A dynamic O-D matrix estimator using truncated analytic derivaties

Ch.D.R. Lindveld
Faculty of Civil Engineering and Geosciences, Delft University of Technology
Transportation Section
P.O. Box 5048
2600 GA Delft
The Netherlands
Telephone: 31.15.2784912
Fax: 31.15.2783179
E-mail: K.Lindveld@ct.tudelft.nl

1 INTRODUCTION

Static Origin-Destination (O-D) matrix estimation can be formulated as a bi-level programming problem. The upper level problem tries to estimate a matrix that satisfies certain constraints, such as traffic counts, row and column totals, and similarity to an apriori matrix. The lower level problem typically relates the matrix to link flows through an equilibrium assignment.

This framework can be extended to the dynamic case. However computational issues such as convergence and efficiency of the solution algorithms become a serious problem.

One of the central issues is the calculation of a convergent series of descent steps in the upper level of the problem. Recently a solution was presented by [Codina and Montero (2002)] based on the concept of subgradients. Although the use of subgradients is computationally expensive, they report improvements in convergence and in the speed of convergence.

In this paper we propose to alleviate the problem of finding convergent descent steps for the upper level problem in the dynamic case.

We will derive analytic derivatives of the upper level problem $Z(T)$ with respect to the matrix cells, and calculate them using the analytic stochastic dynamic traffic assignment (DTA) method proposed in [Chabini and He (1990)]. This assignment method uses a fixed number of paths per O-D pair, and adopts stochastic route choice. Both characteristics reduce the difficulties in our estimation problem.

2 BACKGROUND

O-D matrix estimation (whether dynamic or static) is an essential part of transportation modelling. Transportation models are used to support infrastructure investment decisions, support what-if analyses on policies, to set fare structures, etc.

In the past 4-5 years we have seen an interest in within-day-dynamic transport modelling, specifically within-day dynamic traffic assignment. The reasons for this interest are:

1) within-day dynamic traffic assignment can be more accurate and realistic in its treatment of congestion, specifically its location and spillback effects;
2) within-day-dynamic transportation models are better suited to support dynamic traffic management (DTM) measures;
3) dynamic traffic assignment, which in recent years has become computationally feasible, is a more natural way of modelling traffic than static assignments.

The drawbacks of dynamic traffic modelling are of a practical nature:

1) the assignment models are just becoming available as operational software packages
2) dynamic demand modelling is not yet as developed as static demand modelling
3) much more data than before are needed to feed and calibrate the models
4) applying dynamic models is more expensive than applying static models because it takes more time and some re-training
5) the visualisation and interpretation of the model results also take more time and effort.

In this paper we address the question of how to estimate one specific input dataset: dynamic O-D matrices. In this we aim at transportation planning applications instead of real-time applications.

In general, O-D matrices are not observable unless the vast majority of people can be tracked on a continuous basis (which incidentally might become a reality with the increasing penetration of cell-phones). Typically observable characteristics of an O-D matrix are production and attraction (e.g. from household surveys public registers), traffic counts (loop detectors, video cameras etc.), partial matrices (e.g. through cordon surveys), and some route choice information (e.g. through licence plate surveys). The challenge is to use these data to estimate a full (dynamic) O-D matrix.

As noted in the literature on static O-D matrix estimation such estimation problems tend to be extremely over-parametrised (see e.g. [Maher (1983)]), which may be addressed through the use of *apriori* information, in the form of apriori O-D matrices. In transportation planning one often has access to reasonable apriori matrices in the form of synthetic O-D matrices.

Synthetic O-D matrices are derived from transportation planning models which can be based on disaggregate mode-destination choice models, possibly supplemented by time-of-day choice. Such models are usually static models that assume equilibrium conditions, which may or may not give an accurate account of the actual situation in the study area.

Despite the drawbacks of such models, one can take the view that such models summarise the available data, and therefore give the best possible apriori estimate of an O-D matrix. By combining a synthetic static O-D matrix with a suitable departure-time choice model, one can arrive at an estimate of an apriori dynamic O-D matrix, or O-D-t matrix.

Given that we have an apriori O-D-t matrix, and that we have sufficient confidence in its structure, we wish to estimate an O-D-t matrix using this apriori matrix and traffic observations.

2.1 Literature overview

Static and dynamic O-D matrix estimation have a similar mathematical structure, and a review of static O-D matrix estimation methods is therefore an obvious step. For a general overview of the issues related to static O-D matrix estimation we refer to [Cascetta (2001)] and [Ortuzar and Willumsen (2001)]; a survey of the literature on static O-D matrix estimation can be found in [Abrahamsson (1998)].

Bi-level programming provides a natural formulation of the static O-D matrix estimation problem, as shown in [Florian and Cen (1991)] and [Yang *et al.* (1992)]. The use of equilibrium assignments as lower-level problem are reported in [Yang *et al.* (1994)], and used in [Yang (1995)], [Maher *et al.* (2001)], [Drissi-Kaitouni and Lundgren (1998)], and [Florian and Cen (1991)].

As is noted in the literature, the O-D matrix estimation problems are easy to formulate but difficult to solve, the main problem being to find a suitable descent direction as noted in [Drissi-Kaitouni and Lundgren (1998)].

Recent advances in calculating suitable descent directions for static O-D matrix estimation are reported in [Patriksson (2001)], [Codina and Montero (2002)], [Codina *et al.* (2002)], and [Lundgren and Peterson (2001)].

Real-time dynamic O-D matrix estimation was reported in [Camus *et al.* (1997)] and in [Ashok and Ben-Akiva (1993)]. A maximum-likelihood approach for the case of a fully monitored road corridor was proposed in [Van der Zijpp (1996)] and successfully applied in [Van der Zijpp (2002)].

The possibility of reducing dynamic O-D-t matrix estimation to static matrix estimation on a Space-Time Expanded Network (STEN) was noted in [Bell and Iida (1997)], and applied in [Lindveld and Van der Zijpp (2000)], and [Van der Zijpp and Lindveld (2000)]. In this case the STEN becomes one of the endogenous variables, which complicates convergence checks.

