Transportation Research Part B 167 (2023) 1–31

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Travel demand matrix estimation for strategic road traffic assignment models with strict capacity constraints and residual queues

Luuk Brederode [a,b,*], Adam Pel [a], Luc Wismans [b,c], Bernike Rijksen [d], Serge Hoogendoorn [a]

[a] Department of Transport & Planning, Delft University of Technology, Delft, The Netherlands
[b] DAT.mobility – A Goudappel Company, Deventer, The Netherlands
[c] Centre for Transport Studies, Faculty of Engineering Technology, University of Twente, Enschede, The Netherlands
[d] Discrete mathematics and mathematical programming, University of Twente, Enschede, The Netherlands

ARTICLE INFO

Keywords:
Demand matrix estimation
Static traffic assignment model
Capacity constrained
Congestion patterns
Route travel times
Prior OD demand matrix
Large scale
Strategic
mathematical properties

ABSTRACT

This paper presents an efficient solution method for the matrix estimation problem using a static capacity constrained traffic assignment (SCCTA) model with residual queues. The solution method allows for inclusion of route queuing delays and congestion patterns besides the traditional link flows and prior demand matrix whilst the tractability of the SCCTA model avoids the need for tedious tuning of application specific algorithmic parameters.

The proposed solution method solves a series of simplified optimization problems, thereby avoiding costly additional assignment model runs. Link state constraints are used to prevent usage of approximations outside their valid range as well as to include observed congestion patterns. The proposed solution method is designed to be fast, scalable, robust, tractable and reliable because conditions under which a solution to the simplified optimization problem exist are known and because the problem is convex and has a smooth objective function.

Four test case applications on the small Sioux Falls model are presented, each consisting of 100 runs with varied input for robustness. The applications demonstrate the added value of inclusion of observed congestion patterns and route queuing delays within the solution method. In addition, application on the large scale BBMB model demonstrates that the proposed solution method is indeed scalable to large scale applications and clearly outperforms the method mostly used in current practice.

1. Introduction

Traditionally, travel demand origin-destination (OD) matrix estimation for road traffic is a process in which a prior demand matrix specifying travel demand between origin and destination nodes in the road network is enriched using observed flows on link level. It is a bi-level optimization problem where in the upper level, the OD demand matrix is altered to minimize differences between observed and modelled link flows and between the prior and modelled OD demand matrix, while in the lower level a traffic assignment model is

* Corresponding author at: Department of Transport & Planning, Delft University of Technology, Delft, The Netherlands.
E-mail address: lbrederode@dat.nl (L. Brederode).

https://doi.org/10.1016/j.trb.2022.11.006
Received 24 January 2022; Received in revised form 21 September 2022; Accepted 15 November 2022
Available online 28 November 2022

used solving a user equilibrium problem translating the new OD demand into modelled link flows.

Different traffic assignment models can be used in the lower level, varying by capability and complexity. In this paper, we shall use the classification of assignment models described in (Bliemer et al., 2017) when referring to specific assignment model types. The majority of strategic transport model systems used today use static capacity restrained traffic assignment (SCRTA) models. SCRTA models assume separable monotonously increasing link travel time functions, yielding computationally fast and scalable models with desirable convergence properties needed for strategic large-scale transport model systems. Matrix estimation methods using SCRTA models have been studied extensively and are readily available, see e.g. (Cascetta, 2009) and references herein.

However, link flows and speeds from SCRTA models do not correspond to empirically supported macroscopic traffic flow theory that describes the relation between flow, speed and density (the fundamental diagram). This is caused by the lack of a strict capacity constraint and omission of a congested branch and storage constraints in travel time functions used in SCRTA models.

The most common solution for this problem is to is to switch to a (macroscopic) dynamic capacity and storage constrained traffic assignment (DCSTA) model. This model class does incorporate capacity constraints and hence physical effects of congestion and is more realistic compared to the SCRTA model class. However, the dynamic nature of DCSTA class models causes the mapping from OD demand to link flows to be extended with a temporal dimension, causing temporal correlations between model variables which makes estimation of OD demand matrices much more tedious and are therefore limited to small sized networks (see e.g.: Toledo et al., 2015).

1.1. Contribution, positioning and outline

This paper presents an efficient solution method for the matrix estimation problem using a static capacity constrained traffic assignment (SCCTA) model with residual queues. This type of assignment model is described in e.g. (Bliemer et al., 2014; Brederode et al., 2019; Bundschuh et al., 2006; Lam and Zhang, 2000; Smith, 2013), implemented in OmniTRANS Transport planning Software, PTV VISUM and Aimsun Next and applied in various contexts (e.g.: Brederode et al., 2016; Huang et al., 2020; Tajtehranifard, 2017; Tsanakas et al., 2020). The solution method is developed to be used in the context of strategic transport models where the user wants to refine prior OD demand from a demand model with information on link and route level (e.g.: loop detector and floating car data).

Note that this paper only considers SCCTA models that incorporate capacity constraints within the link cost functions, thereby allowing for explicit residual queues in model outcomes, whereas SCCTA models in which the capacity constraints are only added as upper bounds on link flows (often referred to as ‘extended Beckmann’ or ‘capacitated Beckmann’; e.g. (Correa et al., 2004; Larsson and Patriksson, 1999; Nie et al., 2004; Yang and Yagar, 1994)) do not allow for residual queues and are therefore outside of scope of this paper. For reasons of brevity, in the remainder, we shall simply refer to ‘the SCCTA model’ (i.e.: omitting ‘with residual queues’) when referring to the TA model type considered in this paper.

By using a SCCTA model, the solution method combines the favorable properties of SCRTA and DCSTA models in the context of matrix estimation. Similar to DCSTA models, the strict capacity constraints of SCCTA models account for flow metering effects of active bottlenecks, which allows direct comparison and usage of observed flows that are reduced by upstream bottlenecks. The strict capacity constraints also extend the supported set of datatypes for estimation with observed (link- or route-) travel times and observed congestion patterns because queues are explicitly modelled. Similar to SCRTA models, the static nature of SCCTA models removes the temporal dimension in the relation between link flows and OD-demands, which allows demand estimation at a time-aggregate level. This avoids temporal correlations between model variables which causes the solution method to be relatively fast and suitable for large scale networks.[1]

Note that there is a big difference in what is considered a large network in the DSCTA compared to the SCCTA context. In the DSCTA context, networks containing in the order of tens of thousands OD pairs are considered large scale (e.g.: Castiglione et al., 2021; Osorio, 2019a), whereas SCCTA models are typically applied on networks containing in the order of millions OD pairs (Brederode et al., 2019). The proposed solution method presented In this paper targets networks that are considered large in the SCCTA context. To the best of the authors’ knowledge, the sheer size of these networks prevent usage of any DSCTA based method. Therefore, this paper takes the SCCTA model as a starting point for the matrix estimation method and does not include a comparison between DSCTA and SCCTA models. Instead, the interested reader is referred to (Brederode et al., 2019).

Further note that, similarly to the methodology proposed in this paper, the quasi dynamic approaches by (Marzano et al., 2018; Zijpp, Van der, 1996) also employ time-aggregation on observed variables, but they do so to reduce -not avoid- temporal correlations between model variables. Furthermore, the meta model approach in (Osorio, 2019b), the computational graph approach in (Ma et al., 2020; Wu et al., 2018) and various SPSA-based approaches (e.g.: Qurashi et al., 2020) show promising results in handling and reducing temporal correlations that exist in DCSTA models.[2] The approach proposed in this paper is different because it is only applicable to SCCTA (and SCRTA) models, which means that it solves a less complex problem which should make it more efficient compared to the more generic approaches developed for DCSTA models.

The remainder of this paper is organized as follows. Section 2 defines the matrix estimation problem for SCCTA models, the SCCTA model itself, solution methods to similar problems currently used in practice and the proposed solution method. Section 3 elaborates on the proposed solution method that uses a combination of analytical and approximated relationships in the lower level as well as the

[1] Note that in reality (and therefore also in time-aggregated observed variables), temporal correlations do occur, and -just like with any other static traffic assignment model- this knowledge must be taken into account when assessing the SCCTA models outcomes.

[2] Note that (Osorio, 2019) reports that her meta model approach should be transferable to any traffic assignment model, thereby making it applicable to the strategic context, but to the best of the authors knowledge, this has not been tested yet (Wu et al., 2018).

mathematical properties of its solution(s). In Section 4 the added value of the proposed solution method is demonstrated using several test case applications on the small Sioux Falls model, whereas Section 5 presents application of the proposed solution method on a large scale strategic transport model, demonstrating its performance and scalability. We end with discussion and conclusions in Section 6.

2. The matrix estimation problem for SCCTA models

In this section, the travel demand matrix estimation problem is defined for road traffic using input data consisting of a prior demand matrix, observed link flows and route queuing delays. Note that to the best of our knowledge, the inclusion of route queuing delays in the context of static traffic assignment models is novel, and it is only possible because strict capacity constraints are included. In Section 2.3.3 we shall present another novelty in the context of static traffic assignment models which is to include observed congestion patterns in the optimization problem.

2.1. Problem formulation

Consider a general network $G = (N, L)$ where N denotes the set of nodes n and L denotes the set of directed links l. Let $R \subset N$ and $S \subset N$ be the set of origins r and destinations s, respectively and $RS = R \times S$ the set of all OD-pairs rs. Furthermore, let $\widetilde{L} \subset L$ be the set of links for which flow has been observed ('observed links'). Then, the bi-level matrix estimation problem using a prior OD matrix, observed link flows and observed route queuing delays is defined as:

$$\begin{aligned} &\mathbf{D}^* = \underset{\mathbf{D}}{\operatorname{argmin}}(F) = \underset{\mathbf{D}}{\operatorname{argmin}}[f_1(\mathbf{D}, \mathbf{D}_0) + f_2(\mathbf{y}(\mathbf{D}), \widetilde{\mathbf{y}}) + f_3(\tau(\mathbf{D}), \widetilde{\tau})] \\ &\text{s.t. } \mathbf{y}(\mathbf{D}), \tau(\mathbf{D}) = assign(\mathbf{D}), \\ &\mathbf{D} \geq 0 \end{aligned} \quad , \tag{1}$$

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 $\widetilde{\mathbf{y}}$ denote vectors of current and observed link flows in $\widetilde{L}$, $\tau(\mathbf{D})$ and $\widetilde{\tau}$ denote vectors of current and observed route queuing delays (for the set $\widetilde{P}$ of routes for which travel time has been observed), while f_1, f_2 and f_3 denote distance functions measuring the differences between observed (or prior) and current OD demand, link flows and route queuing delays, respectively. In the lower level, the function *assign* represents the traffic assignment model used (i.e. here the SCCTA model described in Section 2.2).

