Matrix estimation for static traffic assignment models with queuing

Brederode, Luuk; Pel, Adam; Hoogendoorn, Serge

Publication date
2014

Published in
hEART 2014 - 3rd symposium of the European association for research of transportation, Leeds UK

Citation (APA)
Brederode, L., Pel, A., & Hoogendoorn, S. (2014). Matrix estimation for static traffic assignment models with queuing. In *hEART 2014 - 3rd symposium of the European association for research of transportation, Leeds UK*

Paper presented at the 3rd Symposium of the European Association for Research in Transportation, 10-12 September 2014, ITS Leeds

Matrix estimation for static traffic assignment models with queuing

Luuk Brederode [a,b]*, Adam John Pel [a], Serge Paul Hoogendoorn[a]

[a] *Delft University of Technology, Faculty of Civil Engineering and Geosciences, Stevinweg 1, 2600 GA, Delft, the Netherlands*
[b] *DAT.mobility , Snipperlingsdijk 4, 7414 BJ, Deventer, the Netherlands*

Abstract

A matrix estimation method using the semi dynamic assignment model STAQ is developed exploiting its methodological advantages over full DTA models. The matrix estimation problem is formulated as a bi-level problem and is solved on the node level taking flow metering into account. In the lower level the method uses marginal simulation of the node model within the assignment model to approximate the response function. The implicit relations between turn demand and link flows as defined by the directional capacity proportional node model are analyzed and made explicit. In the upper level an objective function minimizing differences between estimated and observed link flows and differences between prior and posterior ODmatrices is used, both components using a MSE distance function. The two components in the objective function are weighted and normalized. A method to prevent overshooting due to approximation errors is proposed as well as a method to correct the prior ODmatrix in case of insensitivity of the link flow due to supply constraints inconsistent with observed link flows. Test runs are conducted showing that the method finds (non-unique) solutions to the matrix estimation problem when only differences in link flows are taken into account, but may fail to converge when also differences between prior and estimated ODmatrix are taken into account. Further investigation suggests that secondary interaction effects should be included in the response function to solve the problem in these cases.

1. Introduction

The majority of strategic transport model systems used today use classical static traffic assignment (STA) models. STA models assume separable monotonously increasing travel time functions, yielding computationally fast and scalable models with desirable convergence properties needed for strategic large scale transport model systems. In these systems, the same STA models are used to derive the relation between origin-destination (OD) travel demand and link flows (the so called assignment matrix) for estimation of the OD (travel demand) matrix. Matrix estimation methods using STA models have been studied extensively and are readily available, see e.g. Cascetta (2001) and references herein.

However, link flows and speeds from STA models do not correspond to empirically supported traffic flow theory that describes the relation between flow, speed and density in the form of a fundamental diagram. This is mainly caused by the lack of a true capacity constraint and a congested branch in travel time functions used in STA models. This becomes clear when the relation between flow and speed from a cost function from a STA model is compared to this relation in a fundamental diagram. The cost function and fundamental diagram behave similarly whenever the road segment is in uncongested state, where larger flows correspond to lower speeds (and higher densities). Critical differences however occur in congested state, where the cost function allows flows to exceed capacity, while in the fundamental diagram flows are monotonously decreasing when the density exceeds the critical density.

Therefore, STA models cannot cope with capacity constraints, nor represent the physical effects of congestion (flow metering and queue formation). This means that matrix estimation procedures using an assignment matrix from an STA model are not able to correctly incorporate flows observed on links in congested state, as these will be interpreted as uncongested flows by definition.

Macroscopic dynamic traffic assignment (DTA) models typically use a fundamental diagram and therefore incorporate capacity constraints and physical effects of congestion and are thus far more realistic compared to STA models. However, these models are poorly scalable, data intensive (i.e.: they need dynamic travel demand matrices), have convergence issues and suffer from a decreased stability and tractability, mainly because the (implicit) travel time functions in a fundamental diagram are non-separable across both space and time.

This paper focuses on matrix estimation for strategic transport model systems using a model that combines advantages of STA and DTA models: Static Traffic Assignment with Queuing (STAQ, Brederode et al. 2010, Bliemer et al. 2012). Following the unified framework for traffic assignment models described in Bliemer et al. (2014) STAQ is typified as a semi-dynamic model. It consists of a node- and a link- model and makes use of route fractions from a route choice model. STAQ accounts for flow metering and queue formation, but does not use a time dimension to propagate traffic through the network. Instead, all demand is assigned to the network in a single time period where a vehicle may either reach its destination or remain in a traffic queue. Queues start to grow from nodes where the (reduced) supply on downstream links is restrictive, while the node model distributes the available supply.

Similar semi-dynamic models (but without spillback) are described in Kohler and Strehler 2010, Smith 2012 and Bliemer et al. 2013. In Smith et al. 2013 a similar semi-dynamic model with spillback is described, however without a proper node model. Other semi dynamic models with spillback (e.g. Bifulco and Chrisalli 1998, Lam and Zhang 2000, Bundschuh 2006, 4Cast 2009) use link exit capacities, but this approach unrealistically locates queues inside the bottleneck links contrary to upstream of the bottleneck. In this paper STAQ is used, but findings may apply to any (semi-)dynamic model that uses a node model that accounts for supply constraints.

1.1. Contributions

To the best of our knowledge, this contribution is the first to propose a matrix estimation method using a semi-dynamic model. We show how matrix estimation for semi dynamic models is unique in that it can benefit from both low data requirements due to the absence of a time dimension (similar to STA models, but contrary to DTA models where multiple time slices are estimated) as well as the inclusion of traffic count observations in the congested regime (similar to DTA models, but contrary to STA models where queue formation is not modelled). Also, our proposed method exploits the properties of the STAQ model leading to the following methodological advantages:

- The assignment matrix (capturing the relation between link flows and OD-flows) is directly derived from the reduction factors on turn level, one of the variables in STAQ.
- The response function with respect to flow metering is numerically approximated by a marginal simulation of the node model, without the need to (iteratively) run the full simulation model.
- The upper and lower bounds of demand-change for which the first order approximation of the response function is valid, can be derived.

Furthermore, this paper is the first to make the relations between demand and supply existing in the directed capacity proportional node model explicit and shows that interaction effects between demand on different paths plays a major role in matrix estimation when using such a node model.

2. The matrix estimation problem

The (travel demand) matrix estimation problem is often formulated as a bi-level optimization problem where in the upper level differences between observed and modelled link flows and OD-demands are minimized, while in the lower level the traffic assignment problem is solved using a STA or DTA model. The matrix estimation problem using STA models is a Cournot-Nash game that is solved by alternatingly solving the lower and upper level problem, whereas the matrix estimation problem using DTA models is a Stackelberg game that can only be solved when the response function (i.e. the response of the link flows to changes in OD-demand) is incorporated into the upper level objective function.

Consider a general network $G=(N,A)$ where N denotes the set of nodes and A denotes the set of directed links. Let $R \subset N$ and $S \subset N$ be the set of origins and destinations respectively and

$RS = R \times S$ the set of all OD-pairs. Furthermore, let $\tilde{A} \subset A$ be the set of links for which flow has been observed (from now on 'observed links'). The matrix estimation problem can now be formulated as:

$$\mathbf{D}^* = \arg\min_{\mathbf{D}} F = \arg\min_{\mathbf{D}} \left[f_1(\mathbf{D}, \mathbf{D}_0) + f_2(\mathbf{y}(\mathbf{D}), \tilde{\mathbf{y}}) \right] \tag{1}$$

where F denotes the upper level objective function to be minimized, $\mathbf{D}^*$, $\mathbf{D}$ and $\mathbf{D}_0$ denote vectors containing posterior, current and prior (or observed) OD demand respectively for all OD pairs in RS, $\mathbf{y}(\mathbf{D})$ and $\tilde{\mathbf{y}}$ denote vectors of estimated and observed link flows in $\tilde{A}$ and f_1 and f_2 denote distance functions measuring the differences between observed and estimated demand and flows. Note that $\mathbf{D}_0$ is possibly only available for a subset of observed OD pairs $\widetilde{RS} \in RS$. In the upper level, Equation (1) is solved given some response function $\mathbf{y}(\mathbf{D})$ from the lower level. For methods that use separable cost functions (typically STA models),

$$\mathbf{y}(\mathbf{D}) = \mathbf{M}(\mathbf{D})\mathbf{D} \tag{2}$$

where, the assignment matrix $\mathbf{M}(\mathbf{D})$ is a matrix of size $\left|\tilde{A}\right| \times |RS|$ that follows from the assignment model, and is incorporated from the lower level into the upper level. For methods that use non-separable cost functions, such as DTA and semi dynamic assignment models,

$$\mathbf{y}(\mathbf{D}) = \left[\mathbf{M}(\mathbf{D}) + \frac{d\mathbf{M}(\mathbf{D})}{d\mathbf{D}}\bigg|_{\mathbf{D}} \right] \mathbf{D} \tag{3}$$

where also changes in the assignment matrix due to changes in the demand are accounted for (by the differential function). Note that in DTA models vectors $\mathbf{D}$ and $\mathbf{M}(\mathbf{D})$ are expanded by a time dimension T denoting the number of time intervals modelled.

2.1. Relation with the network loading and route choice model

In the matrix estimation problem, the assignment matrix expresses the interaction effects between supply and demand on the network, which originate on locations where demand exceeds the network supply. STAQ is a network loading model describing such interaction effects. Within the network loading model, nodes represent (possible) spatial discontinuities in travel demand (i.e. merge/diverge) and/or link capacities. These discontinuities can cause the formation of boundaries between traffic states in the form of shockwaves. The node model describes macroscopic behaviour of drivers confronted with such discontinuities on nodes. As such, the node model defines the locations where congestion is initially formed along with the congestion severity. Furthermore it defines how shockwaves are 'distributed' over ingoing and outgoing links (from now 'inlinks' and 'outlinks') whenever they encounter a node. These shockwaves are passed on to the link model in the form of turn based reduction factors, the link model propagates these reduction factors over links.