3 THE DYNAMIC MATRIX ESTIMATION PROBLEM

Like static O-D matrix estimation, the dynamic O-D matrix estimation problem can be stated as a bi-level programming problem in which the upper level minimises an objective function Z with respect to the unknown matrix given by:

$$\hat{\mathbf{T}}^{rs} = \arg\min_{\mathbf{T}^{rs}\geq 0} Z\left(\mathbf{T}^{rs}, \mathbf{T}^{rs,0}, \mathbf{S}_l\left(\mathbf{T}^{rs}\right), \tilde{\mathbf{S}}_l, \mathbf{v}_a\left(\mathbf{T}^{rs}\right), \tilde{\mathbf{v}}_a\right) \quad (1)$$

with $\mathbf{T}^{rs} = \left\{\mathbf{T}_k^{rs}\right\}_{k\in K}$ an O-D-t matrix, $k \in K$ a time period, rs an O-D pair, and a a link and; in which the lower level enforces a dynamic equilibrium flow pattern, represented in a link-based *quasi-VI*[1] link formulation as:

$$\mathbf{c}\left(\mathbf{v}_a^*\right)\left(\mathbf{v}_a - \mathbf{v}_a^*\right) \geq 0 \quad \forall \mathbf{v}_a \in \Omega\left(\mathbf{v}_a^*\right) \quad (2)$$

The variables used are: $\hat{\mathbf{T}}^{rs} = \left\{\hat{T}_k^{rs}\right\}_{k\in K}$ the matrix estimate, $\mathbf{T}^{\mathbf{rs,0}} = \left\{T_k^{rs,0}\right\}_{k\in K}$ the apriori matrix, $\mathbf{S}_l = \left\{S_{kl}\right\}_{k\in K}$ the observable totals, $\tilde{\mathbf{S}}_l = \left\{\tilde{S}_{kl}\right\}_{k\in K}$ the observed totals, $\mathbf{v}_a\left(\mathbf{T}^{rs}\right) = \left\{v_{ak}\left(\mathbf{T}^{rs}\right)\right\}_{k\in K}$ the assigned flow, $\tilde{\mathbf{v}}_a = \left\{\tilde{v}_{ak}\right\}_{k\in K}$ the observed link flow, $\mathbf{v}_a^*$, and $c\left(\mathbf{v}_a^*\right)$ the corresponding link cost. In the following we will drop the vector notation, and use T_k^{rs}, v_{ak}, S_k to denote $\left\{T_k^{rs}\right\}_{k\in K}, \left\{v_{ak}\right\}_{k\in K}, \left\{S_k\right\}_{k\in K}$. The structure of this bi-level problem is shown in Figure 1.

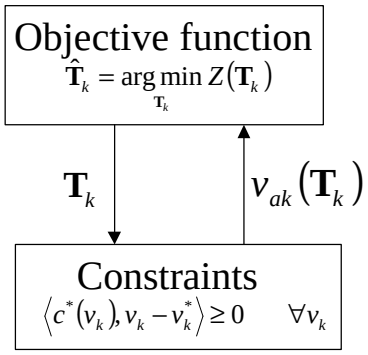


Figure 1: Bi-level formulation of the dynamic matrix estimation problem

A description of dynamic traffic assignment a formulated as a quasi-VI problem can be found in [Chen (1999)] and [Bliemer(2001)]. The Dynamic Traffic Assignment (DTA) method that we are using, is given in [Chabini and He (1998)].

3.1 The objective function

The objective function Z can take various forms: e.g. least squares, maximum-likelihood, or Bayesian estimation. To arrive at an efficient estimation approach, we will split the objective function into three parts: a part $z_0\left(\mathbf{v}_a\left(\mathbf{T}^{rs}\right), \tilde{\mathbf{v}}_a\right)$ that deals with the apriori matrix, an *assignment-independent* part $z_1\left(\mathbf{S}_l, \tilde{\mathbf{S}}_l\right)$ with $S_{kl} = \sum_{(rs)\in I_l} d_{kl}^{rs} T_k^{rs}$, $d_{kl}^{rs} \in \{0,1\}$ representing a group of O-D-t cells (e.g. zonal production or attraction) and an *assignment-dependent* part: $z_2\left(\mathbf{v}_{ap}\left(\mathbf{T}^{rs}\right), \tilde{\mathbf{v}}_{ap}\right)$ (with p a specific path):

[1] In a VI problem the space Ω of feasible solutions is fixed; in (2) the feasible flow pattern depends on the travel-times that result from the flow pattern, so that $\Omega = \Omega(v)$. See [Bliemer (2001)].

$$Z\left(\mathbf{T}^{rs},\tilde{\mathbf{S}}\right)=z_0\left(\mathbf{T}^{rs},\mathbf{T}^{rs,0}\right)+z_1\left(\mathbf{S}_l,\tilde{\mathbf{S}}_l\right)+z_2\left(\mathbf{v}_a\left(\mathbf{T}^{rs}\right),\tilde{\mathbf{v}}_a\right) \quad (3)$$

A possible choice for the objective functions is constrained generalised least-squares with diagonal weighting matrices:

$$z_0\left(\mathbf{T}^{rs}-\mathbf{T}^{rs,0}\right)=\sum_{rsk} w^0_{rsk}\left(T^{rs}_k-T^{rs,0}_k\right)^2 \quad (4)$$

$$z_1\left(\mathbf{S}_l-\mathbf{S}^0_l\right)=\sum_{lk} w^1_{lk}\left(S_{lk}-S^0_{lk}\right)^2=\sum_{lk} w^1_{lk}\left(\sum_{(r,s)\in I_{lk}} T^{rs}_k-S^0_{lk}\right)^2 \quad (5)$$

$$z_2\left(\mathbf{v}_{ap}\left(\mathbf{T}^{r,s}\right),\tilde{\mathbf{v}}_{ap}\right)=\sum_{ak} w^2_{ak}\left(v_{apk}\left(\mathbf{T}^{r,s}\right)-\tilde{v}_{apk}\right)^2 \quad (6)$$

Whereas the GLS estimator can be solved explicitly, we have a CGLS estimator which cannot, (due to the constraint that $T^{rs}_k \geq 0$). An iterative solution algorithm for CGLS estimators is presented in [Bell (1991)].