Note that $\widetilde{\mathbf{y}}$, $\mathbf{D}_0$ and $\widetilde{\tau}$ contain aggregate variables observed over some period(s) of time. Therefore, the observed values in these vectors are in fact instances of some probability distribution. Although, when known, these distributions can be considered when solving the upper level, this is not subject of this paper. In the remainder we therefore choose the least squared error as distance function for all three components since it does not require any additional data on the distribution of the observed flow values, prior matrix or route queuing delays. Furthermore, we introduce parameters that allows for weighing and normalization of the three components in the objective function. Using least squared errors and weighting parameters w_1,w_2 and w_3, the objective function now reads:

$$F = w_1 \sum (\mathbf{D} - \mathbf{D}_0)^2 + w_2 \sum (\mathbf{y}(\mathbf{D}) - \widetilde{\mathbf{y}})^2 + w_3 \sum (\tau(\mathbf{D}) - \widetilde{\tau})^2 \tag{2}$$

2.1.1. Decomposition of observed link flows

As described in (Brederode and Verlinden, 2019) active bottlenecks in a network influence flow values both upstream (queues will form) and downstream (flow is metered). This means that an observed link flow value represents either 1) the unaffected travel demand for that link, 2) a proportion of the capacity of (a set of) upstream link(s), 3) the capacity of the normative (in terms of capacity deficit) downstream link or 4) a combination of these quantities.

Only flows measured under conditions (1) or (4) contain information about the absolute level of traffic demand, whereas flows measured under conditions (2) or (3) only contain information about network capacities and bottleneck locations (hence a lower bound on the level of traffic demand).

Because strict capacity constraints are lacking in SCRTA models, matrix estimation methods using these models do not take flow metering (2, 4) nor queuing effects (3) of bottlenecks into account. Instead, all traffic is (implicitly) considered to be unaffected (1), thereby forcing incorrect assumptions upon the estimation (Brederode and Verlinden, 2019). Formulated differently: SCRTA models and their matrix estimation methods assume that travel demand on route level equals route flow by definition, whereas in SCCTA models, route flow is lower than route demand on links downstream from the first bottleneck on the route. This advocates for the use of an SCCTA model and a different matrix estimation method, such that the conditions under which link flows are observed are considered during the estimation.

2.1.2. Solution methods: current practice

In SCRTA model context, bi-level problem (1) is typically solved by iteratively assigning the OD matrix from the upper level into the lower level to determine the relationships between link flows and OD-demands (assignment matrix $\mathbf{A}(\mathbf{D})$ of size $|\widetilde{L}|$ x $|RS|$) and the relationships between route queuing delays and OD-demands and then use these relationships to solve the upper level. In the SCRTA model context the relationships are considered constant while solving the upper level yielding the following response function for link

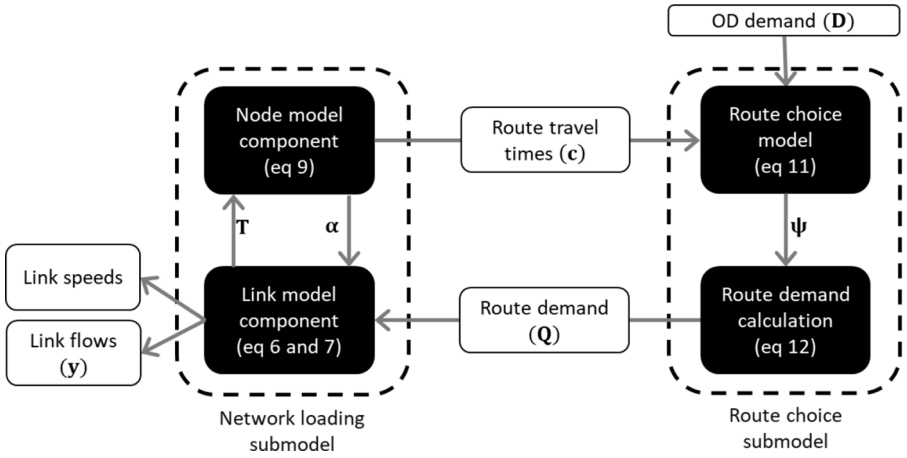


Fig. 1. Framework for SCCTA models.

flows:

$$\mathbf{y}(\mathbf{D}) = \mathbf{A}(\mathbf{D}_{k-1})\mathbf{D} \tag{3}$$

where $\mathbf{D}_{k-1}$ represents the OD demand from the previous upper level solution.

For SCRTA models, numerous solution algorithms using constant response functions have been proposed and successfully applied as summarized by e.g. (Abrahamsson, 1998). For SCCTA models, to the best of our knowledge, only the following three solution approaches described in (Brederode and Verlinden, 2019) have been proposed and/or applied. Below, these approaches are summarized, an extensive description of the practical implications for all three methods can be found in (Brederode and Verlinden, 2019; Brederode et al., 2017).

The longest and most widely used approach to estimate demand for SCCTA models is to estimate unconstrained link demand values from observed link flow values and apply a traditional SCRTA-based solution algorithm assuming Eq. (3) in the upper level. By using (estimated) link demands instead of directly observed link flows as input, this approach constructs a synthetic matrix estimation problem in which all observations adhere to condition 1 from Section 2.1.1, allowing usage of an SCRTA model. However, this approach does not use any information about local network conditions on observed links as the SCRTA model cannot provide it. Instead, this approach relies on the unconstrained link demand values that are derived using heuristics based on generic model-wide temporal demand distributions. This means that using this approach will yield an OD matrix that fits to the estimated link demand values, but it does not guarantee that the final assignment of this OD matrix using the SCCTA model will yield link flows that fit to the observed link flows. Because of this, these methods exhibit poor tractability and robustness.

The second approach is to use the SCCTA model to determine the assignment matrix and then apply matrix estimation only on unmetered demand, while assuming Eq. (3) in the upper level. This is the first solution approach that does not require usage of estimated link demands. This approach was tested on the Dutch regional strategic transport model of the Randstad Agglomeration (Brederode et al., 2017) and is implemented in the 2018 version of the strategic transport models of the Dutch province of Noord Brabant.

The last solution approach described in (Brederode and Verlinden, 2019) refers to an early version of the solution method that is described in this paper. To the best of our knowledge this is the only estimation method for SCCTA models that can include observed queuing delays and/or congestion patterns in the estimation. Note that all but the proposed solution approaches for SCCTA models assume Eq. (3) in the upper level. By doing so, these methods all treat the matrix estimation problem as if it were a Cournot-Nash game by omitting any sensitivities in the response function, whereas (Frederix et al., 2013; Maher et al., 2001; Yang, 1995) point out that it intrinsically is a Stackelberg game. Although incorrect in theory, given the widespread usage with SCRTA models, this appears to not be a problem in practice in this context.

However, strict capacity constraints cause the true response function to be more sensitive and less separable, which means that to solve it, $\mathbf{A}(\mathbf{D})$ should no longer be considered constant while solving the upper level. Instead, the sensitivity of link flows (and the sensitivity of route queuing delays) for changes in OD-demand need to be included in the response functions.

This was recognized in the DCSTA model context for which most approaches in literature use either direct finite differences (e.g.: Djukic et al., 2017; Frederix et al., 2013; Shafiei et al., 2017; Toledo and Kolechkina, 2013) or some form of the Simultaneous Perturbation Stochastic Approximation (SPSA) method (e.g.: Antoniou et al., 2015; Cantelmo et al., 2017; Cipriani et al., 2013; Tympakianaki et al., 2015) to approximate the sensitivity of link flows to changes in OD-demand $\partial\mathbf{y}(\mathbf{D})/\partial\mathbf{D}$and use it in a first order Taylor expansion around the current solution yielding the following response function for link flows:

$$\mathbf{y}(\mathbf{D}) = \mathbf{y}(\mathbf{D}_{k-1}) + \frac{\partial \mathbf{y}(\mathbf{D})}{\partial \mathbf{D}}\Big|_{\mathbf{D}_{k-1}} (D - \mathbf{D}_{k-1}). \tag{4}$$

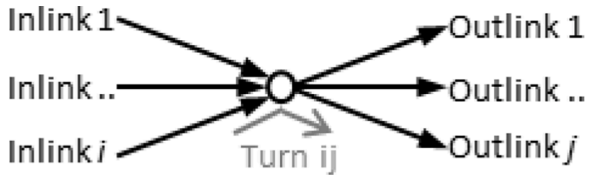


Fig. 2. Inlinks, outlinks and turns associated with a node.

In these studies, minimization of the upper level objective function in DCSTA context is done using a solver that can handle the quadratic objective function specified by (2) and (4) in combination with the linear constraints on link flows (4) and the bound constraints enforcing non-negativity in (2). Both the direct finite difference and SPSA approaches require additional assignment model runs in the lower level to determine the sensitivity of the link flows and therefore exhibit large calculation times and thus poor scalability. Also, SPSA-based approaches entail tedious tuning of application specific algorithmic parameters (e.g.: Cipriani et al., 2011).

2.2. SCCTA model formulation

This section describes the mathematical relationships within an SCCTA model, as these will be used by the solution method that will be proposed in 2.3. An SCCTA model consists of two submodels: a network loading submodel and a route choice submodel (Fig. 1). The network loading submodel uses route demand **Q** from the route choice submodel to calculate route travel times which are used by the route choice submodel to calculate route choice probabilities **ψ** to distribute OD demand over routes.

The network loading submodel uses this route demand to compute the resulting link flows and speeds and thereby (route) travel times. The most important components within the network loading submodel are the node model component that calculates flow acceptance factors **α** on links entering nodes with active capacity constraints, and the link model component that applies these factors to the route demands yielding turn demands **T,** which by aggregation yield link flows **y**. Note that, as mentioned in Section 1, some SCCTA models do not have a node model, but use link exit capacities instead. Further note that besides link flows (Section 2.3.1) the acceptance factors **α** also define the queuing delays (Section 2.3.2) and the congestion patterns (Section 2.3.3) used in the demand estimation.