These turn based reduction factors can be translated into path based reduction factors by:

$$\hat{\alpha}_{ap} = \prod_{ij \in IJ_{ap}} \alpha_{ij} \tag{4}$$

where α_{ij} is a turn based reduction factor on a turn from inlink *i* to outlink *j* determined by the node model and IJ_{ap} is the set of turns used by path *p* when travelling from origin to link *a*. The path based reduction factors $\hat{\alpha}_{ap}$ describe the fraction of traffic on path *p* that is not held up by supply constraints upstream from link *a* and can be combined with route fractions from the route choice model to determine elements in the assignment matrix by:

$$m_a^{rs} = \sum_{p \in P_a} \psi_p^{rs} \hat{\alpha}_{ap} \tag{5}$$

where m_a^{rs} is the fraction of demand from OD pair *rs* that flows over link a (and represents one element in $\mathbf{M}$), ψ_p^{rs} is the fraction of demand from OD pair *rs* that chooses to use path *p* and (determined by the route choice model) and P_a is the set of paths using link *a*. Note that due to the FIFO assumption that exists in STAQ, reduction factors for all turns on an inlink of node are equal by definition, thereby also defining a relation between turn based and link based reduction factors.

In this paper we develop a method to solve the matrix estimation problem on the node level, thus taking into account interaction effects between demand on turns and supply on outlinks of a node, causing flow metering. Interaction effects on the level of links and paths are considered exogenous in the remainder of this paper. This means that route fractions and reduced supply of outlinks due to spillback from other supply constrained nodes downstream are assumed to be given. Then we can express OD demand on the path level using

$$D_p = D_{rs} \psi_p^{rs} \tag{6}$$

and the assignment entails merely a run of the STAQ propagation model translating path demands into link flows and speeds. Note that for notational convenience, in the remainder we will omit superscript *rs* from path variables (as a path implies the origin and destination), unless strictly necessary.

Note that route fractions and reduced supply are both explicit variables from the STAQ assignment model, and that the method described in this paper allows for future extensions to incorporate interaction effects caused by these phenomena, through the relations between reduction factors on the level of turns, paths, links and OD pairs described by equations (4) and (5).

3. Proposed method: lower level

In order to solve the bi-level problem, the upper and lower level is solved iteratively. In each iteration:

- in the lower level one STAQ assignment is run yielding the assignment matrix and corresponding link flows. Furthermore, several marginal runs of the node model within STAQ are performed, yielding the approximated sensitivity of the assignment matrix to changes in OD-demand without the need to (iteratively) run the full simulation model.
- in the upper level, the assignment matrix and its approximated sensitivity from the lower level are used to find the OD matrix that minimizes differences between modelled and observed link flows and differences between estimated and a prior OD matrix.

A general description of STAQ is already given in section 1, along with references to detailed descriptions. A general overview of the method used in the lower level is described in section 3.1. The method involves marginal simulation using only the node model within the assignment method as described in section 3.2. A description of the node model used in STAQ is given in section 3.3. Section 3.4 describes how this node model defines the relation between demand, flow and the assignment matrix on turn and link level using a numerical example. Based on insights from section 3.4, properties of the node model relevant for the matrix estimation problem are described in sections 3.5.

3.1. Approximation of response function

In line with the state of the art matrix estimation methods for DTA models Frederix (2012) derived the first order Taylor approximation of the response function as:

$$y_a^l = \sum_{t=1}^{l} \sum_{p \in P_a} \hat{\alpha}_{ap}^{tl}(\mathbf{D_0}) D_p^t + \sum_{t=1}^{l} \sum_{p' \in P} (D_{p'}^t - D_{0,p'}^t) \left[\sum_{t=1}^{l} \sum_{p \in P_a} \left. \frac{d\hat{\alpha}_{ap}^{tl}(\mathbf{D})}{dD_{p'}^t} \right|_{\mathbf{D_0}} D_{0,p}^t \right] \tag{7}$$

where y_a^l denotes the inflow of link a during time period *l*, $D_{0,p}^t$ and D_p^t are the prior and posterior demand for path *p* departing in time period *t* respectively, $\hat{\alpha}_{ap}^{tl}$ (one element in the assignment matrix

given fixed route fractions) is the fraction of $D^t_{0,p}$ that enters link a during time period *l*, and P is the set of all paths. The second term in (7) describes the first order effects.

In STAQ, the time dimension is absent and response function (7) simplifies to:

$$y_a = \sum_{p \in P_a} \hat{\alpha}_{ap}(\mathbf{D_0}) D_p + \sum_{p' \in P} (D_{p'} - D_{0,p'}) \left[\sum_{p \in P_a} \left. \frac{d\hat{\alpha}_{ap}(\mathbf{D})}{dD_{p'}} \right|_{\mathbf{D_0}} D_{0,p} \right] \tag{8}$$

that is separable across time periods, but inseparable across all paths. This means that the derivatives in the second term usually are approximated through complete runs of the assignment model, as done in finite difference methods and SPSA (Spall 1998). This entails both very large calculation times and tedious tuning of algorithmic parameters (Cipriani et al. 2012).

Using STAQ, such methods can be avoided by use of the turn based reduction factors α_{ij} that are endogenous variables of STAQ. These reduction factors represent the ratio between the demand and realised flow on a turn *ij* and are calculated by the node model. Derivatives of the reduction factors on turn level to demand on path level ($d\alpha_{ij} / dD_p$) can be derived as follows. First, $d\alpha_{ij} / dD_p$ is approximated using the node model. This approximation only requires several runs of the node model that comes at negligible computational cost compared to full simulation runs as required in other methods. Then, the derivative of an element in the assignment matrix (given fixed route fractions) can be calculated using the product rule:

$$\frac{d\hat{\alpha}_{ap}}{dD_p} = \left(\prod_{ij \in IJ_{ap}} \alpha_{ij} \right) \left(\sum_{ij \in IJ_{ap}} \frac{d\alpha_{ij} / dD_p}{\alpha_{ij}} \right) \tag{9}$$

Note that changes in the assignment matrix as a result of changes in demand may itself result in additional changes in demand (which we will call secondary interaction effects). By using partial derivatives only to each single OD pair we assume that such secondary interaction effects are negligible. Also, because of the marginal simulation on node level, the proposed method cannot be used to calculate derivatives to demand on paths that do not use link *a*.

3.2. Marginal simulation: the node model

Below, the notation that will be used to describe the node model is presented. Consider a node n connected to a set of inlinks I_n and a set of outlinks J_n forming the set of turn movements using the node $IJ_n = I_n \times J_n$. Furthermore, we define the set of outlinks directly related to inlink *i* as $J_i = \{ j \mid D_{ij} > 0 \}$ and the set of inlinks directly related to outlink *j* by $I_j = \{ i \mid D_{ij} > 0 \}$.

For all $n \in N \subset G$ a node model $\Gamma_n(\cdot)$ is defined that calculates the vector of turn-flows $\mathbf{y}_n$ over n as a function of the vector of travel demand for each turning movement on the node ($\mathbf{D_n}$), the vector of link capacities of inlinks ($\mathbf{C}_n$) and the vector of supply constraints on the outlinks of the node ($\mathbf{R}_n$) defined by link geometry or spillback from downstream supply constraints. This yields:

$$\begin{aligned} &\mathbf{y}_n = \Gamma_n(\mathbf{D}_n, \mathbf{C}_n, \mathbf{R}_n) \\ &\text{where: } \mathbf{y}_n = \{ y_{ij} \quad \forall ij \in IJ_n \}, \\ &\qquad \mathbf{D}_n = \{ D_{ij} \ \forall ij \in IJ_n \}, \\ &\qquad \mathbf{C}_n = \{ C_i \quad \forall i \in I_n \} \text{ and} \\ &\qquad \mathbf{R}_n = \{ R_j \quad \forall j \in J_n \} \end{aligned} \tag{10}$$

The (reduced) demand on turn level for turns over the considered node is calculated by summing all (reduced) demand of paths using that turn:

$$D_{ij} = \sum_{p \in P_{ij}} \hat{\alpha}_{ij} D_p \tag{11}$$

where D_p is calculated by the route choice model using (6) and $\hat{\alpha}_{ij}$ is calculated using (4) as part of the propagation model combining results from upstream node models. Once solved, turn based reduction factors can be derived by:

$$\alpha_{ij} = y_{ij} / D_{ij} \ . \tag{12}$$

Using the node model a point derivative of α_{ij} to any D_{ij} can be approximated by running the model twice around the current value of D_{ij} . These point derivatives are then used as an approximate of $d\mathbf{A}(\mathbf{D})/d\mathbf{D}$ in the upper level. It is important to note here, that by approximating derivatives we determine all the partial derivatives (forming the Jacobian), but we choose to omit approximating secondary interaction effects, since we assumed that these effects are negligible (section 3.1). This means that we omit the fact that when simultaneously changing multiple elements in **D**, the effect on **A** might not be simply the sum of the effects of changing D sequentially per OD pair.

Further note that supply constraint values in $\mathbf{R}_n$ are equal to or lower than the link capacity of the outlink, depending on the state of the outlink defined by the link model. For the sake of simplicity, in this paper, constraints imposed by geometry of the node itself (the so called internal node constraints) are assumed to be non-existent.

3.3. The node model used in STAQ

The node model used in STAQ is adopted from Tampère et al. (2011) who describe a set of requirements for realistic first order macroscopic node models that yield consistent solutions, along with the specification of a node model that complies to these requirements (which is adopted in STAQ). One of the requirements is that the node model should contain supply constraints limiting the amount of traffic that can flow into an outlink by the capacity or reduced supply of that outlink. If supply constraints are active, the limited supply of an outlink must be distributed over the different turning movements towards this link according to so called supply constraint interaction rules (SCIR).

Smits et al. (2014) define a generic class of first order node models based on the requirements by Tampère et al. (2011) and point out that adding a specific set of SCIR leads to a specific node model. They identify SCIR for different node models found in literature and point out that the node model described by Tampère et al. (2011) is equivalent to the model described in Flötteröd and Rohde (2011) and uses directed capacity proportional distribution as SCIR. This means that whenever turning movements from multiple inlinks are competing for supply of one outlink, the available supply is distributed proportional to the directed capacity of the competing turn movements defined as:

$$C_{ij} = \frac{D_{ij}}{\sum_{j \in J_i} D_{ij}} C_i \tag{13}$$

As Smits et al. (2014) point out, due to the SCIR in this node model, each turning movement can only be affected by one constraint. Therefore the proportionality only holds for inlinks that are not affected by another (supply or demand) constraint.

Following one of the other requirements (the conservation of turning fractions) the node model assumes FIFO which means that the SCIR implicitly also determines the indirect effects of supply constraints on turning movements sharing an inlink with turning movements affected by the supply constraint. This means that, because of FIFO, α_{ij} is equal for all turning movements that share the same inlink, thus $\alpha_{ij} = \alpha_i$ and we can define $\boldsymbol{\alpha}_n = \{\alpha_i \ \forall i \in I_n\}$.