The structure of the objective function is shown in Figure 2;

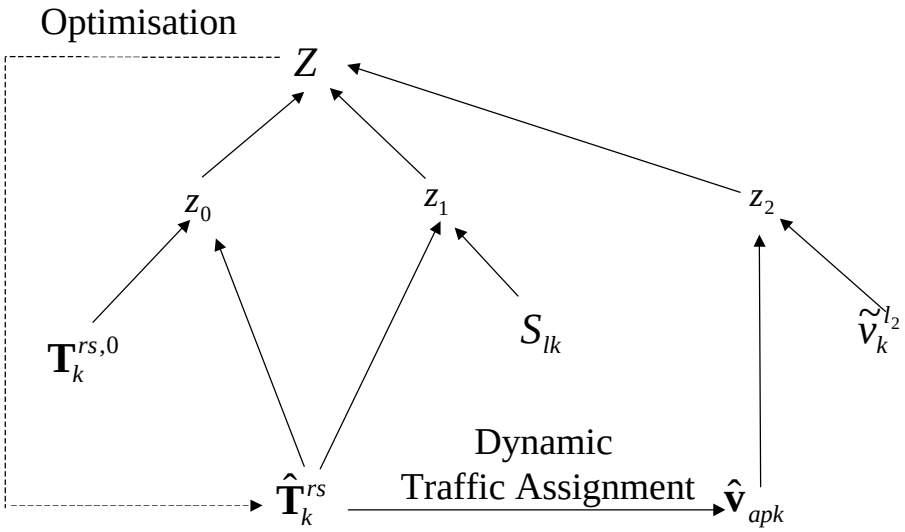


Figure 2: structure of the objective function

3.2 Link with game-theory

As noted in [Yang *et al.* (1992)], [Maher et al. (2001)] and [Florian and Chen (1991)], the bilevel nature of the problem allows one to view the solution as a game-theoretic equilibrium of a game in which the upper level problem (matrix estimation) and the lower level problem (user-equilibrium assignment) are modelled as players. The players take turns in adjusting their action to optimise their criterion function, and can observe the result of the previous player's action.

According to [Basar and Olsder (1999)], if the upper level is unaware of the response of the lower level to changes in the matrix, the game corresponds to a *Cournot* game. If the upper level is completely aware of the response of the lower level to changes in the matrix and anticipates on these, the game corresponds to

a *Stackelberg* leader-follower game, with the upper level as the leader and the lower level as the follower.

As noted in [Yang *et al.* (1992)], the iterative solution of upper- and lower level problems corresponds to a solution algorithm for Cournot games, in which each player tries to maximise his payoff non-cooperatively while assuming that his actions do not affect other player's actions.

As changes to the matrix (the upper level problem) affect the flows in the lower-level problem, the Cournot-type game does not seem an appropriate model, and [Maher et al. (2001)] report that the static matrices estimated using Cournot and Stackelberg games can be different.

Influence factors, as defined in [Yang (1995)], can be used to account for the transport of information from the lower level to the upper level. A possible choice of the influence factors is the Jacobian of the output of the lower level problem (flows) with respect to the control variables of the upper level problem (the matrix cells). From a game theoretic point of view one would expect a Stackelberg game to be over sooner than a Cournot game (all other things being equal); it is hoped that this will lead to more efficient solution methods for dynamic matrix estimation.

4 SOLUTION METHODS

In general, the matrix-estimation problem is a bilevel programming problem, which is generally recognised as difficult.

Rigorous solution methods have been proposed such as the penalty methods described in [Marcotte and Zhu (1996)] and the half-space projection described in [Clegg and Smith (2001)].

Heuristic solution methods have been proposed in [Yang *et al.* (1992)], [Yang (1995)], and [Maher *et al.* (2001)].

4.1 Rigorous solution methods

Penalty methods

The bilevel problem can be solved directly using penalty methods. The penalty methods proposed in [Marcotte and Zhu (1996)] work by defining a penalised optimisation problem with the following objective function:

$$Z_{m_k} = Z + m_k E \tag{7}$$

with Z the upper-level objective function, E a gap-function associated with the lower-level objective function, and m_k a weight, and solving a sequence of such problems defined by a sequence of weights $\{m_k\}$ that tends to infinity.

The half-space projection method

If Z is the upper level objective function, and E is the lower level objective function (giving a measure of disequilibrium of the traffic assignment: $E = 0$ when equilibrium is reached), and a possible function for E is given in [Smith (1984)].

The idea is to determine a simultaneous descent direction for the upper and the lower problem, but to give precedence to the minimisation of the lower problem (the assignment) as long as its objective function exceeds a certain theshold e.

The simultaneous descent direction is defined as:

$$d_e = \begin{cases} (1-l)\dfrac{\nabla E}{\|\nabla E\|} - lP_H\left(\dfrac{\nabla Z}{\|\nabla Z\|}\right) & if \quad E > e \\ -P_H\left(\dfrac{\nabla Z}{\|\nabla Z\|}\right) & if \quad E = e \\ -(1-l)\dfrac{\nabla Z}{\|\nabla Z\|} - lP_H\left(\dfrac{\nabla Z}{\|\nabla Z\|}\right) & if \quad E < e \end{cases} \tag{8}$$

with $l = \dfrac{E}{e}$, and $P_H\left(\dfrac{\nabla Z}{\|\nabla Z\|}\right)$ the projection of $\dfrac{\nabla Z}{\|\nabla Z\|}$ on the half-space H defined by $\dfrac{\nabla E}{\|\nabla E\|}$:

$$H = \left\{ d \,\middle|\, \left\langle d, \frac{\nabla E}{\|\nabla E\|} \right\rangle \geq 0 \right\} \tag{9}$$

and e a threshold. For each value of e a minimisation algorithm is applied with d_e as the search direction. The final solution is obtained by sending e to 0.

4.2 Heuristic solution methods

A heuristic algorithm proposed in [Yang *et al.* (1992)] alternatively minimises the upper or the lower level problem while holding the other one constant; this iterative optimisation-assignment algorithm is the classical transportation planning approach towards O-D matrix estimation. This heuristic corresponds to a Cournot game.

A refinement consists of linearising the map that relates flows and the O-D matrix:

$$\mathbf{v}(\mathbf{T}) \approx \mathbf{v}(\mathbf{T}^*) + \mathbf{A}(\mathbf{T} - \mathbf{T}^*) \tag{10}$$

A natural choice for the matrix of influence factors is the Jacobian of the assignment map (although this is complicated to calculate).

In [Maher *et al.* (2001)] the difference of the flows and matrices is used to produce the linearisation.

From (8) the need for gradients of the objective function Z is clear. Recalling that the objective function Z is a function of the O-D-t matrix cells T_k^{rs}, we need to calculate the derivatives:

$$(\nabla Z)_k^{rs} = \frac{\partial Z(T_k^{rs})}{\partial T_k^{rs}} = \frac{\partial z_0(T_k^{rs})}{\partial T_k^{rs}} + \frac{\partial z_1(T_k^{rs})}{\partial T_k^{rs}} + \frac{\partial z_2(T_k^{rs})}{\partial T_k^{rs}} \tag{11}$$

with: L_k^{rs} the set of all parameters that act on matrix cell T_k^{rs}.

