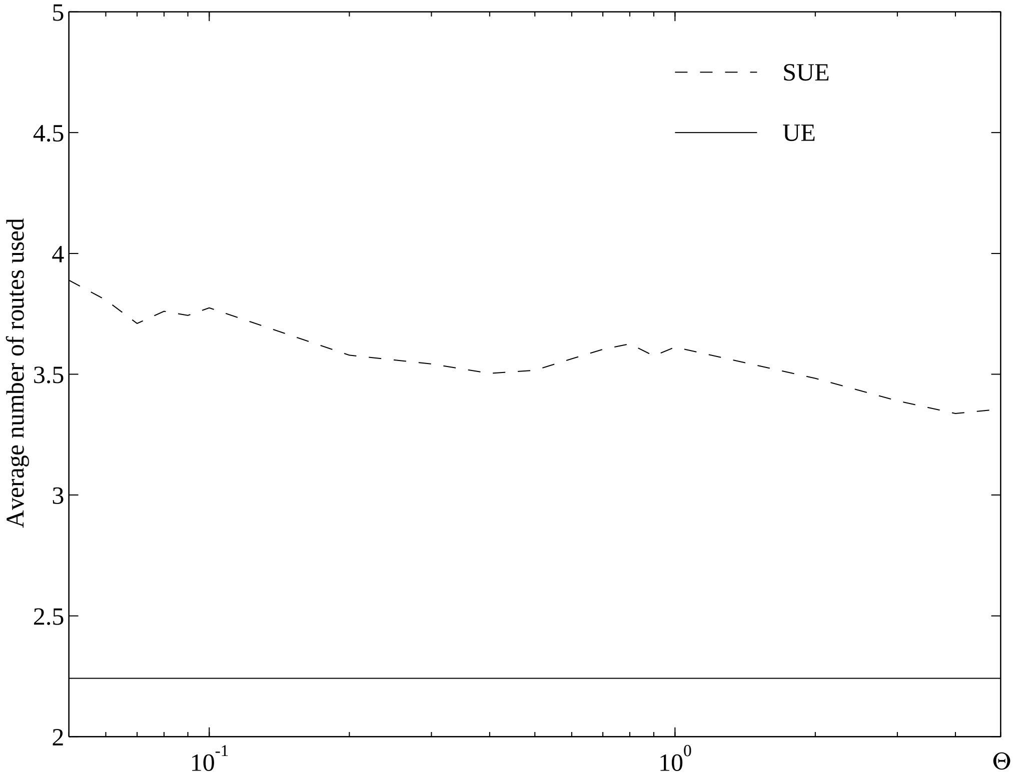


Figure 1: Average number of routes used in an O-D pair. Route generation based on actual costs.