The solution algorithm for the directed capacity proportional node model that is proven to converge to the unique solution of directional supply constrained SCIR can be found in Tampère et al. (2011), Flötteröd and Rohde (2011) and Smits et al. (2014). For convenience of the reader, the algorithm using notation from this paper can be found in appendix 1. The solution algorithm shows

that it is needed to sequentially handle each outlink, because available supply in an iteration $\tilde{R}_j^k$ and the sets of turns competing for this supply I_j^k are dependent on demand constraints (by lines (10) and (12) of the algorithm) or more restrictive supply constraints (by lines (21) and (24)) handled in previous iterations *k*.

Within one iteration of the algorithm, turn flows are determined for turns on one or more inlinks constrained by the most restrictive outlink in that iteration $\hat{j}$ by:

$$y_{ij} = \beta_{\hat{j}} C_{ij} = \frac{\tilde{R}_{\hat{j}}^k C_{ij}}{\sum_{i \in I_{\hat{j}}^k} C_{ij}} \qquad \forall i \in I_{\hat{j}}^k. \tag{14}$$

Then, reduction factors for these inlinks can calculated using:

$$\alpha_i = \frac{y_{ij}}{D_{ij}} \qquad \forall i \in I_{\hat{j}}^k. \tag{15}$$

3.4. Approximating derivatives using the node model

From section 3.3 we conclude that the supply constraints of the node model, together with the SCIR actually define the relationship between $\mathbf{y}_n$ and $\mathbf{D}_n$ and ultimately $\boldsymbol{\alpha}_n$ and $\mathbf{D}_n$. For the node model used in STAQ, these relations are implicit, but can be made explicit within one iteration of the solution algorithm as shown in equations (14) and (15). In order to be able to use the node model to numerically approximate $d\alpha_{ij} / dD_p$, some properties of these implicit relationships are of importance. To get insight into the implicit relations, we use the numerical example presented in section 2.1.4 of Tampère et al. (2011) as a starting point. This example is summarized in Fig. 1, which displays both input and output of the node model. For all inlinks (marked O1-O4 in grey) input consists of link capacities (displayed in italics) and demand for each turn ('turn demands', displayed in normal font). For all outlinks (marked D1-D4 in grey), input consists solely of link capacities (displayed in italics). Output consists of flows per turn from each inlink (displayed bold-green (when demand constrained) or bold-red (when capacity constrained)) and total flows per outlink (displayed in bold-black).

In line with the solution algorithm, we 'explain' the numerical example by handling all outlinks sequentially starting with the most restrictive, distributing remaining supply over competing inlinks in each iteration. Comparing total turn demand towards each outlink with the capacity of the respective outlink shows that only outlink D3 is capacity constrained and as such is the most restrictive outlink with competing turns O1D3, O2D3 and O4D3. When distributing the supply of D3 proportional to the directional capacities of its competing turns, demand on turn O1D3 turns out to be less than its rightful share. Therefore turn O1D3 is demand constrained and all turn demand from inlink O1 can be accommodated ($\alpha_1 = 1$). The remaining capacity on outlink D3 (850) is distributed over O1D3 and O4D3 proportional to the directional capacities of these turns. This yields $\alpha_4 < 1$ and $\alpha_2 < 1$. Inlink O3 is demand constrained, therefore $\alpha_3 = 1$.

To demonstrate the mechanisms resulting from the directional capacity constrained distribution as SCIR we now vary the demand on turn O1D4 from 0 to 800 (its maximum possible value, given the inlink capacity and demand of other turns on the inlink). Resulting values of $\boldsymbol{\alpha}_n$ are displayed in Fig. 2. From this figure, we see that $d\alpha_i / dD_{14} = 0$ whenever $D_{14} \leq 304$. The reason for this is that for this range of D_{14} inlink 1 stays demand constrained and demand towards the only constraining outlink (D3) is not influenced by D_{14}. For $305 \leq D_{14} \leq 390$ inlink 3 becomes capacity constrained by outlink 4, whereas inlink 1 stays demand constrained and inlinks 2 and 4 stay supply constrained by outlink 3 (which is still the most constraining outlink overall). This means that any increase of D_{14} in this range does not reduce the available supply available for inlink 1 (since it is demand constrained) whereas the turn flow from inlink 2 remains constant (since it is supply constrained by the more restrictive outlink 3). Therefore, all extra demand from inlink 1 is directly translated to turn flows, reducing the available supply for inlink 3 ($d\tilde{R}_4^k = -dD_{14}$). In terms of derivatives of equations (14) and (15), this means that for this range of D_{14}:

$$\frac{dy_{34}}{dD_{14}} = \frac{dy_{34}}{-d\tilde{R}_4^k} = -\frac{C_{34}}{\sum_{i \in I_4^k} C_{i4}} \quad \text{and} \quad \frac{d\alpha_3}{dD_{14}} = -\frac{C_{34}}{D_{34}\sum_{i \in I_4^k} C_{i4}} \approx -0.00167$$

which is the slope of the linear decrease of the green line between 305 and 390 in Fig. 2.

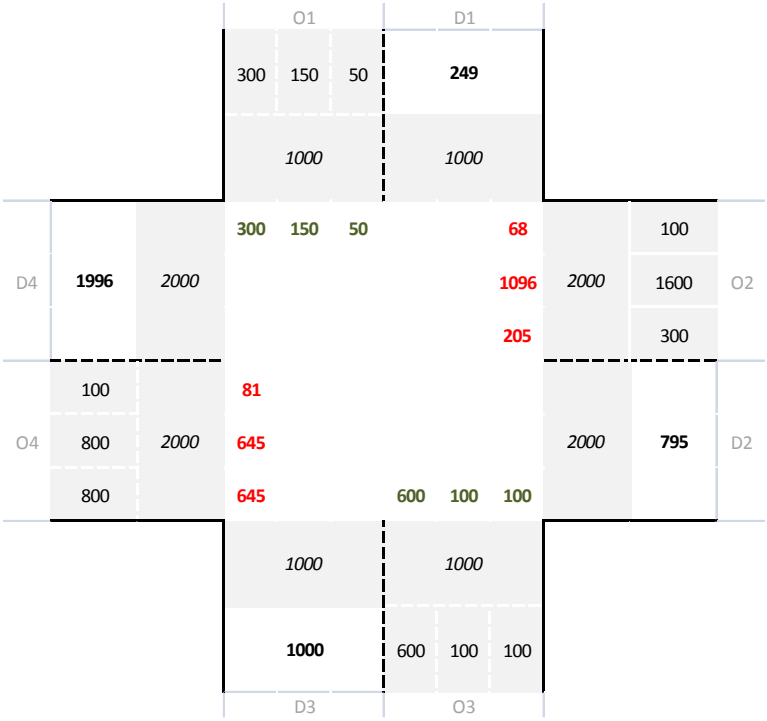

Fig. 1 numerical example of node model used in section 2.1.4. of Tampère et al.

For $391 \leq D_{14} \leq 456$ outlink 4 becomes the most constraining outlink, constraining both inlinks 2 and 3. Outlink 3 remains constraining inlink 4, whereas inlink 1 remains demand constrained. This means that for outlink 4, any increase of D_{14} in this range does not reduce the available supply for inlink 1 (since it is demand constrained), but does reduce available supply for inlinks 2 and 3.

Similar to the previous situation, all extra demand from inlink 1 is directly translated to turnflows, reducing the available supply for inlinks 2 and 3 ($d\tilde{R}_4^k = -dD_{14}$). In terms of derivatives of equations (14) and (15), this means that for this range of D_{14} :

$$\frac{dy_{24}}{dD_{14}} = \frac{dy_{24}}{-d\tilde{R}_4^k} = -\frac{C_{24}}{\sum_{i \in I_4^k} C_{i4}} \quad \text{and} \quad \frac{d\alpha_2}{dD_{14}} = -\frac{C_{24}}{D_{24}\sum_{i \in I_4^k} C_{i4}} \approx -4.25E-04\,;$$

$$\frac{dy_{34}}{dD_{14}} = \frac{dy_{34}}{-d\tilde{R}_4^k} = -\frac{C_{34}}{\sum_{i \in I_4^k} C_{i4}} \quad \text{and} \quad \frac{d\alpha_3}{dD_{14}} = -\frac{C_{34}}{D_{34}\sum_{i \in I_4^k} C_{i4}} \approx -5.32E-04\,,$$

which correspond to the slopes of the linear decrease of the green and red lines between 391 and 456 in Fig. 2.

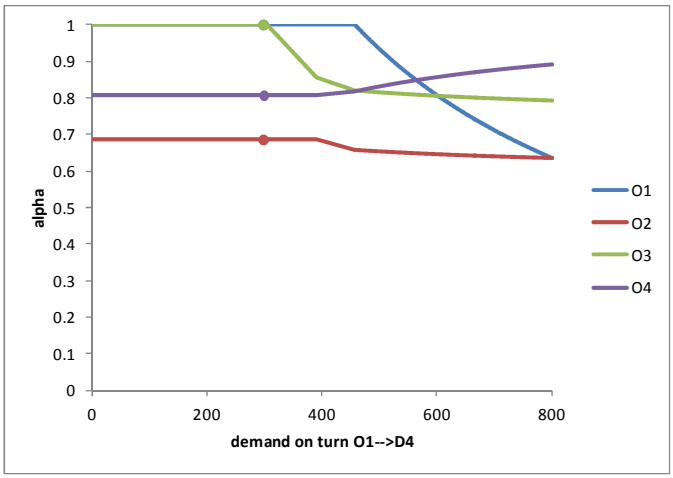

Fig. 2: reduction factors α_n when varying D_{14}

The reduction factor of inlink 4 increases linearly due to the decreased competition of demand from inlink 2 (since it is now constrained by outlink 4). To calculate dy_{43}/dD_{14} we translate y_{43} in terms of D_{14} :

$$y_{43} = \frac{\tilde{R}_3^3 C_{43}}{\sum_{i \in I_3^3} C_{i3}}, \text{where}$$

$$R_3^3 = \tilde{R}_3^1 - y_{13} - y_{23} = \tilde{R}_3^1 - D_{13} - \beta_4 C_{23} \qquad (16)$$

$$= \tilde{R}_3^1 - D_{13} - \frac{\tilde{R}_4^2 C_{23}}{\sum_{i \in I_4^2} C_{i4}} = \tilde{R}_3^1 - D_{13} - \frac{(\tilde{R}_4^1 - D_{14}) C_{23}}{\sum_{i \in I_4^2} C_{i4}}$$

which is recursive calculation of (14) over iterations, taking lines (10), (12), (21) and (24) of the algorithm described in appendix 1 into account. Using this translated y_{43} we can calculate dy_{43}/dD_{14} as:

$$\frac{dy_{43}}{dD_{14}} = d\frac{\tilde{R}_3^3 C_{43}}{\sum_{i \in I_3^3} C_{i3}} / dD_{14} = d\frac{\left(\tilde{R}_3^1 - D_{13} - (\tilde{R}_4^1 - D_{14}) C_{23} / \sum_{i \in I_4^2} C_{i4}\right) C_{43}}{\sum_{i \in I_3^3} C_{i3}} / dD_{14}$$

$$= \frac{C_{23} C_{43}}{\sum_{i \in I_4^2} C_{i4} * \sum_{i \in I_3^3} C_{i3}} \quad \text{and} \quad \frac{d\alpha_{43}}{dD_{14}} = \frac{dy_{43}}{D_{43} dD_{14}} \approx 1.5957E-04$$

which correspond to the slopes of the linear increase of the purple line between 391 and 456 in Fig. 2.