The mathematical definition of the route choice submodel depends on the chosen traffic assignment problem formulation. In this paper the stochastic user equilibrium (SUE, Fisk, 1980) is chosen, which leads to the route choice submodel described in Section 2.2.4. Other (non-equilibrium and/or deterministic) assignment problem formulations may also be used with the SCCTA network loading submodel but are not described here, because fixed route choice probabilities are assumed in the upper level (i.e.: route fractions are assumed to be locally constant), and, similar to approaches used for SCRTA models, it is assumed that iterations between lower and upper level will solve the consistency problem between route choice probabilities and OD demands. The remainder of this subsection describes each of the components from the SCCTA model framework in more detail.

2.2.1. Link model component

The link model component determines link flows taking into account reductions due to active bottlenecks in the form of flow acceptance factors per link $\alpha_l \forall l \in L$, calculated by the node model component (Section 2.2.2). These flow acceptance factors are aggregated to the route-link level using:

$$\widehat{\alpha}_{lp} = \prod_{ij' \in IJ_{p,il}} \alpha_{ij'} \ . \tag{5}$$

where $\widehat{\alpha}_{lp}$ denotes the acceptance factor due to upstream bottlenecks at link l on route p and $IJ_{p,il}$ represents the set of turns on route p up to (and including) the turn from link i to link l. Note that the matrix $\widehat{\alpha}$ containing acceptance factors for all route-link combinations is actually the route-level equivalent of assignment matrix **A** introduced in 2.1.2 Further note that we define $\widehat{\alpha}_{lp} = 0$ for all l not used by p, such that it also doubles as a route-link incidence indicator.

Given route demand Q_p from the route choice submodel (2.2.4), route specific link inflows are calculated using:

$$y_{lp} = Q_p \widehat{\alpha}_{lp} \ \forall l \in L, \ \forall p \in P_{rs}, \ \forall rs \in RS. \tag{6}$$

The route specific link inflows are used to determine turn demands T_{ij} from inlink i to outlink j used as input for the node model component by:

$$T_{ij} = \sum_{RS} \sum_{p \in P_{rs}} \sigma_{jp} y_{ip} \ \forall i \in I_n, \ \forall j \in J_n, \ \forall n \in N, \tag{7}$$

where $\sigma_{jp} \in \{0, 1\}$ indicates if route p uses link j.

2.2.2. Node model component

The node model component in SCCTA models determines which nodes in the network form an active bottleneck. Bottlenecks are activated on nodes where the demand for one or more of the 'outlinks' or turning movements ('turns') exceeds the capacity of the respective outlink(s) or turn(s) (Fig. 2). On nodes that represent an active bottleneck, the node model component also determines how the available supply on its outgoing links and turns is distributed over the competing ingoing links ('inlinks').

Any first order node model can be used, as long as it complies to a set of seven requirements[3] for first order macroscopic node models described in (Tampère et al., 2011). One of these requirements is that the node model should comply with local supply constraints, which is the very reason that SCCTA models obey strict link capacity constraints. Below, a coarse outline of the workings of such node models is sketched; we refer to (Bliemer et al., 2014; Flötteröd and Rohde, 2011; Smits et al., 2015; Tampère et al., 2011) for a more thorough description and solution algorithms for specific node models.

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, define the set of outlinks directly related to inlink i as $J_i = \{j | T_{ij} > 0\}$ and the set of inlinks directly related to outlink j as $I_j = \{i | T_{ij} \rangle 0\}$. For all $n \in N$ a node model $\Gamma_n(\cdot)$ is defined that calculates the vector of turn acceptance factors $\boldsymbol{\alpha}_n$ reducing turn flows traversing n as a function of the vector of travel demand for each turning movement on the node ($\mathbf{T}_n$), 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 either the capacity, or (in case of spillback) the outflow of the outlink . This yields:

$$\begin{aligned} \alpha_n &= \Gamma_n(\mathbf{T}_n, \mathbf{C}_n, \mathbf{R}_n) \\ \text{where}: \ \alpha_n &= \{\alpha_{ij} \ \forall \ ij \in IJ_n\}, \\ \mathbf{T}_n &= \{T_{ij} \ \forall \ ij \in IJ_n\}, \\ \mathbf{C}_n &= \{C_i \ \forall \ i \in I_n\} \text{ and} \\ \mathbf{R}_n &= \{R_j \ \forall j \in J_n\}. \end{aligned} \tag{8}$$

Note that one of the requirements from (Tampère et al., 2011) is the first-in-first-out (FiFo) assumption. It means that traffic flows out of an inlink and into different outlinks in the same order they reached the end of the inlink. In the context of an static traffic assignment model without time-varying traffic flows, this assumption causes the flow acceptance factors for all turns on an inlink of a node to be equal by definition, thereby also defining the relation between turn based and link based flow acceptance factors as $\alpha_l = \alpha_{ij} \Leftrightarrow i = l$. Further note that since we are using an SCCTA model (hence: without storage constraints), spillback cannot occur, and $R_j = C_j$, whereas in models with storage constraints, due to spillback, R_j can also be equal to the outflow of link j.

2.2.3. Fixed point problem and travel time calculation

As Fig. 1 and Sections 2.2.1 and 2.2.2 suggest, turn demands and flow acceptance factors are mutually dependent, and iterations between the node Eq. (8)) and link model (Eqs. (6) and ((7)) are required to reach a fixed point. This fixed point problem was identified by (Bliemer et al., 2014), who have proven that its solution is unique under very mild conditions, whereas (Raadsen and Bliemer, 2018) provide a more general and capable solution scheme for the problem.

Once the fixed point is reached, route travel times are calculated using:

$$c_p = \sum_{l \in L_p} \frac{L_l}{\dot{v}_l} + \tau_p, \tag{9}$$

where L_l and $\dot{v}_l$ are the length and maximum speed on link l, respectively and τ_p represents the route queuing delay. The route queueing delays are a function of all turn based flow acceptance factors on the route as derived in (Bliemer et al., 2014):

$$\tau_p = \frac{T}{2}\left(\frac{1}{\widehat{\alpha}_p} - 1\right), \tag{10}$$

where T represents the study period duration considered by the assignment model. Note that $\widehat{\alpha}_p$ represents the same variable as in Eq. (5), but subscript l was removed because in this context it is the last link of the route p by definition. Further note that, without loss of generality, delay occurring on links in free flow conditions could be added to the first term by using the speed specified by the fundamental diagram in the free flow branch instead of the maximum speed on the link.

2.2.4. Route choice submodel for SUE

Within the route choice submodel, the route choice model uses the route travel times from the network loading submodel to compute route fractions for all route alternatives between an OD pair. The SUE assignment model assumes random utility maximization with perception errors, hence a multinomial logit (MNL) model to calculate route choice probabilities:

$$\psi_{rs,p} = \exp\left(-\mu_{rs} c_p\right) \Big/ \sum_{p' \in P_{rs}} \exp\left(-\mu_{rs} c_{p'}\right) \tag{11}$$

[3] These are: 1) general applicability (not just merges and diverges), 2) maximizing flows, 3) non-negativity, 4) conservation of vehicles, 5) satisfying demand and supply constraints, 6) obeying the conservation of turning fractions, 7) satisfaction of the invariance principle)

where $\psi_{rs,p}$ denotes the probability of choosing route p for demand on OD pair rs and μ_{rs} is a scale parameter describing the degree of travelers' perception errors on route travel times (where perfect knowledge is assumed when μ_{rs} approaches infinity). Note that μ_{rs} is determined using a global scale parameter μ (which can be estimated using the variance in observed data on route choices), normalized over ODpairs by $\mu_{rs} = \mu \,/\, \min_{p \in P_{rs}} \sum_{l \in L_p} \frac{L_l}{\bar{v}_l}$. This normalization ensures that the relative effect of perception errors is the same on all OD pairs (regardless of their average route travel time). Furthermore, the SUE is approximated using route choice iterations between the network loading and route choice submodels. In each route choice iteration, new route demands are calculated using:

$$Q_p = \psi_{rs,p} D_{rs} \tag{12}$$

where Q_p denotes the demand on route p. Note that in practical applications, convergence to SUE conditions is enforced and sped up by averaging the route choice probabilities between the route choice iterations using a smart averaging scheme (in this case the self regulating average (Liu et al., 2009) is used). The way this is done in the test case applications will be described in Section 4. Further note that, without loss of generality, other discrete route choice models may be used (e.g. path size logit (Ben-Akiva and Ramming, 1998), C-logit (Cascetta et al., 1996) or paired combinatorial logit (Chu, 1989)), but this is outside the scope of this paper.

2.3. Proposed solution method

The proposed solution method solves bi-level problem (1) using first order Taylor approximated response functions to replace the SCCTA model to solve a series of simplified optimization problems. The simplified optimization problem (Section 2.3.5) includes the sensitivity of link flows (Section 2.3.1) and route queuing delays (Section 2.3.2) for changes in OD-demand, but, contrary to the methods from current practice, avoids performing costly additional assignment model runs in the lower level to determine these sensitivities. Because the sensitivities used are point approximations, link state constraints are added to prevent their use outside their valid range. These constraints are also used to include observed congestion patterns in the matrix estimation problem (Section 2.3.3).

2.3.1. Response function for observed link flows

To determine the response function for link flows we express link flow as a function of OD demand by substitution of (12) into (6) and summing over OD pairs:

$$y_l(\mathbf{D}) = \sum_{rs \in RS} \sum_{p \in P^{rs}} \widehat{\alpha}_{lp}(\mathbf{D}) \psi_{rs,p} D_{rs} \tag{13}$$

Following (Frederix et al., 2013), we use the first order Taylor approximation around the current solution $\mathbf{D}_{k-1}$ as the response function for link flows yielding:

$$\begin{aligned} y_l(\mathbf{D}) &= \sum_{rs \in RS} \sum_{p \in P_{rs}} \widehat{\alpha}_{lp}(\mathbf{D}_{k-1}) \psi_{rs,p} D_{rs} + \sum_{rs \in RS} \frac{\partial y_l(\mathbf{D}_{k-1})}{\partial D_{rs}} \left(D_{rs} - D_{k-1,rs}\right) \\ &= \sum_{rs \in RS} \sum_{p \in P_{rs}} \widehat{\alpha}_{lp}(\mathbf{D}_{k-1}) \psi_{rs,p} D_{rs} + \sum_{rs \in RS} \frac{\partial \sum_{rs \in RS} \sum_{p \in P^{rs}} \widehat{\alpha}_{lp}(\mathbf{D}) \psi_{rs,p} D_{rs}}{\partial D_{rs}} \left(D_{rs} - D_{k-1,rs}\right) \\ &= \sum_{rs \in RS} \sum_{p \in P_{rs}} \widehat{\alpha}_{lp}(\mathbf{D}_{k-1}) \psi_{rs,p} D_{rs} + \sum_{rs \in RS} \left(D_{rs} - D_{k-1,rs}\right) \left[\sum_{rs' \in RS} \sum_{p' \in P_{rs}} \psi_{rs',p'} \frac{\partial \widehat{\alpha}_{lp'}(\mathbf{D})}{\partial D_{rs}} \bigg|_{\mathbf{D}_{k-1}} D_{k-1,rs'} \right] \end{aligned} \tag{14}$$

Or, in vector-matrix form:

$$\mathbf{y}(\mathbf{D}) = \widehat{\alpha} \psi \mathbf{D}_{k-1} + \frac{\partial \widehat{\alpha}}{\partial \mathbf{D}} \psi \mathbf{D}_{k-1} (\mathbf{D} - \mathbf{D}_{k-1}) \tag{15}$$

where $\mathbf{y}(\mathbf{D})$ is the vector of link flows of size $\widetilde{L}$ x 1, $\widehat{\alpha}$ is the assignment matrix on route level (size $\widetilde{L}$ x P) determined by assigning $\mathbf{D}^{k-1}$, $\boldsymbol{\psi}$ is a matrix of route choice probabilities of size PxRS and $\partial \widehat{\alpha} / \partial \mathbf{D}$ is the sensitivity of the assignment matrix on route level (size $\widetilde{L}$ x RS x P).