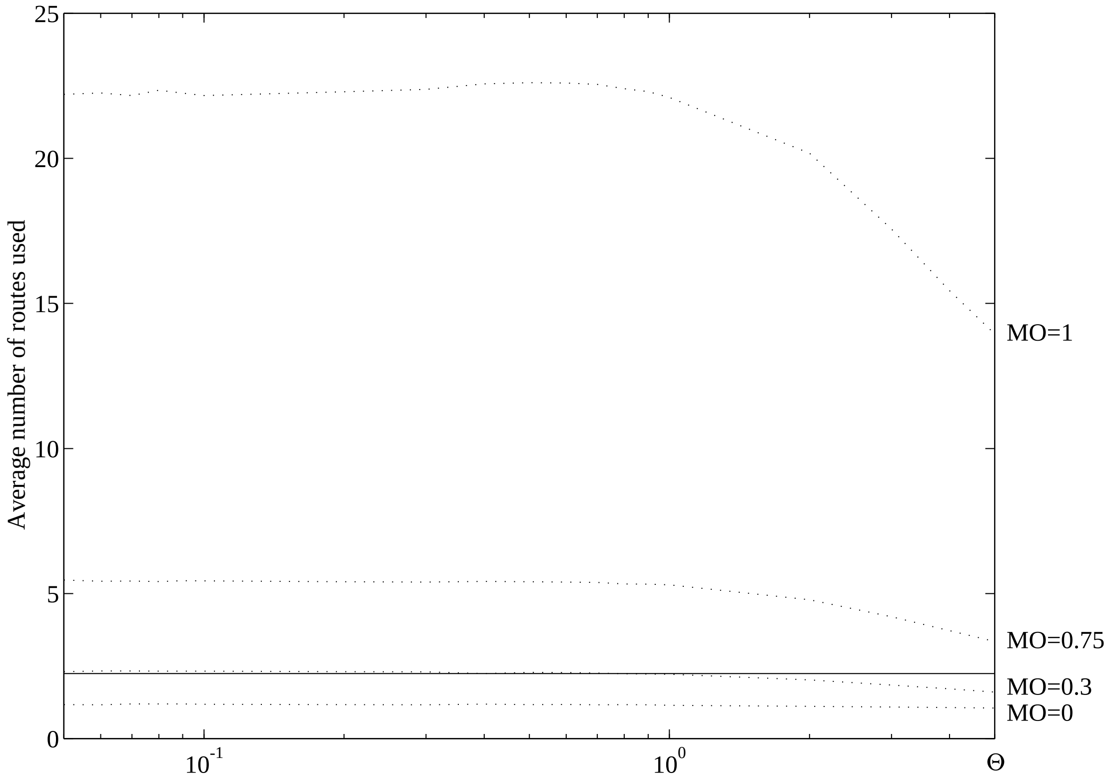


Figure 2: Average number of routes used in an O-D pair. Route generation based on simulated costs.

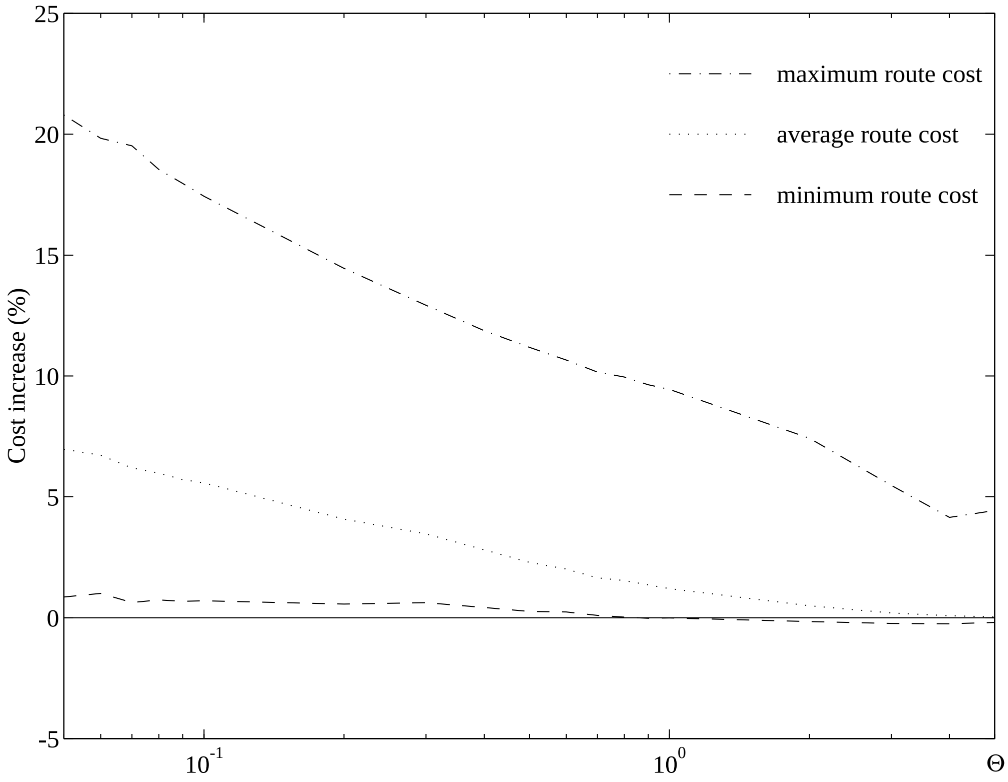


Figure 3: Increase in SUE route costs compared to UE. Route generation based on actual costs.