For $457 \le D_{14} \le 800$ inlink 1 also becomes capacity constrained by most constraining outlink 4, whereas inlink 4 stays supply constrained by outlink 3. There are no demand constrained inlinks anymore. This means that for outlink 4, any increase of D_{14} only changes the distribution of the supply constraint between inlinks 1, 2 and 3 from inlinks 2 and 3 to inlink 1. Considering inlink 3:

$$y_{34} = \frac{\tilde{R}_4^1 C_{34}}{\sum_{i \in I_4^1} C_{i4}},$$

taking the derivative to D_{14} (see appendix 2 equation a1 for derivation) yields:

$$\frac{dy_{34}}{dD_{14}} = \frac{-\tilde{R}_4^1 C_{34} * C *_1 (D_{12} + D_{13})}{\left[D_{14}(C_1 + C_{24} + C_{34}) + (D_{12} + D_{13})(C_{24} + C_{34})\right]^2}$$

which, when divided by D_{34}, describes the slope of the green line between 457 and 800 in Fig. 2. A similar derivation (see appendix 2 equations a2 and a3) holds for inlinks 1 and 2, yielding derivatives of the same form:

$$\frac{dy_{14}}{dD_{14}} = \frac{\tilde{R}_4^1 C_1 (D_{12} + D_{13})(C_{24} + C_{34})}{\left[D_{14} C_1 + (D_{12} + D_{13} + D_{14})(C_{24} + C_{34})\right]^2}$$

$$\frac{dy_{24}}{dD_{14}} = \frac{\left(-\tilde{R}_4^1 C_{24}\right) * C_1 * (D_{12} + D_{13})}{\left[D_{14}\left(C_1 + C_{24} + C_{34}\right) + (D_{12} + D_{13})\left(C_{24} + C_{34}\right)\right]^2}$$

The derivatives are nonlinear function $f : dD_{ij} \rightarrow dy_{i'j'}$ in the form of $c_1/(c_2 D_{14} + c_3)^2$, where c_1, c_2 and c_3 are constants composed of the supply of the considered outlink, directional capacities of other turns towards this outlink, the turn demands on turns sharing the inlink with D_{14} and the capacity of that inlink.

The only remaining inlink 4 is constrained by outlink 3, but this is not the most restrictive constraint. Therefore, some of the capacity of outlink 3 is already used by turns from inlinks 1, 2 and 3 before the constraint of outlink 3 becomes active. In terms of equation (14), this means that, due to the effect of turns restricted by outlink 4 its enumerator changes nonlinearly in a positive sense ($d\tilde{R}_j^k / dD_{14} > 0$), since competition by turns O1D3 and O2D3 decreases due to the increased share of turn O1D4, leaving more room on outlink D3 for turn O4D3. The denominator in equation (14) is a constant, since turn O4D3 is the only turn 'competing' for outlink D3 (i.e.: the denominator only contains the directed capacity of O4D3). The derivative now becomes (see appendix 2 equation a4):

$$\frac{dy_{43}}{dD_{14}} = \frac{\tilde{R}_4^1 D_{13} C_2 C_{23} \left(C_1 + C_{24} + C_{34}\right)}{\left[C_1 D_{14} + (D_{12} + D_{13} + D_{14}) C_{24} + (D_{12} + D_{13} + D_{14}) C_{34}\right]^2}$$

which, when divided by D_{43} ,describes the slope of the purple line between 457 and 800 in Fig. 2. Note that this function is also in the form $c_1/(c_2 D_{14} + c_3)^2$, where the constants are now also composed of the capacity of inlink O2 and the directed capacity of O2D3, besides the constants that where already included in the derivatives of turn flows towards the most restrictive outlink 4.

Based on the solution algorithm and the example described in this section, we conclude that:

- Function $\alpha_i(D_{i'j'})$ exists on the domain $\left\langle 0, C_i - \sum_{j' \in J_i / \{j\}} D_{ij'} \right]$
- Function $\alpha_i(D_{i'j'})$ is continuous on its positive domain, can be constructed piece-wise, and is differentiable almost everywhere. At each interval of $D_{i'j'}$, α_i is determined by the same constraint, and at each non-differentiable point, a switch between active constraints occurs.
- Function $\alpha_i(D_{i'j'})$ is either monotonously increasing or decreasing on its positive domain, depending on the effect that $D_{i'j'}$ has on $C_{i\hat{j}}$ in relation to $\sum_{i \in I_j} C_{i\hat{j}}$. An increasing function can only occur when $i \neq i'$.
- On an interval where turn *ij* is demand constrained, $dy_{ij} / dD_{i'j'} = 0$ and $d\alpha_i / dD_{i'j'} = 0 \; \forall i, j, i', j' \in IJ$
- On an interval where turn *ij* is supply constrained by an outlink to which at least one demand constrained turn exist, function $\alpha_i(D_{i'j'})$ has a linear form.
- Whenever turn *ij* is supply constrained by an outlink to which no demand constrained turns exist, $\alpha_i(D_{i'j'})$ has the form of $c_1/(c_2 D_{i'j'} + c_3)^2$.

These properties of $\alpha_i(D_{i'j'})$ are of importance for the matrix estimation problem, since they can be exploited by the optimization method used in the upper level.

3.5. Interdependencies of turn flows in the node model

The example described in section 3.4 demonstrates that directed capacity proportional SCIR combined with the FIFO assumption introduces interdependencies between demand on the different

turn movements on a node when there are supply constrained outlinks. More specific the flow of each supply constrained turning movement *ij* on node *n* is dependent on:

- The demand on turns that share the inlink with the considered turn (due to the FIFO assumption);
- The demand on turns competing for supply of the most restrictive outlink (due to the SCIR);
- The demand on turns sharing an inlink with turns competing for the most restrictive outlink. (due to the FIFO assumption); and
- The demand on demand constrained turns towards the most restrictive outlink (due to the SCIR).

Assuming a node with four arms and banned u-turns, twelve different turn movements exist. In the worst case there are three inlinks constrained by the most restrictive outlink. In that case, for all turns from these restricted inlinks, the flow is dependent on demand on all eleven other turns: two turns that share the same inlink, three turns that compete for supply of the most restrictive outlink and six turns that share an inlink with turns competing for supply of the most restrictive outlink. This means that we would need to include $\partial \alpha_i / \partial D_{i'j'} D_{i''j''}$ for all combinations of turns into the response function to describe the (first order) interaction effects, which means that we add another dimension to the derivative of the assignment matrix making the matrix estimation problem harder to solve for the upper level.

4. Proposed method: upper level and bi-level problem as a whole

In this section the methods used solving the upper level problem and the bi-level problem as a whole are described.

4.1. Solving the upper level

The choice for a method for solving the upper level is influenced by the properties of the distance functions f_1 and f_2, which in turn are chosen dependant on properties of the variables in the objective function (observed link flows $\tilde{\mathbf{y}}$ and OD demand $\mathbf{D}_0$). Note that these are aggregate variables observed over some period(s) of time. Therefore, the observed values in vector $\tilde{\mathbf{y}}$ are in fact instances of some probability distribution. This is also the case for observed OD demand, since this is also an aggregated value which, on top of that, is only measured indirectly through surveys or derived from some distribution model. Although, when known, these distributions can be taken into account when solving the upper level, this is not subject of this paper. In the remainder we therefore choose the mean squared error (MSE) as distance function for both components, since it does not use any additional data on the distribution of the observed flow values or prior matrix. Furthermore, we introduce an extra parameter that allows for weighing of the two components in the objective function. Using MSE and weighing parameter w, the objective function to be minimized in equation (1) now reads:

$$\min F = \min_{\mathbf{D}} \left[w \sum_{rs \in \bar{RS}} (D_{rs} - D_{0,rs})^2 + (1-w)\theta \sum_{a \in \tilde{A}} (y_a(\mathbf{D}) - \tilde{y}_a)^2 \right] \tag{17}$$

where θ is a normalisation parameter the normalises the scale of the second component relative to the first component. The method used for estimation of θ will be described in paragraph 4.2.

Furthermore, equation (1) is subject to the following constraints:

$$\sum_{p \in P_a} \hat{\alpha}_{ap} D_p \leq C_a \qquad \forall a \in A \text{ ,for outlinks in free flow state} \tag{18a}$$

$$\sum_{p \in P_a} \hat{\alpha}_{ap} D_p > C_a \qquad \forall a \in A \text{ , for outlinks in congested state} \tag{18b}$$

$$D_p \geq 0 \qquad \forall p \in P \text{ .} \tag{19}$$

Constraint (18) ensures that link capacities for all links in free flow state are not exceeded and that demand for outlinks in congested state is greater than its capacity, whereas (19) is a non-negativity

constraint on path demand. In addition to these natural constraints, lower and upper bounds to the trip production per origin could be added:

$$\underline{D_r} \le \sum_{s \in S} D_{rs} \le \overline{D_r} \qquad \forall r \in R\,, \tag{20}$$

The lower and upper bounds in constraint (20) are usually related to prior and/or observed trip productions allowing for a specified maximum (absolute or relative) deviation. The optimization problem defined by (17), (18a), (19) and (20) is a quadratic optimization problem with linear constraints. To solve the problem, the generalized reduced gradient method (GRG2, Lasdon et al. 1975) is used.

4.2. Weighting of objective function components

Parameter w is used to define the relative importance of the two components f_1 and f_2 in objective function F. Typically it is set based on the level of confidence associated with the two types of observed data (prior matrix and count values). However, since these two types of data have a different scale (summation of link flows over number of observed links versus summation of OD demand over number of OD pairs) they must be normalized to allow the weighting parameter to be given a meaningful interpretation expressing the relative importance on a scale of zero to one.