2.3.2. Response function for observed queuing delays

We propose to add observed travel times on observed routes $\widetilde{p} \in \widetilde{P}$ to the optimization problem. As reflected in Eq. (9), the average travel time on a route consists of the free-flow travel time and the queueing delay. Being a constant, the free-flow component per route, optionally including any delay occurring on links in free flow conditions (Section 2.2.3), can be deducted from the observed route travel times to derive an approximated observed route queuing delay $\tau_{\widetilde{p}}$.

Through Eq. (8) $\tau_{\widetilde{p}}$ is a function of $\mathbf{T}_n$, which means that, through Eqs. (6) and (7), is it also a function of $\mathbf{D}$. This means that a response function for observed queuing delays can be included into the optimization problem. Analogue to the approach taken for link flows, the first order Taylor approximation is derived for route queuing delays as:

$$\begin{aligned}\tau_{\widetilde{p}}(\mathbf{D}) &= \frac{T}{2}\left(\frac{1}{\widehat{\alpha}_{\widetilde{p}}(\mathbf{D}_{k-1})}-1\right)+\sum_{rs\in RS}\frac{\partial\tau_{\widetilde{p}}(\mathbf{D})}{\partial D_{rs}}\left(D_{rs}-D_{k-1,rs}\right)\\ &= \frac{T}{2}\left(\frac{1}{\widehat{\alpha}_{\widetilde{p}}(\mathbf{D}_{k-1})}-1\right)-\frac{T}{2}\sum_{rs\in RS}\frac{\partial\widehat{\alpha}_{\widetilde{p}}(\mathbf{D})/\partial D_{rs}|_{\mathbf{D}_{k-1}}}{\widehat{\alpha}_{\widetilde{p}}(\mathbf{D}_{k-1})^2}\left(D_{rs}-D_{k-1,rs}\right),\end{aligned} \tag{16}$$

or, in vector-matrix form:

$$\tau(\mathbf{D}) = \frac{T}{2}\left(\frac{1}{\widetilde{\alpha}(\mathbf{D}_{k-1})}-1\right)-\frac{T}{2}(\mathbf{D}-\mathbf{D}_{k-1})^T\left(\frac{\partial\widetilde{\alpha}(\mathbf{D})}{\partial\mathbf{D}}\cdot\frac{1}{\widetilde{\alpha}^2(\mathbf{D}_{k-1})}\right) \tag{17}$$

where $\boldsymbol{\tau}(\mathbf{D})$ and $\widetilde{\alpha}$ are vectors of size (1 x $\widetilde{P}$) containing route queuing delays and flow acceptance factors on route level, respectively, and $\partial\widetilde{\alpha}(\mathbf{D})/\partial\mathbf{D}$ is a matrix of size (RS x $\widetilde{P}$) containing the sensitivity of the acceptance factors on route level. Note that $\widetilde{p}$ may be any non-cyclical combination of adjacent directed links in the network. Further note that because only the travel time of the whole route $\widetilde{p}$ is relevant, $\widetilde{\alpha}$ only contains flow acceptance factors for the last link for each $p \in \widetilde{\mathbf{P}}$ (removing the l subscript on the $\widehat{\alpha}_{\widetilde{p}}$ variable), whereas $\widehat{\alpha}$ contains flow acceptance factors for each link l within each route $p \in \mathbf{P}$. Therefore, $\widetilde{\alpha} \subseteq \widehat{\alpha}$ and $\partial\widetilde{\alpha}(\mathbf{D})/\partial\mathbf{D} \subseteq \partial\widehat{\alpha}(\mathbf{D})/\partial\mathbf{D}$, which means that calculation of the response function for queueing delay does not require any derivation of additional acceptance factors or sensitivities.

2.3.3. Link state constraints for observed congestion patterns

To include observed congestion patterns, link state constraints are used. Link state constraints enforce and preserve all links in a state known to coincide with an observed congestion patterns and are defined as:

$$\chi_j\left(\sum_{i\in I_j}T_{ij}(\mathbf{D})-\delta_j R_j\right) \le 0 \;\forall\, j \in L, \tag{18}$$

where χ_j indicates the state of link j, which is either constraining ($\chi_j = -1$) or not constraining ($\chi_j = 1$) and δ_j represents the minimum size of the deficit (when $\chi_j = -1$) or surplus (when $\chi_j = 1$) of supply at link j expressed as the ratio between demand for link j and its supply R_j. The response function for turn demands is derived by including (out)link-route incidence indicator σ_{jp} in both terms of (14) yielding:

$$\begin{aligned}T_{ij}(\mathbf{D}) &= \sum_{RS}\sum_{p\in P^{rs}}\sigma_{jp}\widehat{\alpha}_{ip}(\mathbf{D}_{k-1})\psi_{rs,p}D_{k-1,rs}\\ &+\sum_{rs\in RS}\left(D_{rs}-D_{k-1,rs}\right)\left[\sum_{rs'\in RS}\sum_{p'\in P^{rs}}\psi_{rs',p'}\sigma_{jp}\frac{\partial\widehat{\alpha}_{ip'}(\mathbf{D})}{\partial D_{rs}}\bigg|_{\mathbf{D}^{k-1}}D_{k-1,rs'}\right]\\ &\forall i\in I_n,\;\forall j\in J_n,\;\forall n\in N\end{aligned} \tag{19}$$

Contrary to observed link flows and route queuing delays, congestion patterns are not included as an objective function component but as linear constraints to the optimization problem. The reason for this is that the strict capacity constraints in the node model cause discontinuities in $\boldsymbol{\alpha}_n(\mathbf{T}_n)$ whenever a change in $\mathbf{T}_n$ causes an outlink from node n to switch from unconstrained to supply constrained or vice versa. This is illustrated in Appendix B using a numerical example. When such a link state switch would occur during matrix estimation, an update of $\widehat{\alpha}_{lp}$ for all routes using this bottleneck would be necessary, as all downstream count locations change from sensitive to insensitive or vice versa. Furthermore, all (gradient approximation) calculations done so far with respect to these routes would become useless, since they are no longer valid after the state-change of the (potential) bottleneck. Simply updating $\widehat{\alpha}_{lp}$ for all p using l after a link state switch would practically mean starting over the matrix estimation process with an altered prior matrix, causing unnecessary bias from the original prior matrix, wasted calculation time and probably non-convergence of the bi-level optimization problem.

The issue described above is present in all matrix estimation methods using an assignment model with strict capacity constraints. It has been described before in the context of matrix estimation using DTA models by Frederix, 2012 who referred to it as "*Non-convexity* [of the upper level objective function] *due to congestion dynamics*". Frederix suggests that any transitions between traffic regimes during matrix estimation should be avoided, meaning that the link states for all potential bottleneck links should be consistent with the start solution (i.e.: the link states from the assigned prior demand matrix) and that this state should be maintained during matrix estimation. These suggestions are operationalized in the proposed solution method by addition of link state constraints (18) to the simplified optimization problem.

To specify $\boldsymbol{\chi} = \{\chi_j \forall j \in L\}$ the link states from assignment of the prior OD matrix could be used when this congestion pattern sufficiently corresponds to the observed congestion pattern or when no observed congestion patterns are available. Alternatively, $\boldsymbol{\chi}$ could be derived by determining the regime of all $j \in L$ by comparing observed link speeds (from e.g. floating car data or loop detectors) with critical link speeds from the fundamental diagram. When the observed speed is lower than the critical speed, the link is in congested state, otherwise the link is in free flow state. Then, set $\chi_j = -1$ on links that are in free flow state and have one or more inlinks that is in congested state and $\chi_j = 1$ for all other links. In case of diverges with one or more congested inlinks and more than one uncongested outlinks, additional data or knowledge is needed to determine which of the outlink(s) is actively constraining the inlink(s). Note that

link state information from floating car data (on observed links) and from prior demand assignment results (on unobserved links) may be combined, hence the proposed solution method does not require observed link state information for all links in the network.

The minimum capacity surpluses (on non-constraining outlinks $\delta_j < 1 \forall \{j \in L | \chi_j = 1\}$) and deficits (on constraining outlinks $\delta_j > 1 \forall \{j \in L | \chi_j = -1\}$), act as a buffer around the discontinuity in $\boldsymbol{\alpha}_n(\mathbf{T}_n)$ and should be set to a value as close to one as possible, but sufficiently far away from one to prevent unintentional regime switches when running the lower level.

Note that there are three reasons to include link state information as constraints instead of objective function components. Firstly, constraints guarantee that transitions between traffic regimes during matrix estimation can indeed not occur. Secondly, since a link state is a binary variable, it is more natural to include it as a constraint. Thirdly, under the hood, any analytical gradient based solver will use some sort of barrier function to penalize constraint violations, which is effectively the same as including it in the objective function, but only now the solver (instead of the user) determines sufficiently large weight values.