The resulting derivatives are:

$$\frac{\partial z_0(T_k^{rs})}{\partial T_k^{rs}} = 2w_{rsk}^0 (T_k^{rs,0} - T_k^{rs}) \tag{12}$$

$$\frac{\partial z_1(S_{lk})}{\partial T_k^{rs}} = \sum_{lk|(r,s)\in I_{lk}} \frac{\partial z_0(S_{lk})}{\partial S_{lk}} \frac{\partial S_{lk}}{\partial T_k^{rs}} = \sum_{lk} 2w_{lk}^1 (S_{lk} - S_{lk}^0) d_{lk}^{rs} \tag{13}$$

$$\frac{\partial z_2(T_k^{rs})}{\partial T_k^{rs}} = \frac{\partial z_2(v_{ak}(T_k^{rs}), \tilde{v}_{ak})}{\partial v_{ak}} \frac{\partial v_{ak}(T_k^{rs})}{\partial T_k^{rs}} \tag{14}$$

The first and second terms do not involve traffic flows, and can therefore be calculated with comparative ease. The third term part of the derivative of the objective function involves the derivative of path flow w.r.t. to the matrix cells $\frac{\partial v_{ak}(T_k^{rs})}{\partial T_k^{rs}}$, and is much harder. In order to derive expressions for $\frac{\partial v_{ak}(T_k^{rs})}{\partial T_k^{rs}}$ we will first present the analytic dynamic assignment procedure we use, and use this to calculate (truncated) expressions for the derivative in chapter 5.

5 THE DYNAMIC TRAFFIC ASSIGNMENT PROCEDURE

Several implementations of DTA models exist that are based on micro-simulation (e.g. DYNAMIT, INTEGRATION, AIMSUN), packet simulation (CONTRAM) and at least one that uses macroscopic traffic simulation (METANET). However, we will be using an analytic discrete-time Dynamic Traffic Assignment (DTA) procedure to model the relationship between dynamic traffic flows and the O-D-t matrix. The advantage of an analytic assignment procedure is that is easier to obtain derivatives of the traffic flow w.r.t. changes in the O-D-t matrix.

We will adopt the DTA procedure described in [Chabini and He (1998)]. This DTA procedure uses a fixed limited set of paths, and must therefore be expected to yield an *approximation* of the true user-equilibrium, except in networks with very few route alternatives. We are addressing this issue heuristically by determining an initial path-set using a Monte-Carlo approach on the unloaded network followed by several iterations of dynamic network assignment followed by shortest path search on the loaded dynamic network after assignment.

For dynamic O-D matrix estimation, dynamic link flows are needed, as would result from the solution of (2). This quasi-VI problem however must solve the route-choice and traffic propagation problem simultaneously, and is harder to solve than a combination of a *path-based* VI formulation for route choice combined with a special-purpose Dynamic Network Loading (DNL) algorithm.

The DTA proposed in [Chabini and He (1998)] uses this insight, and consists of a Stochastic Route-Choice Model and an analytic Dynamic Network Loading part (DNL).

5.1 The stochastic route-choice model of the DTA

The route-choice model assumes a stochastic user-equilibrium: for each O-D pair (r,s) at any time t, the perceived experienced travel time of a path that is chosen equals the minimum perceived experienced travel time. In the implementation we use, the path-choice probabilities are derived from a C-logit model, but the model (and the software) can work with more sophisticated route-choice models.

Given the path costs and the demand, the user-equilibrium path flow f_{kp}^{rs} is related to the O-D demand T_k^{rs} and the route-choice probability P_{pk}^{rs} as:

$$f_{pk}^{rs} = T_k^{rs} P_{pk}^{rs} \tag{15}$$

A dynamic generalization of the conventional static user-optimal (Wardrop) condition with path travel time defined as experienced (actual) travel time. The condition can be expressed as follows:

$$c_{pk}^{*rs} \geq p_k^{rs} \tag{16}$$

$$f_{pk}^{rs}\left[c_{pk}^{*rs} - p_k^{rs}\right] = 0 \tag{17}$$

$$f_{pk}^{rs} \geq 0 \tag{18}$$

The users' route choice behaviour model can be formulated (see [Chabini and He (1998)]) as an equivalent Variational Inequality (VI) problem:

$$\sum_{k=0}^{K} \sum_{rs} \sum_{p} F_{pk}^{*rs}\left[f_{pk}^{rs} - f_{pk}^{*rs}\right] \geq 0 \tag{19}$$

with:

$$F_{pk}^{*rs} = \left[f_{pk}^{*rs} - f_{pk}^{rs}\right] P_{pk}^{*rs} \frac{\partial c_{pk}^{rs}}{\partial f_{pk}^{rs}} \tag{20}$$

where the route costs depend on the travel time, wich in turn depends on the link travel times that result from the DNL. If we assume that $\frac{\partial c_{pk}^{rs}}{\partial f_{pk}^{rs}} > 0 \quad \forall_{r,s,p,k}$, then

we have that $\left[f_{pk}^{*rs} - f_{pk}^{rs}\right]P_{pk}^{*rs} = 0$ when $F_{pk}^{*rs} = \left[f_{pk}^{*rs} - f_{pk}^{rs}\right]P_{pk}^{*rs}\frac{\partial c_{pk}^{rs}}{\partial f_{pk}^{rs}}$. Therefore the equilibrium path flow is:

$$f_{kp}^{*rs} = T_k^{rs} P_{pk}^{*rs} \tag{21}$$

The route choice probabilities P_{pk}^{*rs} are calculated through a C-logit model to account for route-overlap (see [Cascetta (2001)]):

$$P_{pk}^{*rs}\frac{e^{V_{pk}^{rs} - CF_{pk}}}{\sum_q e^{V_{qk}^{rs} - CF_{qk}}} \tag{22}$$

with CF_{pk} the commonality factor of all routes *p* and all other routes *q* between *r* and *s*, and V_{pk}^{rs} the systematic part of the path utility function. Although the framework supports general forms of V_{pk}^{rs}, we will use the simplest form possible:

$$V_{pk}^{rs} = \sum_{a\in p} t_{ak}\left(X_{ak}\right) \tag{23}$$

with X_{ak} the total amount of traffic on link a during period k, as defined below.