One of the most common ways to normalise f_1 and f_2 (see e.g. Alpcan 2013), is by calculating the range between the optimal (so called *Utopia*) and pseudo-worst (so called *Nadir*) points in objective space for each component of the objective function. Using these points, the scale of each component relative to the other can be calculated and used for normalisation within the weight variable.

The objective function value of the Utopia point of the first component of F (denoted as f_1^U) is zero, which is the case when $\mathbf{D}^* = \mathbf{D}_0$. The objective function value of the Utopia point of the second component of F (denoted as f_2) is also zero which is the case when $y_a(\mathbf{D}) - \tilde{y}_a = 0 \;\forall a \in \tilde{A}$; which can only be the case when observed flows are consistent with link capacities (thus link capacity constraint (18a) does not prohibit count values to be reached). Since inconsistent observations should be removed prior to matrix estimation, in the remainder it is assumed that this condition is satisfied.

The objective function value of Nadir points f_1^N and f_2^N can be calculated by solving

$$f_1^N = \max(f_1) = \max_{\mathbf{D}} \left[\sum_{rs \in RS} (D_{rs} - D_{0,rs})^2 \right] \text{ and} \tag{21}$$

$$f_2^N = \max(f_2) = \max_{\mathbf{D}} \left[\sum_{a \in \tilde{A}} (y_a(\mathbf{D}) - \tilde{y}_a)^2 \right] \tag{22}$$

separately, both subject to constraints (18a), (19) and (20). Note that whereas (21) can be solved directly, solving (22) would require an iterative solution algorithm involving running the lower level several times. Given the sole purpose of normalisation, this would take too much calculation time. Therefore an approximation $f_2^{N'}$ is used instead, omitting constraint (20) and neglecting the interdependencies of link flows through the assignment in the lower level. In that case in the Nadir point, each observed link either operates at capacity or does not accommodate demand at all, simplifying (22) to:

$$f_2^{N'} = \sum_{a \in \tilde{A}} \max\left[(C_a - \tilde{y}_a)^2, \tilde{y}_a^2 \right]. \tag{23}$$

Then, the scale of f_2 relative to f_1 can be calculated by

$$\theta = \frac{f_1^N - f_1^U}{f_2^{N'} - f_2^U} = \frac{f_1^N}{f_2^{N'}} \tag{24}$$

which is used in (17). In the remainder of this paper, constraint (20) is not used, allowing for approximation of the Nadir point using (23). The effect of constraints based on trip production is left for future research.

4.3. Convergence and consistency between lower and upper level

Sections 3 through 4.3 discussed the methods used for solving the upper and lower level. In order to solve the whole bi-level problem, solutions of the two levels need to be consistent. This means that the estimated demand in the upper level should be stable over iterations, and the approximated link flows used in the upper level should be replicated by the 'true' assignment in the consecutive lower level.

Because the derivative in the second component of the response function (8) is an approximation, the upper level optimization can 'overshoot' the optimum and may never find the optimum. There are three causes for this overshooting. Firstly, by equation (9) $d\alpha_i / dD_p$ is based on point approximations of $d\alpha_i / dD_{i'j'}$ whereas this function consists of piece wise differentiable intervals (section 3.4). This means that the point approximations are only valid within their respective interval. Secondly, the approximation of $\alpha_i(D_{i'j'})$ is linear, whereas the true function is only linear when at least one of the turns toward the restricting outlink is demand constrained (section 3.4). Thirdly, we omit to include secondary interaction effects in the response function, whereas from section 3.5 we know that potentially all turns on the node model may interact.

We propose to improve the first order approximation by reducing approximation errors due to the piece wise and possibly non-linear form of $\alpha_i(D_{i'j'})$ by constraining the change $\Delta\mathbf{D}$ that can be made to $\mathbf{D}$ within one (upper level) iteration. We relate $\Delta\mathbf{D}$ to the error of the linear approximation of $\alpha_i(D_{i'j'})$ for each OD pair in $\mathbf{D}$ by setting some upper bound ε_α on the tolerated approximation error. The method is illustrated by Fig. 3 that displays $\alpha_4(D_{43})$ in the example from section 3.4.

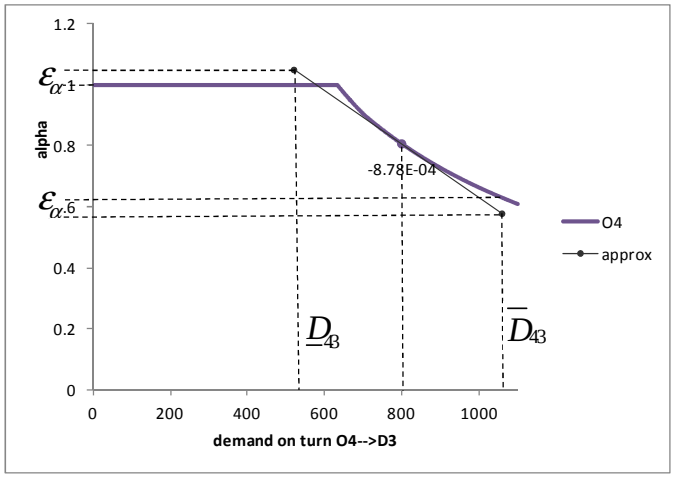

Fig. 3: reduction factor α_4 when varying D_{43}

In this example, ε_α was set to 0.05. We use two binary searches starting from $D_{0,43}$ that compare the difference between approximated and true $\alpha_4(D_{43})$, one in downward and one in upward direction yielding lower bound $\underline{D}_{43}$ and upper bound $\overline{D}_{43}$ respectively. Since we know that $\alpha_4(D_{43})$ is monotonously decreasing, the binary searches are guaranteed to find the lower and upper bounds. Note that the binary searches need to recalculate the node model at each candidate point, at the cost of computation time. The example demonstrates that this method detects both approximation errors due to non-linearity of the function (at the upper bound) as well as approximation errors due to piece wise form of the function (at the lower bound). This method is implemented in a prototype and tested in section 5.

Dependent on the extent to which this improvement solves the problem of overshooting, two further improvements may be researched in the future. Firstly, the point approximation of $d\alpha_i / dD_{i'j'}$ could be replaced by using the actual function that is valid for the respective interval $D_{i'j'}$ (for which the first order derivatives are already derived in 3.4). For this method, the non differential points of $\alpha_i(D_{i'j'})$ (the boundaries of the intervals) must be known. These can be found using a binary search in a similar way as described above. When applied for each interval, this improvement would rule out any approximation errors due to the piece wise and potentially non-linear form of $\alpha_i(D_{i'j'})$. To also overcome the last cause of overshooting, secondary interaction effects could be described by adding interaction terms to the approximations. In theory these two future improvements can fully accommodate for all three causes of overshooting. Note however, that it might not be possible to analytically

derive the secondary interaction effects as a function, and that the number of partial derivatives to calculate (and include in the upper level) increases from the number of turning movements (on node level) or paths (on network level) to the square of these, which probably limits scalability of the method. Further research is needed to develop methods described in this section. Therefore, in section 5.2 we test to what extent the first solution solves the overshooting problem, and based on the outcome discuss the prospects of pursuing the other improvents.

4.4. Insensitivity of link flows due to supply constrained turns

As described in section 3.4, each outlink can either be demand or supply constrained and flow into a constrained outlink will remain equal to the supply constraint of that outlink as long as the supply constraint is active:

$$\text{if } \sum_{i \in I_n} \alpha_i D_{ij} = R_j \text{ then } \frac{dy_j}{dD_{ij}} = 0 \quad \forall ij \in \{IJ_n \mid j \in J_n\}. \tag{25}$$

This property is a direct result of the existence of supply constraints in the node model and is responsible for the metering of traffic flow as a result of bottlenecks or spillback of traffic from other bottlenecks.

For matrix estimation, a problem arises when the considered outlink *j* is an observed link (i.e. $j \in \tilde{A}$). In this case the upper level cannot alter the flow on this link. Whenever this is the case, prior demand on paths using link j and observed flow on link j are inconsistent. Either the prior demand is too high (the supply constraint for *j* is not active in reality) or the observed flow on the considered link is too low (the supply constraint for j is active in reality). Depending on which information is thought of as the most reliable, either the considered link should be removed from $\tilde{A}$or the prior demand on paths using the considered link should be adjusted.

In case the observed link flow is considered more reliable, the prior demand level of one or more turns needs to be changed to a value that lies within a demand range where (25) does not hold for one or more turns towards link a. To find the closest (upper or lower) bound of this range of sensitive demand, we propose to use a binary search on each turn in IJ_n, starting from its prior demand level in downwards direction. Note that although when (25) holds, $y_j(D_{ij})$ is insensitive, but the function $\alpha_i(D_{i'j'})$ can still be sensitive change reflecting a change in the (directed capacity proportional) distribution of the supply of the outlink when D_{ij} changes. Therefore in the binary search the variable $\sum_{i \in I_j} dy_{ij} / dD_{i'j'}$ is evaluated, where a value not equal to 0 indicates that a sensitive range has been found.

Then, the turn for which the prior demand is closest to its sensitive range is selected and the prior demand value for this turn is 'set' to a value just within the sensitive range. Furthermore, we add this value as an upper bound to the demand of the adjusted turn in the upper level. By adjusting the prior demand of the turn with original prior demand closest to its sensitive range, deviation of the original prior demand matrix is minimized. Also, since we are looking for the closest sensitive range, we can stay away from using more costly searching techniques for global optima to find the sensitive range.

Note that the two different states of a turn or outlink on a node are closely related to the two different states defined in the fundamental diagram of a link. In fact, the change of a demand constrained to a supply constrained turn causes the inlink of the turn to change from a free flow to a congested state and vice versa. In that sense the adjustment of the demand using the method described above can be considered as equivalent to the adjustment of the prior matrix to make sure the prior assignment results in the correct regime for each observed link as described by Frederix (2012). This dissertation also describes the necessity to stay in the correct regime, adding the upper bound on the turn demand can be seen as a method to ensure that this condition is maintained.

5. Application of proposed method on node level

In this section we add observed link flows on two of the outlinks of the example from section 3.4 and solve the matrix estimation problem to demonstrate the matrix estimation method described in sections 3 and 4 using a prototype implementation of the methods described in sections 3 and 4. Because we consider a network consisting of only one node, $I = I_n$ and $J = J_n$ meaning that all variables on turn level are equivalent to the variables on path level, and application of equation (4) can be omitted.