Further note that the capacity deficits $\delta_j > 1$ may also be used to include observed capacity deficits (but then as a lower bound) derived from the prior assignment, by setting:

$$\delta_j = \frac{\sum_{i \in I_j} T_{ij}(\mathbf{D}^0)}{R_j} \; \forall \; \{j \in L \mid \chi_j = -1\}, \tag{20}$$

or alternatively by setting to values derived from observed queue lengths in front of the link using a simple point queue model (Brederode et al., 2017). Note that the observed queuing delays from routes in $\widetilde{P}$ represent the observed size of capacity deficits on the links it traverses. Therefore, to prevent overspecification (and the risk of infeasibility) of the optimization problem this may only be done for constraining outlinks that are not traversed by routes in $\widetilde{P}$.

2.3.4. Normalization of weights

Introduced in 2.1, weighing parameters w_1, w_2 and w_3 are used to define the relative importance of the three objective function components f_1 (prior OD demand), f_2 (link flows) and f_3 (route queuing delays). Typically, these weights are set proportional to the relative level of confidence associated with the three types of observed data. However, since these types of data have a different scale (a summation of OD demand differences versus a summation of link flow differences versus a summation of route queuing delay differences) 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. Note that we choose to separate normalization and weighting for sake of tractability; alternatively, one could combine the normalization and weighting into a single weight value for each component.

To normalize different objective function components a method described in e.g. (Alpcan, 2013) is used to determine 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 normalization within the objective function.

In our case, the value of the Utopia points $f_{1,U}, f_{2,U}$ and $f_{3,U}$ are all zero, occurring when $\mathbf{D} = \mathbf{D}_0$, $\mathbf{y}(\mathbf{D}) = \widetilde{\mathbf{y}}$ and $\tau(\mathbf{D}) = \widetilde{\tau}$, respectively. Nadir point $f_{1,N}$ is defined as the summation of quadratic differences between either the prior demand and zero or the prior demand and its upper bound. Nadir point $f_{2,N}$ is defined as the summation of quadratic differences between either the observed link flow and zero or the observed link flow and the links capacity. Nadir point $f_{3,N}$ is approximated by the summation of quadratic differences between either the observed route queuing delays and zero or the observed route queuing delays and route queuing delays from an assignment of the upper bound on OD demand. For all three Nadir points, for each element, the largest quadratic difference is chosen. Eqs. (21) summarize the Nadir point definitions described above.

$$\begin{aligned} f_1^N &= \sum_{rs \in RS} \max\left[D_{rs,0}^2, \left(\overline{D}_{rs} - D_{rs,0}\right)^2\right] \\ f_2^N &= \sum_{l \in L} \max\left[\widetilde{y}_l^2, \left(C_l - \widetilde{y}_l\right)^2\right] \\ f_3^N &= \sum_{p \in \widetilde{P}} \max\left[\widetilde{\tau}_p^2, \left(\widetilde{\tau}_p - \tau_p(\overline{\mathbf{D}})\right)^2\right] \end{aligned} \tag{21}$$

After (arbitrary) normalization of f_2 and f_3 to f_1 the objective function in (1) reads:

$$\mathbf{D}^* = \underset{\mathbf{D}}{\operatorname{argmin}}\left(w_1 \sum (\mathbf{D} - \mathbf{D}_0)^2 + w_2 f_{2,N} \Big/ f_{1,N} \sum (\mathbf{y}(\mathbf{D}) - \widetilde{\mathbf{y}})^2 + w_3 f_{3,N} \Big/ f_{1,N} \sum (\tau(\mathbf{D}) - \widetilde{\tau})^2\right) \tag{22}$$

2.3.5. Simplified optimization problem

We aim to solve the bi-level problem defined in Eq. (1) by use of objective function (22) and the response functions for observed link flows and observed route queuing delays defined in Eqs. (15) and (17), respectively. To avoid transitions between traffic regimes during estimation link state constraints are added in the form of inequality (18) on top of the non-negativity constraints. Furthermore, upper bounds on the OD demands $\overline{\mathbf{D}}$ are added which can be used to reduce the number of potential constraining links from L to $J_{\overline{\mathbf{D}}}$, thereby decreasing the calculation time of the solution method. Upper bounds $\overline{\mathbf{D}}$ are typically related to $\mathbf{D}_0$ reflecting a maximum (absolute or relative) allowed increase per cell. This yields the following simplified optimization problem:

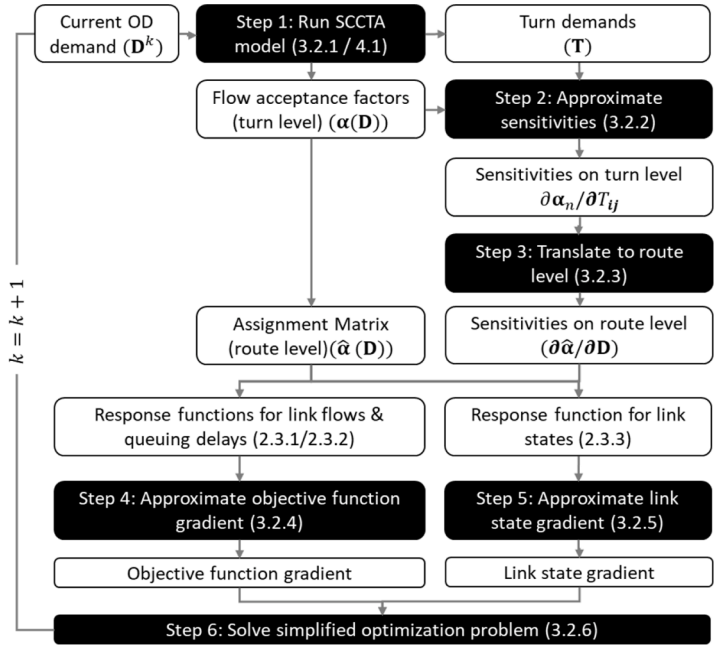


Fig. 3. Overview of proposed solution approach.

$$
\begin{aligned}
\mathbf{D}^* = \underset{\mathbf{D}}{\operatorname{argmin}} \Big(& w_1 \sum (\mathbf{D} - \mathbf{D}_0)^2 + w_2 f_{2,N} \Big/ f_{1,N} \sum (\mathbf{y}(\mathbf{D}) - \widetilde{\mathbf{y}})^2 + w_3 f_{3,N} \Big/ f_{1,N} \sum (\tau(\mathbf{D}) - \widetilde{\tau})^2 \Big) \\
\text{Subjectto} : \ & \mathbf{y}(\mathbf{D}) = \widehat{\alpha} \psi \mathbf{D}_{k-1} + \frac{\partial \widehat{\alpha}}{\partial \mathbf{D}} \psi \mathbf{D}_{k-1} (\mathbf{D} - \mathbf{D}_{k-1}) \\
& \tau(\mathbf{D}) = \frac{T}{2} \left(\frac{1}{\widetilde{\alpha}(\mathbf{D}_{k-1})} - 1 \right) - \frac{T}{2} (\mathbf{D} - \mathbf{D}_{k-1})^T \left(\frac{\partial \widetilde{\alpha}(\mathbf{D})}{\partial \mathbf{D}} \cdot \frac{1}{\widetilde{\alpha}^2(\mathbf{D}_{k-1})} \right) \\
& 0 \le \mathbf{D} \le \overline{\mathbf{D}} \\
& \chi_j \left(\sum_{i \in I_j} T_{ij}(\mathbf{D}) - \delta_j R_j \right) \le 0 \ \forall j \in J_{\overline{\mathbf{D}}}
\end{aligned}
\tag{23}
$$

Optimization problem (23) is a simplified version of the true optimization problem (1), because the link flows and link demands are approximated instead of determined by the assignment model, because the link state constraints restrict the solution space in order to be able to safely use the approximated variables, and because the vector of route fractions $\boldsymbol{\psi}$ is assumed constant. The simplified optimization problem has a quadratic objective function, linear inequality constraints and is typically very large (given the number of elements in $\mathbf{D}$ in real world transport models).

3. Solution algorithm

This section describes the proposed solution algorithm. Section 3.1 provides an overview, whereas Sections 3.2 and 3.3 describe the algorithm details and some of its mathematical properties, respectively.

3.1. Overview

The proposed solution algorithm is summarized in Fig. 3. Each iteration consists of six steps to solve (an updated version of) the simplified optimization problem. Within an iteration, only a single SCCTA model assignment is run to determine the assignment matrix (Section 3.2.1). Then, only for turns traversing an active bottleneck node, the local sensitivity of its bottleneck flow acceptance factor to local turn demand is approximated using finite differences, requiring one additional run of only the node model component (Section 3.2.2). The resulting local sensitivities are used to construct the approximated sensitivity of the assignment matrix, as described in Section 3.2.3. Furthermore, the approximated sensitivity of the assignment matrix is used to approximate gradients of the link flow and route section delay components within the objective function (Section 3.2.4) and its linear constraints (3.2.5), which are used to efficiently solve the simplified optimization problem (Section 3.2.6). Using locally approximated sensitivities, the computational cost of the proposed method is negligible compared to methods using full assignment runs to determine sensitivities.

The proposed solution method is fast, scalable, robust, tractable and reliable because conditions under which a solution to the

simplified optimization problem exists are known and because the problem is convex and has a smooth objective function. These favorable mathematical properties are discussed in Sections 3.3.1, 3.3.2 and 3.3.3, respectively as they played an important role during the development of the solution method and the implementation of the solution scheme.

3.2. Solution scheme

Using knowledge about the SCCTA model (2.2) and the simplified optimization problem (2.3.5), in this section the solution scheme is presented. It consists of six steps, each of which is executed in each iteration. The sixth step includes a stop criterion to determine whether the solution to the true optimization problem has been found or if an additional iteration is required.

3.2.1. Step 1: run SCCTA model and derivation of the assignment matrix

As a reference, we first describe how the assignment matrix relates to assignment model results for SCRTA class models before we describe this relationship is the case of SCCTA models. Because SCRTA class models lack strict capacity constraints, all traffic arrives at its destination within the study period. Therefore, an element in the assignment matrix from SCRTA class models merely describes the proportion of demand on an OD pair that has chosen a route using the considered observed link and can be derived from the route choice probabilities calculated by the route choice submodel only using:

$$A_{rs,l} = \sum_{p \in P_l} \psi_{rs,p}, \tag{24}$$

where $A_{rs,l}$ denotes the entry in the assignment matrix for observed link l and OD pair rs and P_l denotes the set of routes using link l.