5.2 The DNL

As noted in [Chabini (2001)] if the travel time on individual links during each time interval is known, then the flows can be constructed through simple network loading. However if the flows depend on the network conditions then the link travel times that result from the loaded network may not be consistent with those used to load the network in the first place. In other words a fixed-point problem results.

The DNL algorithm (called the “C-load” algorithm) listed below calculates a consistent set of flows and travel times is a single pass. The limitation of this method is that it is a path-based DNL that uses a fixed number of paths.

The discrete-time DNL works with two types of intervals: long intervals Δ, with

$$\Delta = \min_{a\in A} t_a^{free-flow}, \tag{24}$$

and short intervals d

$$d = \frac{\Delta}{M} \tag{25}$$

M a positive integer

The amount of traffic on a link at time instant kd: $X_{ap}^{rs}(kd)$ is the difference between the cumulative inflow $U_{ap}^{rs}(kd)$ and the cumulative outflow $V_{ap}^{rs}(kd)$ at that time, given by:

$$X_{ap}^{rs}(kd) = U_{ap}^{rs}(kd) - V_{ap}^{rs}(kd) \tag{26}$$

The inflow rate on a link is the path flow for the first link on a path, and the outflow of its upstream neighbour for all other links:

$$u_{ap}^{rs}(kd) = \begin{cases} f_{ap}^{rs}(kd) & if \quad \text{a is first link on path p} \\ v_{a'p}^{rs}(kd) & if \quad \text{a follows directly after a'} \end{cases} \tag{27}$$

The cumulative inflows are related to the inflow rates as:

$$U_{ap}^{rs}(kd) = \sum_{j=0}^{k} u_{ap}^{rs}(jd)d \tag{28}$$

The cumulative outflow during period k consists of the inflows of all time periods l that exit no later than period k. This means that the time of entry plus the travel time at the time of entry does not exceed k: $d + t_a(ld) \le (k-1)d$. This leads to:

$$V_{ap}^{rs}(kd) = \sum_{j \in J_a(k)} u_{ap}^{rs}(jd)d \tag{29}$$

With the index set: $J_a(k)$defined as:

$$J_a(k) = \left\{ l \middle| 0 \le ld + t_a(ld) \le (k-1)d \right\} \tag{30}$$

The link travel time during time period kd is given by $t_a(kd)$, which is related to the total traffic flow on the link $X_{ap}^{rs}(kd)$ through a link-performance function defined by $D_a\left(X_a(kd)\right)$:

$$t_a(kd) = D_a\left(X_a(kd)\right) \tag{31}$$

The total amount of traffic on link a $X_a(kd)$ is the sum of the path-flow related amounts of traffic:

$$X_a(kd) = \sum_{rsp} X_{ap}^{rs}(kd) \tag{32}$$

$$X_{ap}^{rs}(0) = U_{ap}^{rs}(0) = V_{ap}^{rs}(0) = 0 \quad \forall_{r,s,a,p} \tag{33}$$

The link outflow rate, formally the derivative of the cumulative link outflow rate, is approximated as:

$$v_{ap}^{rs}(kd) = \frac{V_{ap}^{rs}(kd) - V_{ap}^{rs}((k-1)d)}{d} \tag{34}$$

5.3 The structure of the DTA as a combination of route-choice and DNL

The relationship between these variables within the complete DTA is illustrated in Figure 3: At the to we have the O-D-t matrix T_k^{rs}, which is split into route-flows

f_{kp}^{rs} through the application of route-choice probabilities P_{kp}^{rs}. The route flows f_{kp}^{rs} yield route-specific link flows u_{ak}^{rs} when multiplied by the dynamic path-link incidence variables d_{apk}^{rs} that result from the DNL level of the model.

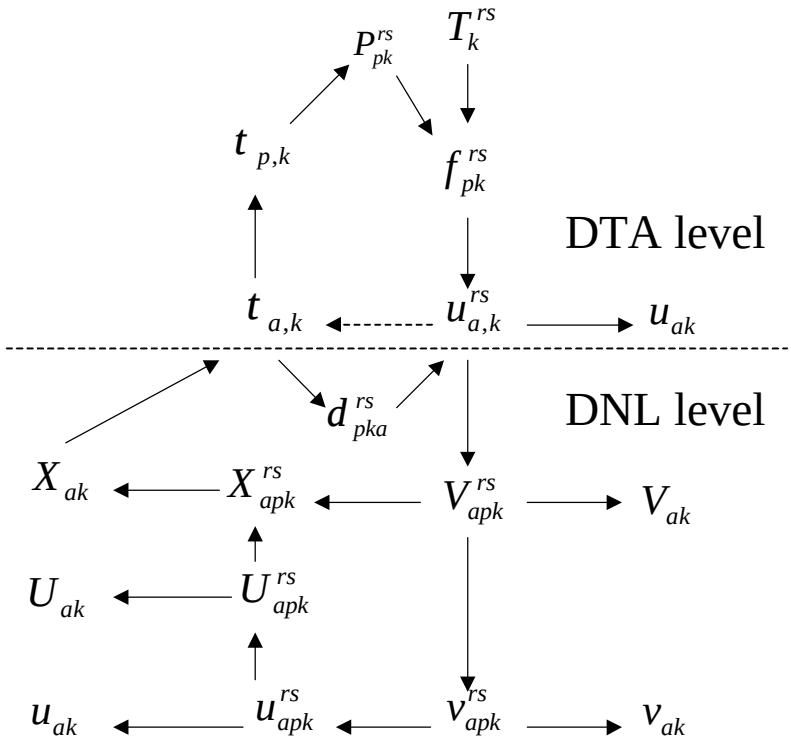


Figure 3: structure of the DTA model proposed by Chabini and He

At the DNL level we have the calculation of cumulative outflows $V_{ap}^{rs}(kd)$, which lead to link outflow rates $v_{ap}^{rs}(kd)$, which define the inflows $u_{ap}^{rs}(kd)$ of downstream links. The link inflow rates can be summed to obtain the cumulative link inflows $U_{ap}^{rs}(kd)$, which combine with the cumulative link inflows to yield the amount of traffic $X_{ap}^{rs}(kd)$ on each link in each time interval per path. The path-dependent traffic loads can be summed to obtain the total link loads $X_a(kd)$, which determine the link travel times $t_a(kd)$, which finally determine the dynamic link-path incidence matrix d_{pka}^{rs} and the path travel times $t_a^p(kd)$.