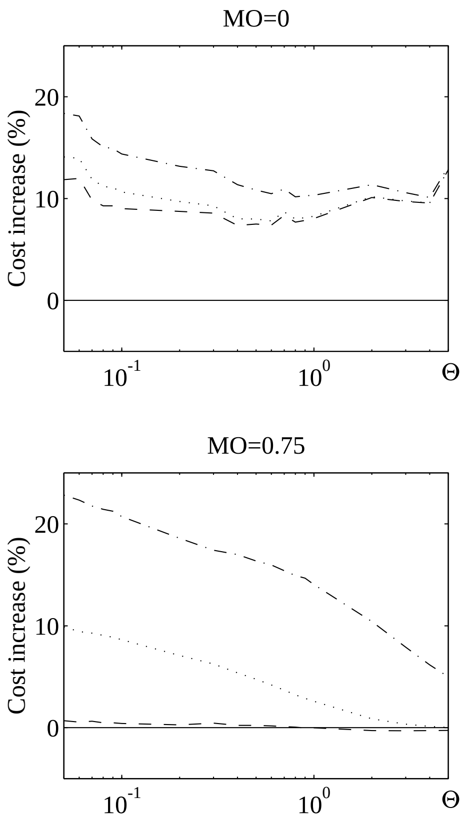


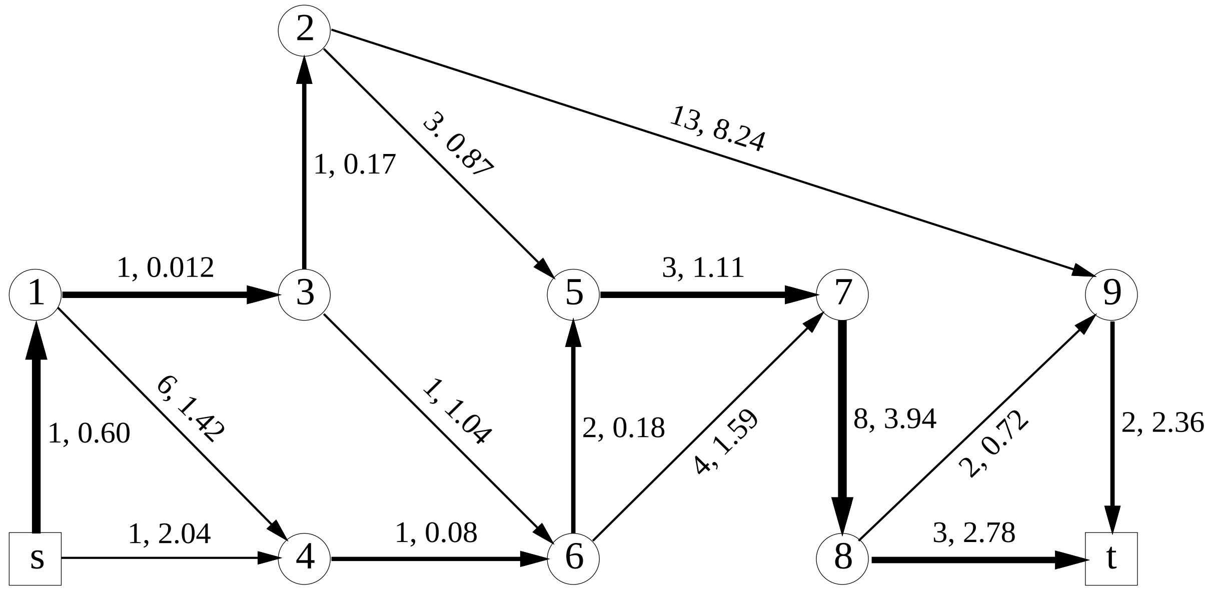
Figure 4: Increase in SUE route costs compared to UE. Route generation based on simulated costs.

	Value of MO	1	≥ 2	2	3	4	5	6
Actual cost	0	100						
	0.25	97	3	3				
	0.50	89	11	10	1			
	0.75	78	22	17	4	1		
	1.00	64	36	20	9	4	2	1
Simulated cost	0	100						
	0.25	91	9	8	1			
	0.50	81	19	15	3	1		
	0.75	72	27	18	6	2	1	
	1.00	68	32	19	8	3	1	1
UE	—	62	38	21	9	5	2	1

Table 1: Average portion (in percent) of the total number of links which are traversed by a given number of routes in an O-D pair.