When using a SCCTA class model, entries in the assignment matrix are reduced by the proportion of OD flow being held up by capacity constraints on links upstream from the considered link as calculated by the network loading submodel, yielding:

$$A_{rs,l} = \sum_{p \in P_l} \psi_{rs,p} \widehat{\alpha}_{lp}, \tag{25}$$

where the route-level assignment matrix $\widehat{\alpha}_{lp}$ is calculated using (5). Note that, because within each iteration constant route probabilities are assumed (Section 2.2), elements in the route based assignment matrix ($\widehat{\alpha}_{lp}$) are the driving variables in the lower level. They are used to approximate link flows (14), queuing delays (16) and turn demands (19) and to approximate the gradients of the objective function (Section 3.2.4) and the link state constraints (Section 3.2.5). Therefore, in the remainder of this paper we do show how $A_{rs,l}$ and its sensitivities are derived, but in the solution algorithm, its two components $\psi_{rs,p}$ and $\widehat{\alpha}_{lp}$ are used separately.

3.2.2. Step 2: approximate sensitivities on turn level

Analogue to the assignment matrix itself (Section 3.2.1), the sensitivity of the assignment matrix (to be captured in $\partial\widehat{\alpha}\ /\partial\mathbf{D}$) is constructed from the sensitivities of the acceptance factors on turn level.

Using the node model, a point derivative of $\alpha_{ij}$to any T_{ij} can be approximated using finite differences. Only for turns that are actively constrained by an outlink or turn (i.e. $\alpha_{ij} < 1$), the local sensitivity of its flow acceptance factor to local turn demand needs to be approximated. This is being done using the (one sided) finite difference method around the solution obtained by the (single) full assignment run in the lower level:

$$\frac{\partial \alpha_n}{\partial T_{ij}^-} = \frac{\alpha_n^* - \Gamma_n\left(\mathbf{T}_n^-, \mathbf{C}_n, \mathbf{R}_n\right)}{\varepsilon} \ \forall \left\{ ij \in IJ \middle| \alpha_{ij}^* < 1 \right\}, \tag{26}$$

where α_n^* and $\mathbf{T}_n^*$ are the vectors of turn flow acceptance factors and turn demands from the solution calculated by the full assignment, ε is the step size used for the finite difference calculation, $\mathbf{T}_n^- = (\mathbf{T}_n^* \backslash \{T_{ij}^*\}) \cup \{T_{ij}^* - \varepsilon\}$, the set of turn demands where the turn demand for the considered turn ij is lowered by ε for finite differences. This requires only one additional application of a node model for each actively constrained turn. These point derivatives are then used as an approximate of $\partial\alpha_{ij}/\partial T_{ij'}$ in the upper level. Note that these point derivatives approximate the function well if no discontinuity occurs. This is illustrated in the example from Appendix B.

Note that by approximating derivatives we determine the partial derivatives for all turns on the network, but we choose to omit approximating secondary interaction effects. This means that we omit the fact that when simultaneously changing multiple elements in $\mathbf{D}$, the effect on the set of route based flow acceptance factors $\widehat{\alpha}_{lp}$ and therefore the effect on the assignment matrix $\mathbf{A}$ might not be simply the sum of the effects of changing D_{rs} sequentially per OD pair. Furthermore, note that although the proposed method would also work for nodes on which constraints imposed by geometry of the node itself exist, for the sake of simplicity, in this paper we assume these so called internal node constraints to be non-existent.

3.2.3. Step 3: translate sensitivities to route level

The sensitivities on turn level (Section 3.2.2) and the link states constraints (Section 2.3.3) allow for calculation of the sensitivity of the assignment matrix on route level ($\partial\widehat{\alpha}/\partial\mathbf{D}$ in (23)). This is a three-dimensional matrix containing elements $\partial\widehat{\alpha}_{lp}/\partial D_{rs'}$ that describe the sensitivity of link l used by route p for changes in demand on OD-pair rs'. We calculate these elements as follows.

First, we determine $\partial\widehat{\alpha}_{lp}/\partial Q_{p'}$ the sensitivity of link l on route p for changes in demand on route p', by taking the derivative of Eq. (5)

to **Q** (using the product rule), yielding:

$$\frac{\partial \widehat{\alpha}_{lp}(\mathbf{Q})}{\partial Q_{p'}} = \left(\prod_{ij' \in IJ_{p,il}} \alpha_{ij'}(\mathbf{Q}) \right) \left(\sum_{ij' \in IJ_{p,il}} \left(\frac{\partial \alpha_{ij'}}{\partial Q_{p'}}\Big|_{\mathbf{Q}} \Big/ \alpha_{ij'}(\mathbf{Q}) \right) \right) \tag{27}$$

To simplify Eq. (27), we first define the following variables. Let ij_p^* be the first blocked turning movement on route p (i.e.: $\alpha_{ij_p^*} < 1$). Let $\overline{IJ_p^*}$ be the set of turns on route p located upstream from ij_p^* and $\underline{IJ_p^*}$ be the set of turns on route p downstream from turn ij_p^* until and including turn il. Furthermore let $T_{ij''_{p'}}$ be demand on turn $ij''_{p'}$ on route p' that influences $\alpha_{ij_p^*}$ (and hence must located on the same node as ij_p^*). Note that $\alpha_{ij_p^*}$ may be influenced by routes using the turn itself (i.e.: $ij''_{p'} = ij_p^*$), but it can also be influenced by routes on other turns sharing their outlink with one of the turns that share their inlink with turn ij_p^* (in which case $ij''_{p'} \neq ij_p^*$). Then, given that link state constraints will maintain to be satisfied, three properties of Eq. (27), all related to the strict capacity constraints in the assignment model, are considered:

1. On turns ij' located upstream from the first blocking turn on route p, by definition, all demand passes, hence $\alpha_{ij'} = 1 \; \forall \; ij' \in \overline{IJ_p^*}$ and $\partial \alpha_{ij'} / \partial Q_{p'} = 0 \; \forall \; ij' \in \overline{IJ_p^*}$;
2. On turns ij' located downstream from the first blocking turn on route p, due to the strict capacity constraints, the acceptance factor on the first blocking turn ij_p^* will neutralize any changes in demand on p, such that its downstream turns become insensitive: $\partial \alpha_{ij'} / \partial Q_{p'} = 0 \; \forall \; ij' \in \underline{IJ_p^*}$.

Incorporating these two properties into Eq. (27) yields:

$$\begin{aligned} \frac{\partial \widehat{\alpha}_{lp}(\mathbf{Q})}{\partial Q_{p'}} &= \left(\prod_{ij' \in \left\{ ij_p^*;\, \underline{IJ_p^*} \right\}} \alpha_{ij'}(\mathbf{Q}) \right) \frac{\partial \alpha_{ij_p^*}}{\partial Q_{p'}}\Big|_{\mathbf{Q}} \Big/ \alpha_{ij_p^*}(\mathbf{Q}) \\ &= \left(\prod_{ij' \in \underline{IJ_p^*}} \alpha_{ij'}(\mathbf{Q}) \right) \frac{\partial \alpha_{ij_p^*}}{\partial Q_{p'}}\Big|_{\mathbf{Q}} \end{aligned} \tag{28}$$

To use the approximated point derivatives from 3.2.2, the third property is considered:

1. Analogue to the second property, when turn $ij''_{p'}$ on route p' (the turn for which its demand is influencing $\alpha_{ij_p^*}$) is located downstream from the first blocking turn on route p', acceptance factor $\alpha_{ij_p^*}$ will neutralize any changes in demand on p', such that $\frac{\partial \alpha_{ij'}}{\partial Q_{p'}} = 0 \; \forall \left\{ p' : ij''_{p'} \in \underline{IJ_{p'}^*} \right\}$. In other cases, $Q_{p'} = T_{ij''_{p'}}$ and thus $\partial \alpha_{ij'} / \partial Q_{p'} = \partial \alpha_{ij'} / \partial T_{ij''_{p'}} \; \forall \; \{ p' : ij''_{p'} \in \{ ij_{p'}^*;\, \overline{IJ_{p'}^*} \} \}$.

Incorporating the third property into Eq. (28) yields:

$$\frac{\partial \widehat{\alpha}_{lp}(\mathbf{Q})}{\partial Q_{p'}} = \begin{cases} \left(\prod_{ij' \in \underline{IJ_p^*}} \alpha_{ij'}(\mathbf{Q}) \right) \frac{\partial \alpha_{ij'}}{\partial T_{ij''_{p'}}}\Big|_{Q} & \forall \left\{ p' : ij''_{p'} \in \left\{ ij_{p'}^*;\, \overline{IJ_{p'}^*} \right\} \right\} \\ 0 & \forall \left\{ p' : ij''_{p'} \in \underline{IJ_{p'}^*} \right\} \end{cases} \tag{29}$$

Example network

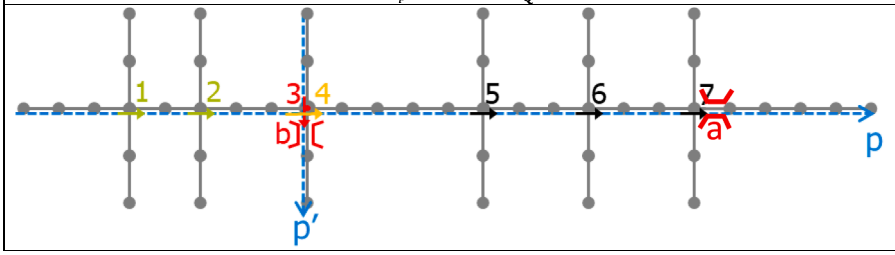
To further clarify these variables and properties, the figure below shows an example network (nodes and links in gray) with routes p and p' indicated by the blue dashed lines. Small arrows indicate turning movements used by the routes, each turning movement is labeled by a colored number. There are bottlenecks a and b causing turn 7 on route p and turn 3 on route p' to be blocking. Due to FiFo on the upstream node of the bottleneck link, this also causes turn 4 on route p to be blocking, thereby defining $ij_p^* = 4$, $\overline{IJ_p^*} = \{1,2\}$, $\underline{IJ_p^*} = \{5,6,7\}$ and $ij'' = 3$. Considering route p, the first two properties indicate that only the sensitivity of the alpha on turn $ij_p^* = 4$ is relevant, whereas the third property indicates that demand on route p' is represented by turn demand on turn $ij'' = 3$. This means that:

- for links upstream from turn 3: $\frac{\partial \widehat{\alpha}_{lp}(\mathbf{Q})}{\partial Q_{p'}} = 0$;
- for links between turns 4 and 7: $\frac{\partial \widehat{\alpha}_{lp}(\mathbf{Q})}{\partial Q_{p'}} = \frac{\partial \alpha_4}{\partial T_3}|_{\mathbf{Q}}$, and;
- for links downstream from turn 7 $\frac{\partial \widehat{\alpha}_{lp}(\mathbf{Q})}{\partial Q_{p'}} = \alpha_7 \frac{\partial \alpha_4}{\partial T_3}|_{\mathbf{Q}}$.