The DTA procedure yields dynamic link traffic flow rates $v_{ap}^{rs}(kd)$ which correspond to observable quantities.

5.4 Relevance of the DTA description for the O-D-t estimation problem

As the DTA algorithm has in itself a bilevel structure with the route choice as upper level problem and the DNL as lower level problem, the dynamic O-D matrix

estimation problem has *three levels* instead of the usual two in the case of static O-D estimation. This complicates the calculation of a descent direction for the O-D matrix estimation problem.

However, when the lower level conditions are satisfied, a linear relationship between O-D-t matrix cells and dynamic link flows results:

$$v_{ak}\left(T_k^{rs}\right) = \sum_{rsp} a_{apk}^{rs} T_k^{rs} \qquad (35)$$

6 CALCULATION OF TRUNCATED DERIVATIVES OF THE OBJECTIVE FUNCTION

We propose to calculate truncated values of the linkflow derivatives wirth respect to the matrix cells using a sensitivity analysis of the link flows. Sensitivity analyses are reported in [Patriksson, (2001)], [Tobin and Friesz (1988)], and [Bell and Iida (1997)].

We can simplify these calculations by making a number of assumptions:

1. the number of paths per O-D pair is limited, and given in advance
2. stochastic user-equilibrium assignment is used throughout
3. absense of constraints
4. absense of back-blocking; we have vertical queuing

The first assumption ensures that we will not have to worry about additional paths becoming active as the O-D flow changes: all paths receive nonzero flow regardless of their level of service. This assumption can be relaxed by searching for new paths after the assignment has finished.

The second assumption ensures that the path flow pattern corresponding to user-optimality is unique, and that we will not have to deal with the issue of selecting sets of pathflows. This is not restrictive.

The third assumption ensures that we will not have to worry about the effect of the constraints on the derivatives. This is a restrictive assumption.

The fourth assumption frees us from greater interdependence between link flows and matrix cells caused by back-blocking. This restricts the area of application to cases with limited congestion.

6.1 Relating link flows to path flows

The derivatives $\frac{\partial v_{ap}^{rs}(kd)}{\partial T_h^{rs}}$ can be calculated in a number of ways, depending on the amount of information on the behaviour of the second and third level that we assume present in the toplevel.

As the DNL gives path-specific link flows, we can start by decomposing the derivative of the total link flow into the derivatives of the path-specific link flows:

$$\frac{\partial v_a(kd)}{\partial T_h^{rs}} = \sum_{rs}\sum_{p} \frac{\partial v_{ap}^{rs}(kd)}{\partial T_h^{rs}} \tag{36}$$

Using (34), we can write:

$$\frac{\partial v_{ap}^{rs}(kd)}{\partial T_h^{rs}} = \sum_{j\in J} \frac{\dfrac{\partial V_{ap}^{rs}(jd)}{\partial T_h^{rs}} - \dfrac{\partial V_{ap}^{rs}((j-1)d)}{\partial T_h^{rs}}}{d} \tag{37}$$

and using (29) we can write :

$$\frac{\partial V_{ap}^{rs}(jd)}{\partial T_h^{rs}} = \sum_{j} \frac{\partial u_{ap}^{rs}(jd)}{\partial T_h^{rs}}$$

Because of (27) we can write:

$$\frac{\partial u_{ap}^{rs}(jd)}{\partial T_h^{rs}} = \begin{cases} \dfrac{\partial v_{a^- p}^{rs}(jd)}{\partial T_h^{rs}} & if \qquad a^- \text{ preceeds } a \\ \dfrac{\partial f_{ap}^{rs}(jd)}{\partial T_h^{rs}} & if \quad a \text{ is the first link on the path} \end{cases}$$

This means that almost all of the terms $\frac{\partial u_{ap}^{rs}(jd)}{\partial T_h^{rs}}$ can be calculated recursively, and what remains is to take the derivative of the path flow at the entrance of the path. From (15), the equilibrium flow is $f_{kp}^{*rs}(j) = T_k^{rs} P_{pk}^{rs}$, so that:

$$\frac{\partial f_{jp}^{rs}}{\partial T_h^{rs}} = \frac{\partial\left(P_{jh}^{rs} T_h^{rs}\right)}{\partial T_h^{rs}} = T_h^{rs} \frac{\partial P_{jh}^{rs}}{\partial T_h^{rs}} + P_{jk}^{rs} d_{jh} \tag{38}$$

6.2 Proportional assignment: a partial Stackelberg game

If we assume that the upper level has no information about the response of the lower levels, we in fact assume proportional assignment corresponding to a Cournot game. The form of the derivatives can be read off from Figure 4:

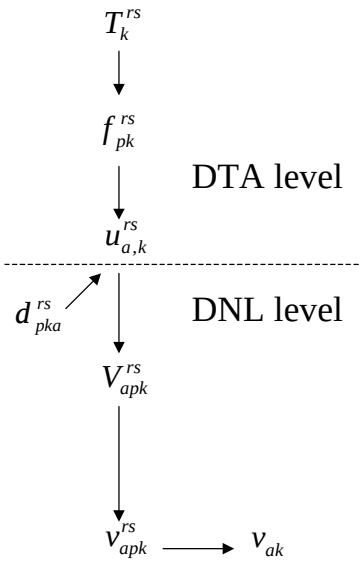


Figure 4: influence diagram for proportional assignment

The derivative is:

$$\frac{\partial v_{ap}^{rs}(kd)}{\partial T_h^{rs}} = \sum_{j \in J} j_{aph}^{rs}(jd) P_{ap}^{rs}(jd) \tag{39}$$

Which is to assume proportional assignment in the upper level problem.

$$\frac{\partial v_a(kd)}{\partial T_h^{rs}} = \sum_{rs} \sum_{p} \sum_{j \in J} j_{aph}^{rs}(jd) P_{ap}^{rs}(jd) \tag{40}$$

6.3 Other effects considered: towards a full Stackelberg game

By incorporating the effects of the feedback loops shown in Figure 3, other terms will appear in the derivative. Their calculation is more involved and will be reported in a subsequent paper.