In section 5.1 we solve the problem for the situation where only the count values are considered by setting *w*=0 in (17)[1]. In section 5.2 we increase *w* to demonstrate the normalisation described in 4.2 and to force the upper level to make a trade off between differences in observed and modelled link flows and differences between estimated and prior matrix. This incurs more simultaneous changes to ODpairs and thus interaction effects, which is a good case to test the method constraining $\Delta\mathbf{D}$ per iteration as described in section 4.3.

5.1. Example optimizing only on observed values

In this section we solve the problem for the situation where observed values $\tilde{y}_1 = 498$ and $\tilde{y}_4 = 1590$ are added to the example from section 3.4 and only count values are considered by setting *w*=0 in (17). Using the prior demand from section 3.4, yields outlinks in the correct regime. The stopping criterion for the iterative method was set on the differences between the objective function < 1E-06. We varied the starting solution (starting from $\mathbf{D}_0$ or the Nadir (f_1^N) solution) and method constraining $\Delta\mathbf{D}$ per iteration (unconstrained or a binary search with ε_α =0.01 or 0.05). Results are shown in Fig. 4, the number of iterations performed and objective function values in Table 1.

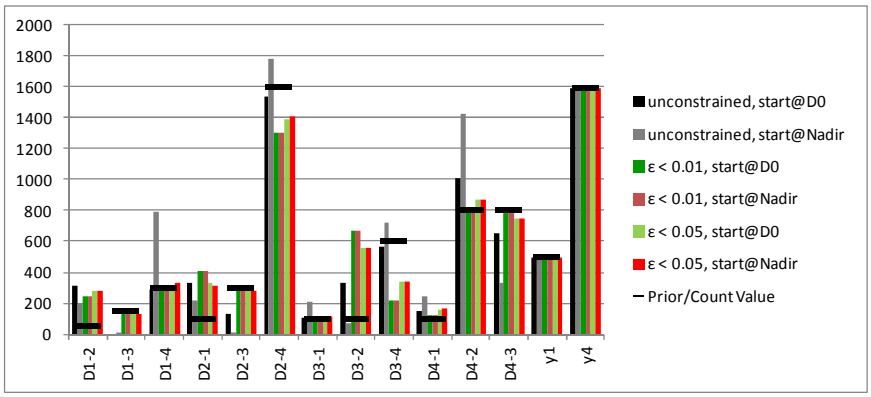

Fig. 4: posterior demand and link flows compared to prior demand and count values

Scenario	unconstrained, start@D0	unconstrained, start@Nadir	ε < 0.01, start@D0	ε < 0.01, start@Nadir	ε < 0.05, start@D0	ε < 0.05, start@Nadir
iterations needed	2	3	2	2	2	2
objf_1 (prior)	298042	1071083	687375	687487	430958	424353
objf_2 (counts)	1.020	0.003	0.026	0.026	1.785	1.671
Objf	3.669E-07	1.1928E-09	9.508E-09	9.512E-09	6.419E-07	6.009E-07

Table 1: number of iterations and objective function values after optimization

These results show that all scenarios yield solutions that are usable in practice: differences in count values are well within ranges that would be considered as uncertainty of the observed values. Considering differences between observed and estimated flow values and objective function,

[1] Note that setting w=1 does not make sense, since the ODmatrix is the variable to be optimized and can directly be set in the upper level.

constraining $\Delta\mathbf{D}$ leads to better solutions at the cost of more difference between prior and posterior matrix (but in the scenarios in this section this component of the objective function is ignored). Choosing a different starting solution and/or constraining method result in different solutions, indicating that the problem has multiple solutions, due to the problem being underspecified. Starting from Nadir yields better solutions for all scenarios contradicting expected behaviour. However, this can be a coincidence due to an arbitrary stop criterion value, further note that the unconstrained scenario starting from Nadir does require an extra iteration.

Furthermore, runs starting with a prior matrix yielding outlinks operating within their insensitive range were conducted, which showed that the method proposed in 4.4 operates as expected. In some runs however, in later iterations the algorithm ended up in the same insensitive range again, indicating that the constraints might need to be persisted during the entire run.

5.2. Varying weights in upper level and test of constraints forcing convergence

In this section we test scenario's in which we set *w*>0, to show the effect of normalisation and to test the method for constraining $\Delta\mathbf{D}$ per iteration as described in section 4.3 to prevent overshooting. The risk of overshooting is proportional to the number of interdependencies that are affected by changes to the demand made by the upper level, which is on its turn, proportional to the number of interdependencies that exist in the network and the amount of change that needs to be made to the OD matrix. This means that a network with more dependencies (i.e. lots of supply constrained outlinks) will be more sensitive to overshooting, especially when the upper level simultaneously changes demand on multiple OD pairs (which can be caused by a high weight on the prior matrix causing the upper level to distribute changes over all sensitive OD pairs, large numbers of (inconsistent) traffic counts and/or traffic counts on multiple outlinks). We changed the example by setting *w*=0.5, which proved to add enough interdependencies to demonstrate performance of the convergence method from section 4.3.

Setting *w*=0.5 should result in components f_1 and f_2 contributing equally to the objective function; i.e.: the ratio between f_1 and f_2 should converge towards 0.5. Fig. 5 shows values of objective function components f_1 and f_2 and the objective function total F for a run without constraints on $\Delta\mathbf{D}$ over iterations. From this figure, a repetitive cycle can be seen where the components converge towards a ratio of 0.5 during three subsequent iterations, but shoots out of convergence every fourth iteration (this first occurs in the second iteration), where after the process repeats. Although the method is not converging (objective function values do not substantially decrease), this does show that the normalisation scheme works. Other runs using values of *w* in the interval <0,0.6] all resulted in similar graphs in which the ratio of f_1 and f_2 converges to *w* during converging iterations, although the frequency of shooting out of convergence differed. Whenever *w* was set to a value greater than 0.6, the algorithm did converge. Apparently in these situations, the high weight on the prior matrix fixes the state of all turns by keeping demand within the original interval of piece wise functions $\alpha_i(D_{i'j'})$.

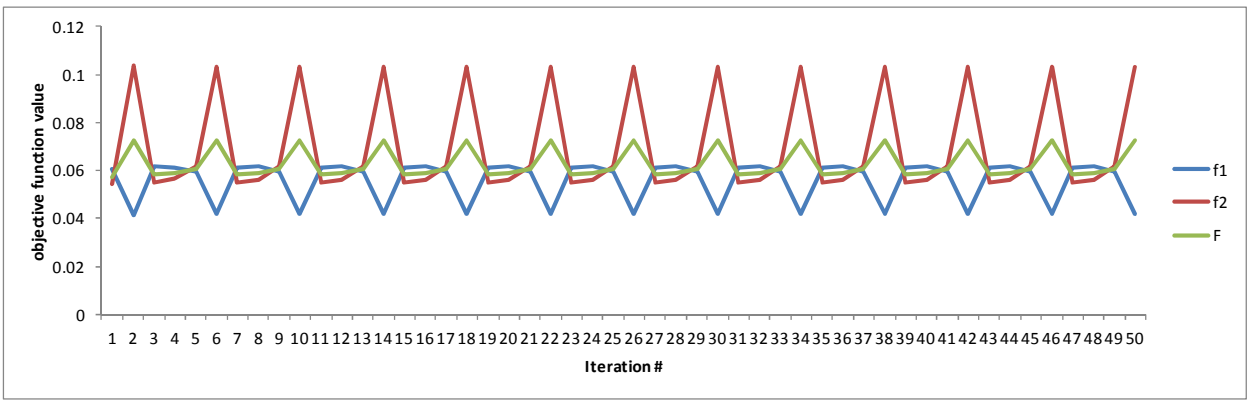

Fig. 5: value of objective function and its components for unconstrained run with *w* = 0.5

Applying a constraint on $\Delta\mathbf{D}$ using the binary searches with $\varepsilon_\alpha = 0.01$ yields slightly better convergence as indicated by Fig. 6, but the objective function value still does not substantially decrease. When looking at the course of $\mathbf{A}$ (Fig. 7) it becomes clear that the tolerance of 0.01 is violated in every iteration. This indicates that secondary interaction effects are a major contributor to the change of alpha and thus must be included to solve the problem in this case.

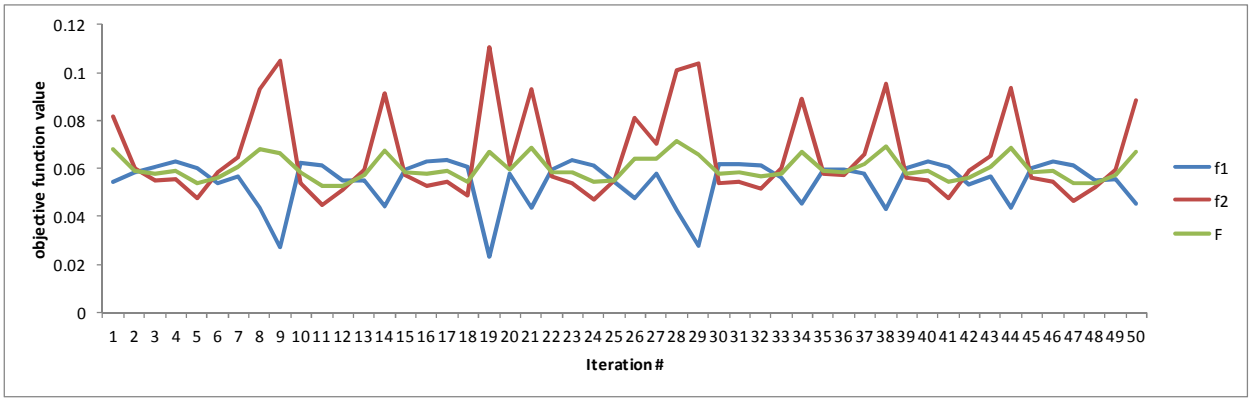

Fig. 6: value of objective function and its components for constrained run with w = 0.5 and $\varepsilon_\alpha = 0.01$