Example network

To further clarify these variables and properties, the figure below shows an example network (nodes and links in grey) with routes p and p' indicated by the blue dashed lines. Small arrows indicate turning movements used by the routes, each turning movement is labeled by a colored number. There are bottlenecks a and b causing turn 7 on route p and turn 3 on route p' to be blocking. Due to FiFo on the upstream node of the bottleneck link, this also causes turn 4 on route p to be blocking, thereby defining $ij_p^* = 4$, $\overline{IJ_p^*} = \{1,2\}$, $\underline{IJ_p^*} = \{5,6,7\}$ *and* $ij'' = 3$. *Considering route p, the first two properties indicate that only the sensitivity of the alpha on turn* $ij_p^* = 4$ *is relevant, whereas the third property indicates that demand on route p' is represented by turn demand on turn* $ij'' = 3$. *This means that:*

- *for links upstream from turn 3:* $\frac{\partial \hat{\alpha}_{lp}(\mathbf{Q})}{\partial Q_{p'}} = 0$;
- *for links between turns 4 and 7:* $\frac{\partial \hat{\alpha}_{lp}(\mathbf{Q})}{\partial Q_{p'}} = \frac{\partial \alpha_4}{\partial T_3}\Big|_{\mathbf{Q}}$, *and;*
- *for links downstream from turn 7* $\frac{\partial \hat{\alpha}_{lp}(\mathbf{Q})}{\partial Q_{p'}} = \alpha_7 \frac{\partial \alpha_4}{\partial T_3}\Big|_{\mathbf{Q}}$.



Eq. (29) expresses Eq. (27) in terms of turn based acceptance factors that are output from the SCCTA model and a single partial derivative that can be derived using finite differences of its node model component. Interpretation of Eq. (29) shows that its second term represents the maximum sensitivity of route flows on p for demand on route p' whereas the first term propagates (and dampens) this sensitivity downstream from turn ij_p^* to turn ij on route p. The derivatives with respect to OD- (instead of route-) demand are defined as:

$$\frac{\partial \hat{\alpha}_{lp}}{\partial D_{rs}} = \sum_{p' \in P_{rs,l}} \psi_{rs,p'} \frac{\partial \hat{\alpha}_{lp}}{\partial Q_{p'}}. \tag{30}$$

and since route choice probabilities are fixed within a single upper level evaluation ($\partial \psi_{rs,p}/\partial D_{rs} = 0 \forall rs \in RS$), the sensitivity of an element in the assignment matrix can now be expressed as:

$$\frac{\partial A_{rs,l}}{\partial D_{rs'}} = \sum_{p \in P_{rs,l}} \psi_{rs,p} \frac{\partial \hat{\alpha}_{lp}}{\partial D_{rs'}}. \tag{31}$$

The sensitivities in (30) are used to approximate link flows (14), queuing delays (16) and turn demands (19) in the upper level using while the solver is evaluating a candidate vector of OD demands **D** and to approximate the gradients of the objective function (Section 3.2.4) and the link state constraints (Section 3.2.5).

3.2.4. *Step 4: approximate objective function gradient*

Gradients can be derived for all three components (f_1, f_2 and f_3) of the objective function. Note that by doing so, the gradient of the total objective function (2) is also determined. The partial derivatives of the first part of objective function to OD-demands are given by:

$$\frac{\partial f_1}{\partial D_{rs}} = \frac{\partial}{\partial D_{rs}} \left(\sum_{rs \in RS} \left(D_{rs} - D'_{rs} \right)^2 \right) = 2\left(D_{rs} - D'_{rs} \right), \ \forall\, rs \in RS. \tag{32}$$

The partial derivatives of the second part of the objective function f_2 to OD-demands **D** are derived using the approximated sensitivity of the assignment matrix from Section 3.2.3. First the gradient of f_2 is translated from OD to route level:

$$\frac{\partial f_2}{\partial D_{rs}} = \sum_{p \in P_{rs}} \psi_{rs,p} \frac{\partial f_2}{\partial Q_p} = \sum_{p \in P_{rs}} \psi_{rs,p} \sum_{l \in \tilde{L}} 2(y_l - \tilde{y}_l) \frac{\partial y_l}{\partial Q_p}. \tag{33}$$

To derive $\partial y_l/\partial Q_p$, first Eq. (13) for link flows is expressed on route level:

$$y_l(\mathbf{Q}) = \sum_{p\in P_l} \widehat{\alpha}_{lp}(\mathbf{Q}_{k-1})Q_p + \sum_{p'\in P}(Q_{p'} - Q_{k-1,p'}) \left[\sum_{p\in P_l} \frac{\partial \widehat{\alpha}_{lp}}{\partial Q_{p'}}\bigg|_{\mathbf{Q}_{k-1}} Q_{k-1,p}\right] \tag{34}$$

Taking the derivative to Q_p yields:

$$\begin{aligned}\frac{\partial y_l}{\partial Q_p} &= \frac{\partial}{\partial Q_p}\left(\sum_{p\in P_l} \widehat{\alpha}_{lp}Q_p + \sum_{p'\in P}(Q_{p'} - Q_{k-1,p'}) \left[\sum_{p\in P_l} \frac{\partial \widehat{\alpha}_{lp}}{\partial Q_{p'}}\bigg|_{\mathbf{Q}^{k-1}} Q_{k-1,p'}\right]\right) \\ &= \widehat{\alpha}_{lp} + \sum_{p'\in P} \frac{\partial \widehat{\alpha}_{lp}}{\partial Q_{p'}}\bigg|_{\mathbf{Q}_{k-1}} Q_{k-1,p'}, \quad \forall l \in L, \ \forall p \in P,\end{aligned} \tag{35}$$

where elements in the assignment matrix $\widehat{\alpha}_p^l$ and their sensitivities are calculated using Eqs. (5) and (29), respectively.

Finally, the partial derivatives of the third part of the objective function f_3 to OD-demands $\mathbf{D}$ are given by:

$$\begin{aligned}\frac{\partial f_3}{\partial D_{rs}} &= \frac{\partial}{\partial D_{rs}}\left(\sum_{\widetilde{p}\in\widetilde{P}} (\widetilde{\tau}_{\widetilde{p}}(\mathbf{D}) - \widetilde{\tau}_{\widetilde{p}})^2\right) = \sum_{\widetilde{p}\in\widetilde{P}} 2\left(\widetilde{\tau}_{\widetilde{p}}(\mathbf{D}) - \widetilde{\tau}_{\widetilde{p}} \frac{\partial \widetilde{\tau}_{\widetilde{p}}(\mathbf{D})}{\partial D_{rs}}\right) \\ = \sum_{\widetilde{p}\in\widetilde{P}} 2(\widetilde{\tau}_{\widetilde{p}}(\mathbf{D}) - \widetilde{\tau}_{\widetilde{p}})\left(-\frac{T}{2}\frac{\partial \widehat{\alpha}_{\widetilde{p}}(\mathbf{D})/\partial D_{rs}}{\widehat{\alpha}_{\widetilde{p}}(\mathbf{D})^2}\right) &= \sum_{\widetilde{p}\in\widetilde{P}} (\widetilde{\tau}_{\widetilde{p}} - \widetilde{\tau}_{\widetilde{p}}(\mathbf{D}))\left(\frac{T\partial \widehat{\alpha}_{\widetilde{p}}(\mathbf{D})/\partial D_{rs}}{\widehat{\alpha}_{\widetilde{p}}(\mathbf{D})^2}\right)\end{aligned} \tag{36}$$

This section has shown that the gradient of the objective function can be approximated using information from a single assignment model evaluation (Section 3.2.1) and a single additional node model evaluation for each turn (3.2.2) only.

3.2.5. Step 5: approximate link state gradient

The gradient of the link state constraints is defined as the derivatives of Eq. (18) to OD demand:

$$\frac{\partial}{\partial \mathbf{D}}\left[\chi_j\left(\sum_{i\in I_j} T_{ij} - \delta_j R_j\right)\right] = \chi_j \sum_{i\in I_j} \frac{\partial T_{ij}}{\partial D_{rs}} \ \forall j \in L, \ \forall \ rs \in RS \tag{37}$$

To calculate $\partial T_{ij}/\partial D_{rs}$, we can use the same approach used for calculation of the gradient of the link flow part of the objective function (Eqs. (34)—(35)), once we have established the relationship between link flow and turn demands. To do so we point out that in Eq. (7), the turn demand is expressed in terms of all route flows on the turns **in**link (y_{ip}), directed towards the considered outlink j (σ_{jp}), thereby excluding the acceptance factor on turn ij itself. This is shown when Eq. (6) is substituted in Eq. (7), yielding:

$$\begin{aligned}T_{ij} &= \sum_{RS}\sum_{p\in P^{rs}} \sigma_{jp}\sigma_{ip}Q_p \prod_{ij'\in IJ_p^{ij}\setminus\{ij\}} \alpha_{ij'} \\ &= \sum_{RS}\sum_{p\in P_{rs}} \sigma_{jp}\sigma_{ip}Q_p \prod_{ij'\in IJ_{p,ij}} \alpha_{ij'} \Big/ \alpha_{ij} \\ &= \sum_{RS}\sum_{p\in P_{rs}} \sigma_{jp}y_{lp} \Big/ \alpha_{ij}.\end{aligned} \tag{38}$$

Realizing that turn demands are related to link flows through Eq. (38), approximations for $\partial \widehat{\alpha}_{ip}/\partial Q_{p'} \forall i \in I_n, \ \forall n \in N$ can be derived by replacing the superscript l with iin Eq. (29) and remove any routes p' for which $ij'' = ij_p^*$ yielding:

$$\frac{\partial \widehat{\alpha}_{ip}(Q)}{\partial Q_{p'}} = \left(\prod\nolimits_{ij'\in \underline{IJ}_p^*} \alpha_{ij'}(Q)\right) \frac{\partial \alpha_{ij_p^*}}{\partial T_{ij''}}\bigg|_{\mathbf{Q}_{k-1}} \ \forall p' \exists \ ij'' \in \overline{IJ_{p'}^*} \tag{39}$$

which is the turn-demand equivalent of Eq. (29). These derivatives can be used to calculate

$$\frac{\partial T_{ij}}{\partial Q_p} = \sigma_{jp}\widehat{\alpha}_{ip} + \sum_{p'\in P} \frac{\partial \widehat{\alpha}_{ip}}{\partial Q_{p'}}|_{\mathbf{Q}_{k-1}} Q_{k-1,p}, \ \forall j \in L, \ \forall p \in P, \tag{40}$$

which is the turn-demand equivalent of Eq. (35) to be translated to OD level using:

$$\frac{\partial T_{ij}}{\partial D_{rs}} = \sum_{p\in P_{rs}} \psi_{rs,p} \frac{\partial T_{ij}}{\partial Q_p} \tag{41}$$

which is multiplied by its corresponding χ_j to yield the gradient of the link state constraint. Analogue to the approximation of the gradient of the objective function (Section 3.2.4), this section has shown that the gradient of the link state constraints can be approximated using information from a single assignment model evaluation (Section 3.2.1) and a single additional node model evaluation for each turn traversing an active bottleneck only (Section 3.2.2).