6.4 Calculation of dynamic assignment fractions

Due to the way the DNL algorithm is set up, no information is available regarding the precise O-D-t cells that contribute to a particular link flow. However, this information can be retrieved from the output of the DTA as described below. Shown in Figure 5 are the trajectories of individual O-D-t cells as they are assigned. The assignment fractions can be read off from the diagram

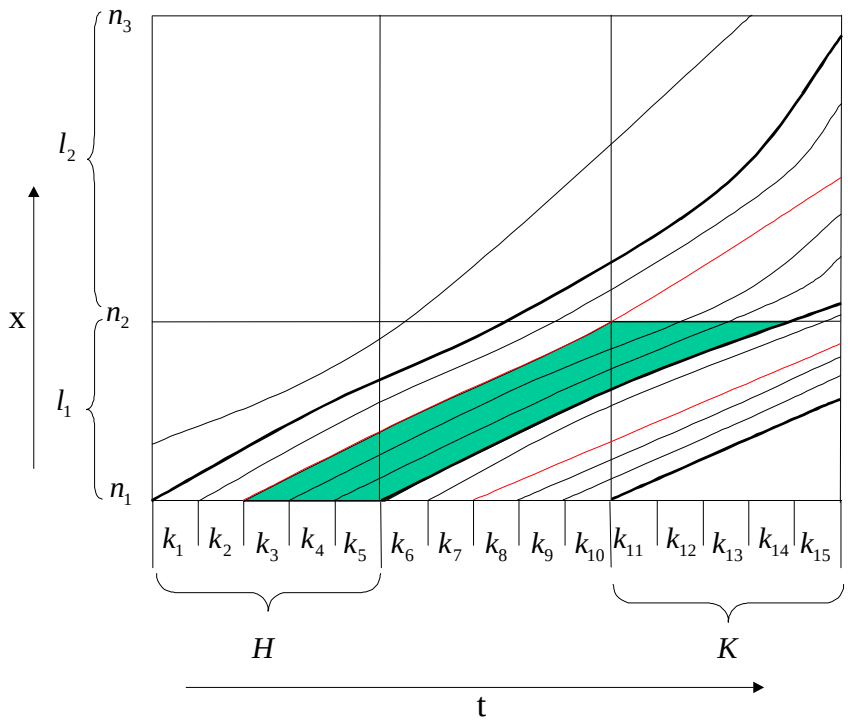


Figure 5: trajectories in the DTA

The relationship between the link outflows and the matrix cells is given by eqn (), with the assignment fractions j_{aphk}^{rs} defined as:

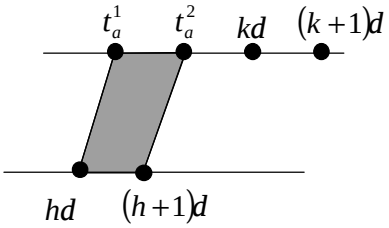
$$j_{aphk}^{rs} = \begin{cases} 0 & \text{if the pathflow from period } h \text{ does not touch link } a \text{ during period } k \\ j \in (0,1) & \text{if the pathflow from period } h \text{ partially touches link } a \text{ during period } k \\ 1 & \text{if the pathflow from period } h \text{ remains on link } a \text{ during period } k \end{cases}$$

Defining :

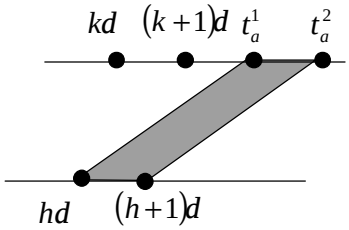
$$t_a^1 = hd + t_{pa}(hd)$$

$$t_a^1 = (h+1)d + t_{pa}((h+1)d)$$

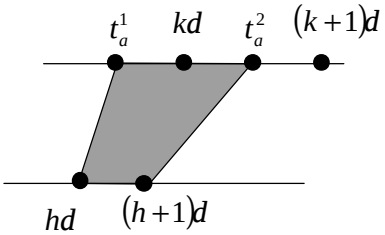
The following six possibilities exist:



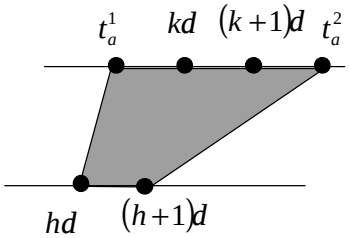
(I)



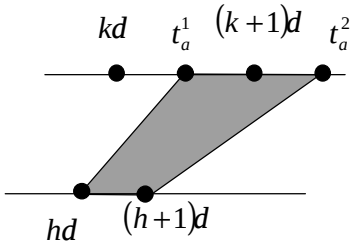
(II)



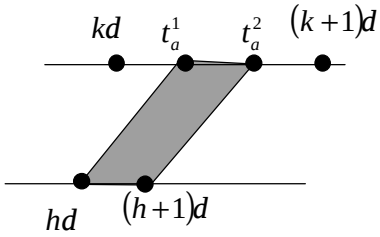
(III)



(IV)



(V)



(VI)

This gives the following assignment fractions:

$$j_{aphk}^{rs} = \begin{cases} 0 & if \quad t_a^2 < kd & (I) \\ 0 & if \quad t_a^1 > (k+1)d & (II) \\ \dfrac{t_a^2 - kd}{t_a^2 - t_a^1} & if \quad \left(t_a^1 < kd\right) \wedge \left(kd < t_a^2 < (k+1)d\right) & (III) \\ \dfrac{t_a^2 - t_a^1}{d} & if \quad \left(t_a^1 < kd\right) \wedge \left(t_a^2 > (k+1)d\right) & (IV) \\ \dfrac{(k+1)d - t_a^1}{t_a^2 - t_a^1} & if \quad \left(kd < t_a^1 < (k+1)d\right) \wedge \left(t_a^2 > (k+1)d\right) & (V) \\ 1 & if \quad \left(kd < t_a^1 < (k+1)d\right) \wedge \left(kd < t_a^2 < (k+1)d\right) & (VI) \end{cases} \tag{41}$$

The assignment fractions can be calculated as follows:

```
Step 0: (Carry out DTA)
     Complete DTA algorithm;
Step 1: (Calculate assignment fractions)
for i do
     for (r,s) do
          for  p ∈ K_rs  do
                    while  a ∈ p  do
                    [t_in , t_out ] =
                      Calculate_arrival_and_departure_times;
                    Calculate_assignment_fractions;
                    Store_assignment_fractions_in_cells;
                    a = a->nextlink;
                    endwhile ap
          endfor  p ∈ K_rs
     endfor (r,s)
endfor i
```

7 DISCUSSION AND CONCLUSIONS

In this paper we have presented a method for dynamic O-D matrix estimation based on an analytic stochastic path-based DTA procedure. The method has certain restrictions: a fixed number of paths, and no (capacity) constraints.