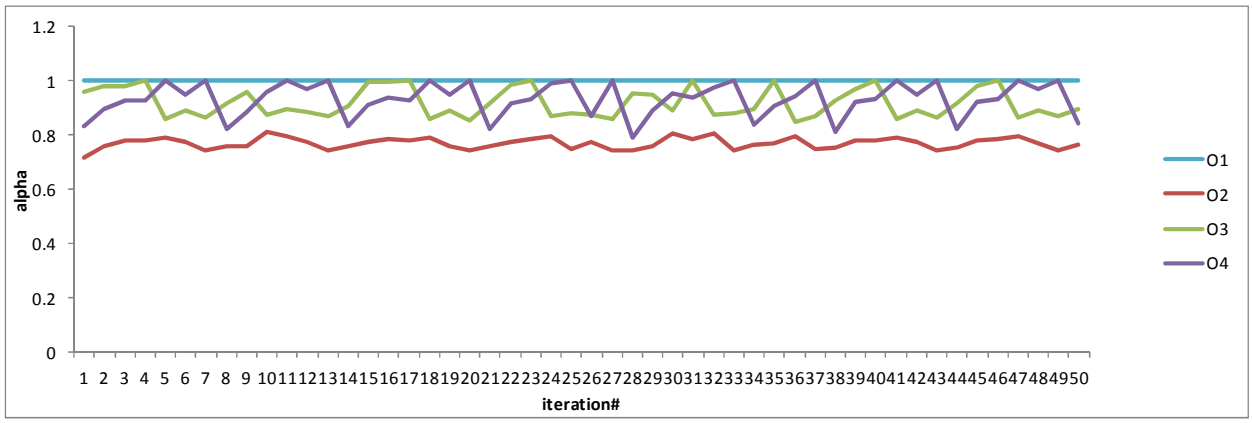

Fig. 7: value of alpha for constrained run with $w = 0.5$ and $\varepsilon_\alpha = 0.01$

Trying to force convergence, we conducted additional runs in which we constrained $\Delta\mathbf{D}$ directly and setting $\Delta\mathbf{D}$ depending on the quality of the approximation, where the difference between the first order approximation and the second order approximation around $\mathbf{D}$ was used as a (rough) quality measure for the approximation. We also conducted runs in which we applied the method of successive averages in combination with the constraints. These runs did show improvement in convergence, but did not result in lower objective function values, confirming that secondary interaction effects should be included to solve this problem.

6. Discussion and conclusions

We conclude with a discussion of our findings in 6.1 and the conclusions based on these findings in 6.2.

6.1. discussion

Using STAQ (or other semi dynamic assignment models with a node model that accounts for supply constraints) for matrix estimation has some methodological advantages over full DTA models. The lack of a time dimension and direct use of turn based reduction factors calculated by STAQ makes the problem more tractable, whereas the possibility to use marginal simulation only using the node model potentially decreases required calculation time and makes the solution method more scalable.

The matrix estimation problem for STAQ is a bi-level optimization problem in the form of a Stackelberg game. In order to solve it, the response of the lower level must be included in the upper level optimization in the form of partial derivatives of the assignment matrix to the demand. Most methods in literature approximate the response using complete runs of the assignment model entailing both large calculation times and tedious tuning of algorithmic parameters. The method proposed in this paper approximates the response function of a path by solving only the node models of nodes encountered on that path.

In this paper the matrix estimation problem is only solved on the node level, thus taking into account interaction effects between demand and supply on a node, causing flow metering. Interaction effects on the level of links and paths (i.e. route choice and spillback) were considered exogenous and their influence is left for further research. Also, we assumed that secondary interaction effects (additional changes in demand due to changes in the response function as a result of changes in demand) are negligible. This means that we neglected the fact that when simultaneously changing demand for multiple OD pairs the response might not be simply the sum of the effects of changing each OD pair sequentially.

To be able to use the node model in the lower level, the relationship between link flows and turn demand for a node were made explicit. The supply constraints of the node model, together with its SCIR actually define this relationship through the inlink based reduction factors. It was shown that the reduction factor of an inlink as a function of demand on some turn on the node is continuous on its positive domain, can be constructed piece-wise, and is differentiable almost everywhere. At each interval of the piece wise function, a reduction factor is determined by the same constraint, and at each non-differentiable point, a switch between active constraints occurs. Within each interval, the reduction function is linear when the inlink is demand constrained or supply constrained by an outlink to which at least one demand constrained turn exists; otherwise this function has the form of $c_1/(c_2 D_{ij} + c_3)^2$. Furthermore it was shown that the function is either monotonously increasing or decreasing on its positive domain and that the directed capacity constrained node model introduces secondary interaction effects between potentially all turns on a node. This means that when simultaneously changing demand for multiple OD pairs the response might not be simply the sum of the effects of changing each OD pair sequentially.

For the upper level a common objective function is proposed containing a first component minimizing differences between estimated and observed link flows and a second component minimizing differences between prior and posterior demand matrix, both components using MSE as a distance function. A weighing parameter was added to the objective function, which was normalized using true Utopia points for the first component and approximated and true Nadir points for the first and second component respectively.

To improve convergence, we proposed to constrain the amount of change that can be made to the OD matrix within one iteration related to the approximation error due to the linear approximation and piece wise character of the relation between turn demand and reduction factors. For this, a binary search was used to translate a tolerance on the approximation error into lower and upper bounds for the elements in the OD matrix to be used in the next iteration.

For cases where the response function is insensitive due to inconsistency between prior matrix and observed link flow a method was added that changes the prior demand forcing consistency before starting the matrix estimation procedure.

Several test runs where conducted using a prototype, showing that the method yields solutions usable in practice when only differences between observed and estimated link flows are considered. In this case, adding constraints to the amount of allowed demand-change per iteration leads to slightly better solutions. Runs with added constraints and runs using a different start solution yield different solutions, indicating that the problem is underspecified in this case. Test runs using weights in the interval <0,0.6] show that the normalisation of the objective function works as intended, but that the algorithm does not converge due to switches between active constraints within the node model. Apparently, higher weights restrict turn demand enough to keep the original active constraints intact as these runs do leading to convergence. In the case of *w>0* applying a constraint on the amount of allowed demand-change yields slightly better convergence, but the objective function value still does not substantially decrease. It is shown that the tolerance that was set on the change of reduction factors was violated in every iteration indicating that secondary interaction effects are a major contributor to the change of alpha and thus should be included to solve the problem in these cases.

6.2. conclusions

Further research is needed on how to efficiently determine, calculate and include relevant secondary (or even higher order) interaction effects. Furthermore the proposed method needs to be generalized to path and network level and extended incorporating route choice.

From the theoretical insights from sections 3.5 and the examples in section 5.2 it becomes clear that secondary interaction effects are a major part of the response function and should be included in its approximation. To the best of our knowledge, this insight has not been recognized before. Possible reasons for this are that, although not explicitly, 'conventional' methods that use complete runs of the assignment model perturbing multiple OD pairs at the same time (e.g. SPSA) do capture these effects. Furthermore, these effects might reside in the shadow of other interaction effects on real life networks such as route choice and spillback and departure time choice in the case of DTA models, which were deliberately excluded from this paper. Other reasons why this problem might be overlooked is the general underspecification of most matrix estimation applications and the relative few occurrences of the problem in a large scale network, but is exposed by looking at the level of nodes in this paper.

If the use of explicit reduction factors for matrix estimation is to be pursued, further research is needed on how to efficiently determine, calculate and include relevant secondary (or even higher order) interaction effects. A further enhancement could be to use exact derivatives for each interval of $\alpha_i(D_{i'j'})$ marking a switch between active constraints, instead of using linear approximated point derivatives.

Once such a method is found, the method should be generalized from node to path level. Although the theoretical framework for this generalization is already described in section 2.1, the translation from turn to paths will lead to practical problems such as possible inconsistency of constraints on different nodes on a path or (non-obvious) inconsistency between observed link flows.

Consecutively, the method should be generalized to include effects of spillback on the network level. Since spillback effects can already be captured on the node level by the $\alpha_i(D_{i'j'})$ relation through $\tilde{R}_4$, the challenge here is to develop some marginal simulation method to transfer spillback effects over links whenever demand is altered by the matrix estimation method in such a way that spillback effects substantially change. The event based submodel that handles spillback within STAQ (the so called 'queuing phase') would be a good starting point for development of such a method. Finally, the method needs further generalisation to include route choice.

Acknowledgements

We would like to thank Erik-Sander Smits for his valuable suggestions regarding mathematical formulations.

References

Alpcan, T. (2013). A framework for optimization under limited information." *Journal of Global Optimization* 55.3 (2013): 681-706.

Bliemer, M.C.J., Brederode, L., Wismans, L., and Smits, E.S. (2012). Quasi-dynamic network loading: adding queuing and spillback to static traffic assignment. *Presented at the Annual Transportation Research Board Meeting 2012, Washington DC, USA.*

Bliemer, M.C.J, Raadsen, M., Smits, E.-S., Brederode, L.J.N., Wismans, L.J.J., Zhou, B. and Bell, M. (2013) Consistent capacity constrained stochastic static traffic assignment with residual point queues, *presented on the 93th annual meeting of the Transportation Research Board, Washinghton D.C., USA.*

Brederode, L.J.N. , Bliemer, M.C.J. , Wismans, L.J.J. (2010) STAQ – Static Traffic Assignment with Queuing - *Proceedings of the European Transport Conference 2010*

Cascetta, E. (2001) Transportation Systems Engineering: Theory and Methods. *Kluwer Academic Publishers, Dordrecht.*

Cipriani, E. , Florian, M. , Mahut, M. , Nigro, M. (2011) - A gradient approximation approach for adjusting temporal origin–destination matrices - *Transportation Research Part C*, Vol. 19 pp 270-282

Flötteröd, G. and Rohde, J. (2011). Operational macroscopic modeling of complex urban road intersections. *Transportation Research Part B*, Vol. 45, pp 903 – 922

Frederix, R., Tampère, C., & Viti, F. (2010). Dynamic origin-destination estimation in congested networks. *Presented at the DTA conference 2010 , Takayama, Japan.*

Köhler, E. and M Strehler (2012) Combining static and dynamic models for traffic signal optimization. *Compendium of papers of the 15th meeting of the EURO Working Group on Transportation*

Lasdon, L.S. , Waren, A.D., Jain, A., Ratner, M.W. (1975) Design and testing of a generalized reduced gradient code for non linear optimization

Smith M.J. (2012) A link-based elastic demand equilibrium model with capacity constraints and queueing delays. *Transportation Research Part C*, Vol. 29, pp 131-147

Smith M.J. , Huang, W., Viti, F. (2013) Equilibrium in Capacitated Network Models with Queueing Delays, Queue-storage, Blocking Back and Control, *Procedia - Social and Behavioral Sciences*, Volume 80, 7 June 2013, pp. 860-879

Smits, E-S., Bliemer, M.C.J., Pel, A., van Arem, B. (2014)- A Family of Macroscopic Node Models – *submitted*

Spall, JC (1998), 'An Overview of the Simultaneous Perturbation Method for Efficient Optimization', *John Hopkins University APL Technical Digest*, Vol. 19, No. 4.