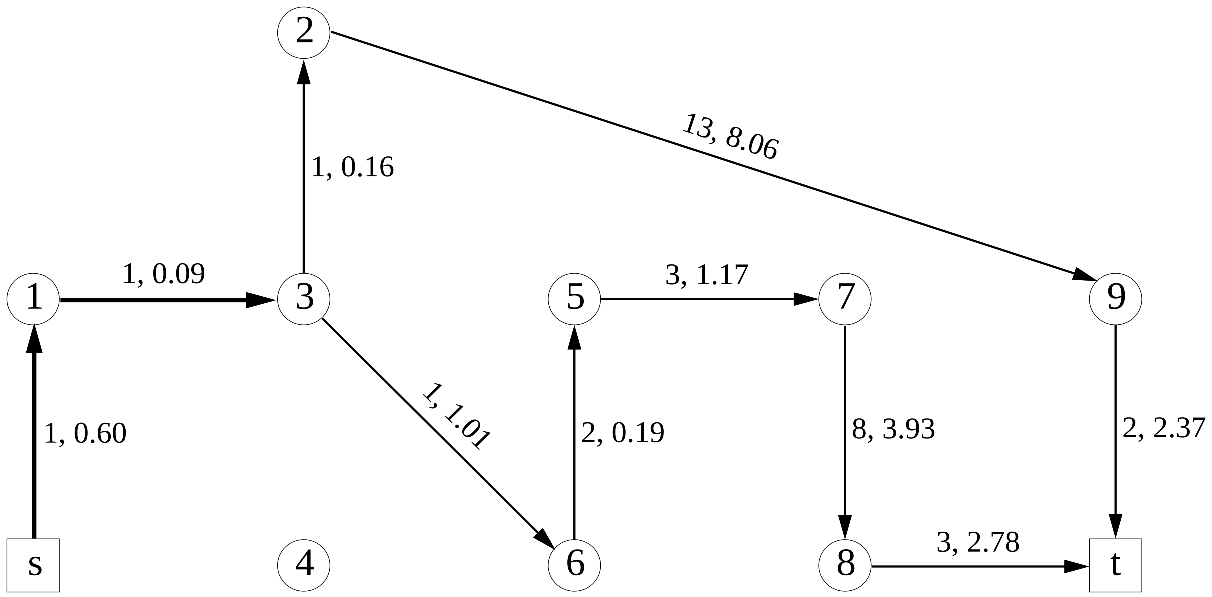
	Value of MO	Number of routes 1	≥ 2	2	3	4	5	6	7
Actual cost	0	100							
	0.25	93	7	7					
	0.50	80	19	16	3				
	0.75	65	35	24	8	2	1		
	1.00	45	55	24	15	9	4	2	1
Simulated cost	0	100							
	0.25	82	17	14	2	1			
	0.50	64	36	24	8	3	1		
	0.75	51	48	25	13	6	3	1	
	1.00	45	53	24	14	8	4	2	1
UE	—	43	57	22	15	9	6	3	2

Table 2: Average portion (in percent) of the total route cost which originates from links which are traversed by a given number of routes in an O-D pair.



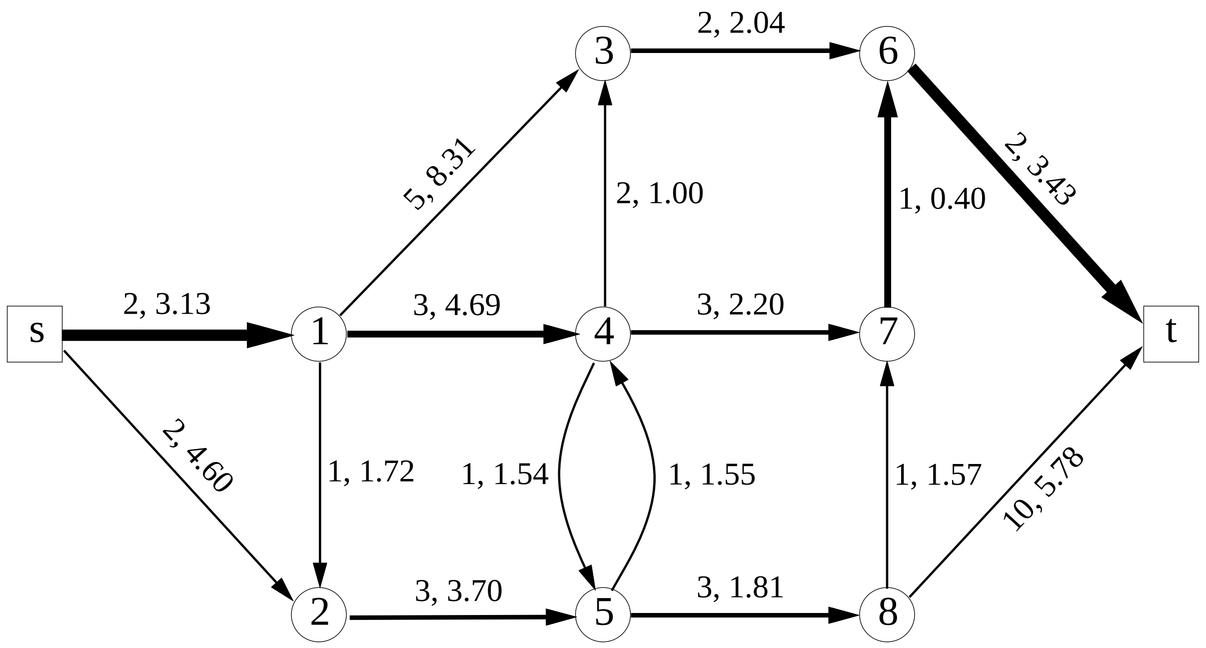
Route	Cost	Portion
s – 1 – 3 – 6 – 5 – 7 – 8 – t	9.77	0.387
s – 1 – 3 – 2 – 5 – 7 – 8 – 9 – t	9.89	0.306
s – 4 – 6 – 7 – 8 – t	10.40	0.101
s – 1 – 4 – 6 – 5 – 7 – 8 – t	10.11	0.193
s – 1 – 3 – 2 – 9 – t	11.49	0.012