Due to the fact that the DTA is analytic and the restrictions that we impose, we can calculate the gradient of assigned link-flows w.r.t. O-D-t matrix cells. This allows us –in principle- to solve the dynamic O-D matrix estimation problem that corresponds to a Stackelberg game.

We note however that if we are prepared to forego an exact solution of the D-T matrix estimation problem in favour of *different* but related problem formulation that corresponds to a Cournot game, that problem can be solved using derivative-free methods, but may take more iterations of the DTA procedure.

We hope to be able to present numerical results during the presentation of this paper.

8 ACKNOWLEDGEMENT

We are indebted to professor Ismail Chabini for making the code of the DTA procedure described in [Chabini and He (1998)] available to us for research purposes.

9 REFERENCES

Abrahamsson, T. (1998) Estimation of Origin-Destination matrices using traffic counts - a literature survey.

Basar, T., Olsder, G. (1999) *Dynamic noncooperative game theory*. Second edition. Classics in applied Mathematics **23**. SIAM

Bell, M. (1991) The estimation of origin-destination matrices by constrained generalised least squares. Transpn. Res. B 25, 1, 13-22.

Bell, M., Iida, Y. (1997) Transportation network analysis. Wiley.

Bliemer, M. (2001) *Analytical dynamic traffic assignment with interacting user-classes*. Ph.D. thesis Delft University of Technology.

Camus, R., Cantarella, G., Inaudi, D. (1997) Real-time estimation and prediction of origin-destination matrices per time slice. Intl. Jnl. of forecasting 13, (1997), 13-19.

Cascetta, E. (2001) Transportation systems engineering: theory and methods. Kluwer Academic Publishers.

Chen, H. (1999) *Dynamic travel choice models. A variational inequality approach*. Springer verlag.

Clegg, J., Smith, M.J. (2001) Cone projection versus half-space projection for the bilevel optimisation of transportation networks. Transp. Res. B 35 (2001) 71-82.

Chabini, I. (2001) The analytical network loading problem: formulation, solution algorithms and computer implementations. Transp. Res. Rec. 1771,191-200.

Chabini, I., Y. He (1990) "A Flow-Based Approach to the Dynamic Traffic Assignment Problem: Formulations, Algorithms and Computer Implementations", Proceedings of TRISTAN III Conference, San Juan, Puerto Rico, June 17-23, 1998, Vol. 2

Codina, E., Garcia, R., Marin, A. (2002) New algorithmic alternatives for the OD matrix adjustment problem on traffic networks. Proceedings of the 13th Mini-Euro conference and the 9th Meeting of the EURO working group on Transportation, Bari 2002, pp. 421-425.

Codina, E., Montero, L. (2002) A method to approximate the steepest descent direction of the O-D matrix adjustment problem. Proceedings of the TRISTAN IV triennial symposium on transportation analysis.

Drissi-Kaitouni, O., Lundgren, J. (1998) A descent algorithm for the bi-level origin-destination matrix estimation problem. Research report LITH-MAT-R-1992-49 (revised version). Linkoping Institute of Technology, Sweden.

Florian, M., Chen, Y. (1991) A bilevel approach to estimating O-D matrix by traffic counts. C.R.T. research report no. C7PQMR PO750-X.

Gibbons, R. (1992) *Game theory for applied economists*. Princeton University Press.

Lindveld, Ch.D.R., van der Zijpp, N. (2000) Estimation of OD demand demand for dynamic assigment with route choice and departure time choice. PTRC conference 2000.

Lundgren, J., Peterson, A. (2002) An heuristic for the bilevel origin destination matrix estimation problem. . Proceedings of the 13th Mini-Euro conference and the 9th Meeting of the EURO working group on Transportation, Bari 2002, pp 647-651.

Maher, M. (1983) Inferences on trip matrices from observations on link volumes: a bayesian statistical approach. Transpn. Res. B 17B, 6, 435-447.

Maher, M.J., Zhang, X., Van Vliet, D. (2001) A bi-level programming approach for trip matrix estimation and traffic control problems with stochastic user equilibrium link flows. Transp. Res. B 35 23-40

Marcotte, P., Zhu. D.L. (1996) Exact and inexact penalty methods for the generalized bilevel programming problem. Mathematical Programming 74 (1996) 141-157

Ortuzar, J., Willumsen, L. (2001) Modelling Transport. Third edition. Wiley.

Patriksson, M. (2001) Sensitivity analysis of traffic equilibria. Paper submitted to Transportation Science.

Sheffi, Y., (1985) Urban transportation networks: Equilibrium analysis with mathematical programming methods. Prentice Hall.

Smith, M.J. (1984) A descent algorithm for solving a variety of monotone equilibrium problems. In: Proceedings of the ninth International Symposium on Transportation and Traffic Theory, The Netherlands, VNU Science Press, Utrecht, pp. 273-297.

Tobin, R., Friesz, T (1988) Sensitivity analysis for equilibrium network flow. Transportation Science, 22 (4) 242-250.

Van der Zijpp (1996) Dynamic Origin-Destination matrix estimation on motorway networks. Ph.D. Thesis, Delft University of Technology.

Van der Zijpp (2002) Novel methods for O/D matrix estimation phase 1: Estimation results for the SISTM and KENT datasets. Research report, Delft University of Technology.

Willumsen, L.G. (1984) Estimating time-dependent trip matrices from traffic counts. In: Proceedings of the ninth International Symposium on Transportation and Traffic Theory, The Netherlands, VNU Science Press, Utrecht, pp. 273-297.

Yang, H. (1995) Heuristic algorithms for the bi-level origin-destination matrix estimation problem. Transpn. Res. B. 29B, 4, 231-242.

Yang, H., Iida, Y., Sasaki, T. (1994) The equilibrium-based origin-destination matrix estimation problem. Transpn. Res. B. 28B, 1, 23-33

Yang, H., Sasaki, T., Iida, Y., Asakura, Y. (1992) Estimation of origin-destination matrices from link traffic counts on congested networks. Transpn. Res. B 26B, 6, 417-434.

Van der Zijpp, N., Lindveld, K. (2000) Estimation of Origin-Destination Demand for Dynamic Assignment with Simultaneous Route and Departure Time Choice. Transportation Research Record No. 1771, 75-82.