Tampère, C.M.J. , Corthout, R., D. Cattrysse and L.H. Immers (2011) A generic class of first order node models for dynamic macroscopic simulation of traffic flows, *Transportation Research Part B*, Vol. 45, pp. 289-309.

Appendix: solution algorithm for node model with directed capacity proportional SCIR

(0) Given: $\mathbf{D}_n, \mathbf{R}_n, \mathbf{C}_n$

(1) Initialise: $I_j^1 = \{i \mid D_{ij} > 0\} \quad \forall j \in J$ # set of considered inlinks per outlink in iteration 0

$J^1 = \left\{ j \mid \sum_{i \in I} S_{ij} > 0 \right\}$ # set of considered outlinks in iteration 0

$\tilde{R}_j^1 = R_j \quad \forall j \in J$ # available supply per outlink in iteration 0

$k = 1$ # iteration number

(2) $C_{ij} = \frac{D_{ij}}{\sum_{j \in J^1} D_{ij}} C_i \quad \forall ij \in IJ$ # directional capacity per turn

(3) While $J^{k+1} \neq \emptyset$ # Start of loop over outlinks

(4) $\beta_j^k = \frac{\tilde{R}_j^k}{\sum_{i \in I_j^k} C_{ij}} \quad \forall j \in J \mid I_j^k \neq \emptyset$ # determine potential outlink restricting factors

(5) $\beta_{\hat{j}}^k = \min_j \{\beta_j^k, 1\}$ # determine most restricting constraint

(6) $\hat{j} = \arg\min \{\beta_j^k\}$ # determine causative outlink

(7) If $\exists i \in I_{\hat{j}}^k \mid D_i \leq \beta_{\hat{j}}^k C_i$ # if there exist demand constrained inlink(s)

(8) $\forall i \in I_{\hat{j}}^k \mid D_i \leq \beta_{\hat{j}}^k C_i$ do: # for these demand constrained inlink(s)

(9) $y_{ij} = D_{ij} \quad \forall j \in J$ # fix turnflows to turndemand

(10) $\tilde{R}_j^{k+1} = \tilde{R}_j^k - D_{ij} \; \forall j \in J$ # reduce available supply of affected outlinks

(11) $\forall j \in J^k$ do: # for all still considered outlinks

(12) $I_j^{k+1} = I_j^k \setminus \{i\}$ # remove from set of inlinks competing for j

(13) If $I_j^{k+1} = \emptyset$: # if set of considered inlinks is now empty

(14) $J^{k+1} = J^k \setminus \{j\}$ # remove outlink from set of considered outlinks

(15) End if

(16) Next j

(17) Next i

(18) Else if $\sum_{j \in J_i} D_{ij} > \beta_{\hat{j}}^k C_i \; \forall i \in I_{\hat{j}}^k$ #there are only supply constrained inlinks left

(19) $\forall i \in I_{\hat{j}}^k$ do: # for all inlinks constrained by most restrictive outlink

(20) $y_{ij} = \beta_{\hat{j}}^k C_{ij} \quad \forall j \in J$ # fix turnflows to match available supply

(21) $\tilde{R}_j^{k+1} = \tilde{R}_j^k - \beta_{\hat{j}}^k C_{ij} \quad \forall j \in J$ # reduce available supply of affected outlinks

(22) $\forall j \in J^k$ do: # for all still considered outlinks

(23) If $j \neq \hat{j}^k$ # if not the most restrictive outlink

(24) $I_j^{k+1} = I_j^k \setminus I_{\hat{j}}^k$ # remove inlinks constrained by most restrictive outlink

(25) If $I_j^{k+1} = \emptyset$ # if set of considered inlinks is now empty

(26) $J^{k+1} = J^k \setminus \{j\}$ # remove outlink from set of considered outlinks

(27) endif

(28) Else if $j = \hat{j}^k$ # if most restrictive outlink

(29) $J^{k+1} = J^k \setminus \{\hat{j}^k\}$ # remove outlink from set of considered outlinks

(30) Endif

(31) Next j

(32) Next i

(33) Endif

(34) k:=k+1

(35) Endwhile

Appendix: calculation of derivatives for numerical example when $457 \le D_{14} \le 800$

$$\frac{dy_{34}}{dD_{14}} = d\frac{\tilde{R}_4^1 C_{34}}{\sum_{i\in I_4^1} C_{i4}} / dD_{14} = d\frac{\tilde{R}_4^1\left(D_{34}/\sum_{j\in J} D_{3j}\right)C_3}{\sum_{i\in I_4^1}\left(\frac{D_{i4}}{\sum_{j\in J} D_{ij}} C_i\right)} / dD_{14}$$

$$= d\frac{\tilde{R}_4^1\left(D_{34}/(D_{31}+D_{32}+D_{34})\right)C_3}{\frac{C_1 D_{14}}{D_{12}+D_{13}+D_{14}} + \frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}}{D_{31}+D_{32}+D_{34}}} / dD_{14} \qquad \text{(a1)}$$

$$= \frac{\left(-\tilde{R}_4^1\left(D_{34}/(D_{31}+D_{32}+D_{34})\right)C_3\right) * C_1 * (D_{12}+D_{13})}{\left[D_{14}\left(C_1 + \frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}}{D_{31}+D_{32}+D_{34}}\right) + (D_{12}+D_{13})\left(\frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}}{D_{31}+D_{32}+D_{34}}\right)\right]^2}$$

$$= \frac{-\tilde{R}_4^1 C_{34} * C *_1 (D_{12}+D_{13})}{\left[D_{14}(C_1 + C_{24} + C_{34}) + (D_{12}+D_{13})(C_{24}+C_{34})\right]^2}$$

$$\frac{dy_{14}}{dD_{14}} = d\frac{\tilde{R}_4^1 C_{14}}{\sum_{i\in I_4^1} C_{i4}} / dD_{14} = d\frac{\tilde{R}_4^1\left(D_{14}/\sum_{j\in J} D_{1j}\right)C_1}{\sum_{i\in I_4^1}\left(\frac{D_{i4}}{\sum_{j\in J} D_{ij}} C_i\right)} / dD_{14}$$

$$= d\frac{\tilde{R}_4^1\left(D_{14}/(D_{12}+D_{13}+D_{14})\right)C_1}{\frac{C_1 D_{14}}{D_{12}+D_{13}+D_{14}} + \frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}}{D_{31}+D_{32}+D_{34}}} / dD_{14} \qquad \text{(a2)}$$

$$= \frac{\tilde{R}_4^1 C_1 (D_{12}+D_{13})\left(\frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}}{D_{31}+D_{32}+D_{34}}\right)}{\left[C_1 D_{14} + \left(\frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}}{D_{31}+D_{32}+D_{34}}\right)(D_{12}+D_{13}+D_{14})\right]^2}$$

$$= \frac{\tilde{R}_4^1 C_1 (D_{12}+D_{13})(C_{24}+C_{34})}{\left[C_1 D_{14} + (C_{24}+C_{34})(D_{12}+D_{13}+D_{14})\right]^2}$$

$$\frac{dy_{24}}{dD_{14}} = d\frac{\tilde{R}_4^1 C_{24}}{\sum_{i\in I_4^1} C_{i4}} / dD_{14} = d\frac{\tilde{R}_4^1\left(D_{24}/\sum_{j\in J} D_{2j}\right)C_2}{\sum_{i\in I_4^1}\left(\frac{D_{i4}}{\sum_{j\in J} D_{ij}} C_i\right)} / dD_{14}$$

$$= d\frac{\tilde{R}_4^1\left(D_{24}/(D_{21}+D_{23}+D_{24})\right)C_2}{\frac{C_1 D_{14}}{D_{12}+D_{13}+D_{14}} + \frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}}{D_{31}+D_{32}+D_{34}}} / dD_{14} \tag{a3}$$

$$= \frac{\left(-\tilde{R}_4^1\left(D_{24}/(D_{21}+D_{23}+D_{24})\right)C_2\right) * C_1 * (D_{12}+D_{13})}{\left[D_{14}\left(C_1 + \frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}}{D_{31}+D_{32}+D_{34}}\right) + (D_{12}+D_{13})\left(\frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}}{D_{31}+D_{32}+D_{34}}\right)\right]^2}$$

$$= \frac{\left(-\tilde{R}_4^1 C_{24}\right) * C_1 * (D_{12}+D_{13})}{\left[D_{14}\left(C_1 + C_{24} + C_{34}\right) + (D_{12}+D_{13})\left(C_{24}+C_{34}\right)\right]^2}$$

$$\frac{dy_{43}}{dD_{14}} = d\frac{\tilde{R}_2^3 C_{43}}{\sum_{i\in I_3^2} C_{i3}} / dD_{14} = d\left[\tilde{R}_3^1 - \frac{\tilde{R}_4^1 D_{13} C_2 C_{23} C_{43}}{\left(D_{14}C_1 + \frac{D_{24}(D_{12}+D_{13}+D_{14})}{D_{21}+D_{23}+D_{24}} C_2 + \frac{D_{34}(D_{12}+D_{13}+D_{14})}{D_{31}+D_{32}+D_{34}} C_3\right) C_{43}}\right] / dD_{14}$$

$$= d\left[\tilde{R}_3^1 - \frac{\tilde{R}_4^1 D_{13} C_2 C_{23}}{D_{14}C_1 + \frac{D_{24}(D_{12}+D_{13}+D_{14})}{D_{21}+D_{23}+D_{24}} C_2 + \frac{D_{34}(D_{12}+D_{13}+D_{14})}{D_{31}+D_{32}+D_{34}} C_3}\right] / dD_{14} \tag{a4}$$

$$= \frac{\tilde{R}_4^1 D_{13} C_2 C_{23}\left(C_1 + \frac{C_2 D_{24}}{D_{21}+D_{23}+D_{24}} + \frac{D_{34} C_3}{D_{31}+D_{32}+D_{34}}\right)}{\left[C_1 D_{14} + \frac{C_2 D_{24}(D_{12}+D_{13}+D_{14})}{D_{21}+D_{23}+D_{24}} + \frac{C_3 D_{34}(D_{12}+D_{13}+D_{14})}{D_{31}+D_{32}+D_{34}}\right]^2}$$

$$= \frac{\tilde{R}_4^1 D_{13} C_2 C_{23}\left(C_1 + C_{24} + C_{34}\right)}{\left[C_1 D_{14} + (D_{12}+D_{13}+D_{14})C_{24} + (D_{12}+D_{13}+D_{14})C_{34}\right]^2}$$