Figure 5: Routes generated in the first O-D pair. Simulated link costs. $\Theta = 2.0, MO = 0.75$.



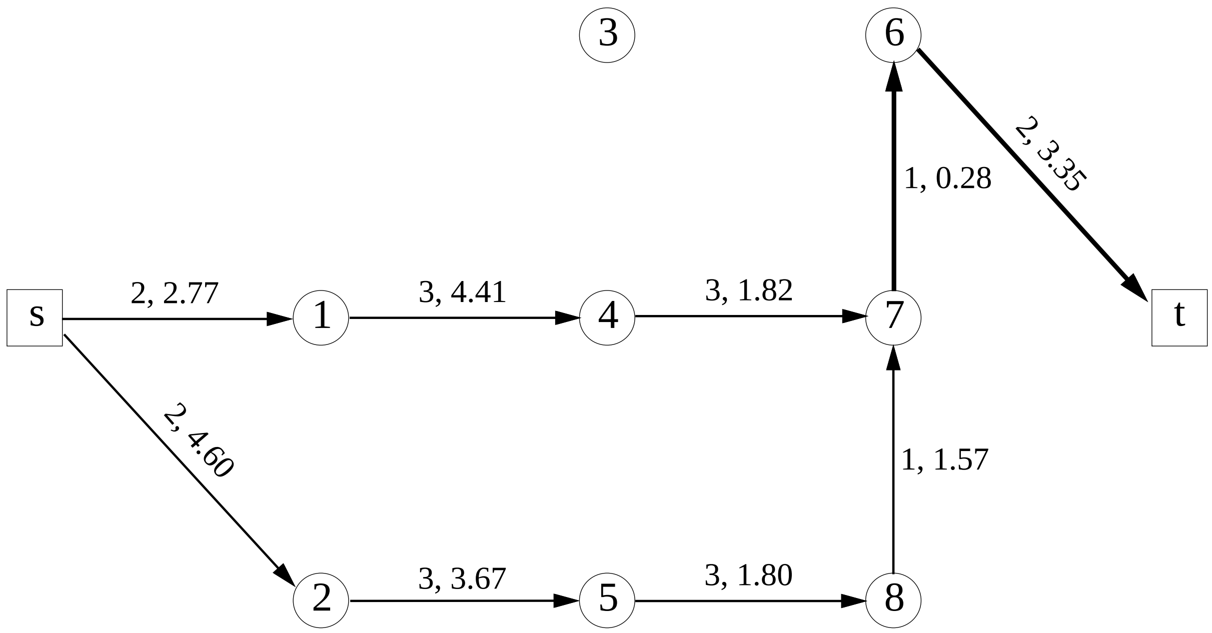
Route	Cost	Portion
s – 1 – 3 – 6 – 5 – 7 – 8 – t	9.77	0.954
s – 1 – 3 – 2 – 9 – t	11.28	0.046

Figure 6: Routes generated in the first O-D pair. Simulated link costs. $\Theta = 2.0, MO = 0.25$.



Route	Cost	Portion
s – 1 – 4 – 7 – 6 – t	13.85	0.230
s – 1 – 4 – 5 – 8 – t	16.95	0.124
s – 1 – 2 – 5 – 4 – 7 – 6 – t	16.13	0.145
s – 1 – 4 – 3 – 6 – t	14.29	0.211
s – 1 – 3 – 6 – t	16.91	0.125
s – 2 – 5 – 8 – 7 – 6 – t	15.51	0.165

Figure 7: Routes generated in the second O-D pair. Simulated link costs. $\Theta = 0.2, MO = 0.75$.



Route	Cost	Portion
s − 1 − 4 − 7 − 6 − t	12.63	0.629
s − 2 − 5 − 8 − 7 − 6 − t	15.27	0.371

Figure 8: Routes generated in the second O-D pair. Simulated link costs. $\Theta = 0.2, MO = 0.25$.