3.2.6. Step 6: solve simplified optimization problem

The simplified optimization problem can be solved by applying any solver that can handle such a problem. In this paper, the interior point algorithm described in (Waltz et al., 2006) is used in combination with the approximated gradients from Sections 3.2.4 and 3.2.5. Once solved, the estimated OD matrix is assigned using the SCCTA model as a step 1 of the next iteration, and the objective function value is evaluated and compared to a user defined threshold value for convergence.

To have a meaningful and comparable convergence criterion, the convergence threshold is defined in terms of average link flow and route delay deviations (Eq. (42)).[4] A run is considered converged when the solver has found a feasible solution and both conditions are met.

$$\frac{\sum_{a\in\widetilde{\mathbf{A}}}|y_a-\widetilde{y}_a|/\widetilde{y}_a}{|\widetilde{\mathbf{A}}|}\leq\varepsilon_{\mathbf{A}}\ \wedge\ \frac{\sum_{p\in\widetilde{\mathbf{P}}}|\tau_p-\widetilde{\tau}_p|\big/\widetilde{\tau}_p}{|\widetilde{\mathbf{P}}|}\leq\varepsilon_{\mathbf{P}} \tag{42}$$

For non-converging runs, criteria on objective function stability (absolute difference between the true objective function value of latest and previous iteration) and the maximum number of iterations are added.

If either the convergence or stability criterion is met, or when the maximum number of iterations is reached, the solution is accepted, and the algorithm stops. In other cases, the algorithm starts a new iteration by using the assignment results from the new OD matrix that was already assigned for objective function evaluation. Note that the difference between the objective function value of the simplified problem (known after step 6) and the objective function value after assignment (known after the assignment in step 1) can be used as an indicator for the size of the errors due to the first order Taylor approximations and the neglection of secondary interaction effects (as defined in Section 3.2.2).

As described in 2.3.3, for the solution algorithm to converge, links states should be consistent with the start solution (the prior demand matrix) and these states should be maintained during matrix estimation. Link state constraints (18) take care of maintaining link states, but do not enforce the start solution to be consistent. To prevent the sensitivity information in the first iteration to be based on a (possibly) inconsistent OD matrix and let the algorithm to take off on a false start, an (optional) nudging iteration is prepended to the solution scheme. In this nudging iteration, the same interior point algorithm is used to solve only the feasibility problem from the link state constraints by setting the objective function to a value of zero.

3.3. Mathematical properties of the simplified optimization problem

3.3.1. Feasibility

When the conditions under which a solution to the problem exists are known, the input can be adapted to satisfy these conditions. By doing so, the solver is guaranteed to find a feasible solution, thereby contributing to the reliability of the solution method. For the simplified optimization problem (23), feasibility is guaranteed when the non-negativity and link state constraints are satisfied by (assignment of) the prior OD matrix. This means that the prior OD matrix may not contain negatives, and that the link state constraints must be satisfied in the assignment results of the prior OD matrix. In all subsequent iterations, feasibility will be automatically maintained through the constraints themselves.

This means that feasibility can be guaranteed by adding a check on negatives in de prior OD matrix to prevent violation of the non-negativity constraints and to set the values for γ_j according to the assignment results of the prior OD matrix itself to prevent violation of the link state constraints. Alternatively, when using γ_j values from an exogenous source (i.e. observed congestion patterns), the simplified optimization problem (23), but with removed objective function, can be solved to nudge the prior OD matrix into the feasible region. Solving this problem is computationally very cheap as it is an instance of the “first phase problem” in the two phase simplex method (Murty, 1991, pp 60).

3.3.2. Convexity

Problems that are convex are likely to be solved using polynomial time algorithms which are relatively fast and scalable. Furthermore, any solution to a convex problem is a global minimum and when the problem is strictly convex this global minimum is unique, contributing to robustness and tractability of the solution method.

In Appendix A it is proven that the first part of the objective function of the simplified optimization problem (23) is strictly convex, whereas the second and third part of the objective function are convex. Furthermore, all considered constraints are linear inequalities, and as such form a closed convex set which means that (23) as a whole is classified as a convex optimization problem.

3.3.3. Smoothness

Problems having a smooth (i.e.: twice differentiable) objective function may be solved using algorithms that exploit information from its gradient and Hessian, which are relatively fast and can provide first order optimality measure values. Formulated differently: smoothness of the objective function avoids the need to resort to derivative free algorithms, thereby improving the tractability of the

[4] Note that the proposed solution method does not capture secondary interaction effects (section 3.2.2) which means that the optimization problem is not fed with information to actively steer it towards solutions where combined changes in the OD matrix yield lower OD matrix deviations whilst still fitting it to observed network data. This means that it can only fit an OD matrix to observed network data by increasing the deviation from the prior OD matrix which means that it makes no sense to include a stop criterion on the deviations to the prior OD demand.

solution method.

In Section 3.2.4, first order derivatives of the objective function where calculated whereas in Appendix A, it is shown that the second order derivatives (the Hessian matrices) of all three objective function components can also be calculated. This means that the objective function is indeed twice differentiable in **D**, which was implicitly already concluded in Section 2.3.5 when the objective function of the simplified optimization problem was classified as quadratic.

4. Application on a small network

In this section, the added value of the proposed matrix estimation methodology is demonstrated using four test case applications on the well-known Sioux Falls test network that gradually build up from the traditional approach used for SCRTA models towards the proposed solution method from Section 3. First, the specifics of the used SCCTA model implementation (4.1) and network (4.2) are described. Then, the evaluation framework (4.3) and test case applications (4.4), are defined and results are presented (4.5).

4.1. SCCTA model implementation

The mathematical relationships in SCCTA assignment models for the SUE have already been described in Section 2.2. In this section, specifics of the used SCCTA model implementation are briefly considered.

With respect to the network loading submodel, the SCCTA model STAQ (Static Traffic Assignment with Queueing, Brederode et al., 2019) is used which possesses all the favorable properties described in Section 1. Note that in (Brederode et al., 2019), the assignment model used in this paper is referred to as STAQ - variation without spillback, but for brevity, in this paper we shall abbreviate this to STAQ. Findings in this paper with respect to the matrix estimation method apply to any (future) SCCTA class network loading submodel using an explicit node model described in 2.2.

The used route choice submodel relies on a route set that is pre-generated from the digitized transport network using a route set generator. The route set generator used combines the Dijkstra algorithm to find the shortest path between each OD pair and the repeated random sampling process on free flow link travel times from (Fiorenzo-Catalano, 2007) to generate alternative routes. Route filters calibrated on GPS data (Fafieanie, 2009) are applied after the repeated random sampling process to reduce route overlap, remove irrelevant routes and restrict the size of the set of potential routes.

To check for convergence to SUE conditions, the adapted relative duality gap as derived in (Bliemer et al., 2013) is used, which accounts for perception errors and thus reaches zero upon convergence when using the MNL route choice model:

$$DG = \frac{\sum_{rs\in RS}\sum_{p\in P_{rs}} Q_p(c_p + \ -\mu_{rs}^{-1}\ln\left(Q_p - \zeta_{rs}\right)}{\sum_{rs\in RS} D_{rs}\zeta_{rs}} \tag{43}$$

where $\zeta_{rs} = \underset{p\in P_{rs}}{min}[c_p + \mu_{rs}^{-1}\ln Q_p]$ represents the minimum stochastic path cost on OD pair *rs*. In line with (Boyce et al., 2004; Brederode et al., 2019; Han et al., 2015; Patil et al., 2021), for all test applications in this paper, a threshold value of 5E-05 is used as the stop criterion for the traffic assignment model. Note that (Bliemer et al., 2014) provides proof for existence and uniqueness of the user equilibrium solution for exactly this SCCTA model implementation.

As mentioned in 2.2.4 convergence to the SUE is enforced and sped up by smartly averaging route demands about iterations. To this end, the method of self-regulating averages (SRA, Liu et al., 2009) is used which tends to provide fast convergence with high precision. Note that, apart from using this more efficient averaging scheme, this paper uses the exact same SCCTA model (i.e.: it uses the same network loading submodel and reaches the same SUE conditions) as described in (Bliemer et al., 2014), who use the method of successive averages instead.

4.2. Network and observed input data

To evaluate the quality and convergence properties of the methodology synthetic test cases on the well-known Sioux Falls network are used which contains 24 centroids (that also serve as nodes) and 35 links. The network and OD matrix are downloaded from (Transportation Networks for Research Core Team, 2019) but the OD matrix was adapted because the original OD matrix represents extremely high levels of congestion: all or nothing assignment yielded 64% of the links having a volume/capacity (V/C) ratio greater than one, whereas 26% of the links had a V/C ratio greater than two.

Such high demand is not desired for test case applications because its model variables are not within a range that is representative for real world situations, because the delays caused by such high demands would force travelers to change their mode, departure time and/or trip frequency, effectively lowering demand for the mode and time period considered by the assignment model.

Dividing the OD matrix by a factor of two is in line with findings in (Chakirov and Fourie, 2014) and yield link demands within a more realistic -but still very high- level of congestion: 22% links with a V/C ratio higher than one and 3% links with a V/C ratio higher than two. Interpreting this OD matrix as the 'true' OD demand it was assigned using STAQ to generate 'true' observed flows, congestion patterns (constrained (out)links) and travel times. This 'true' OD matrix contains 528 OD pairs with nonzero demand, and during assignment 1430 unique routes where generated and used, yielding 2.71 routes per OD pair on average.

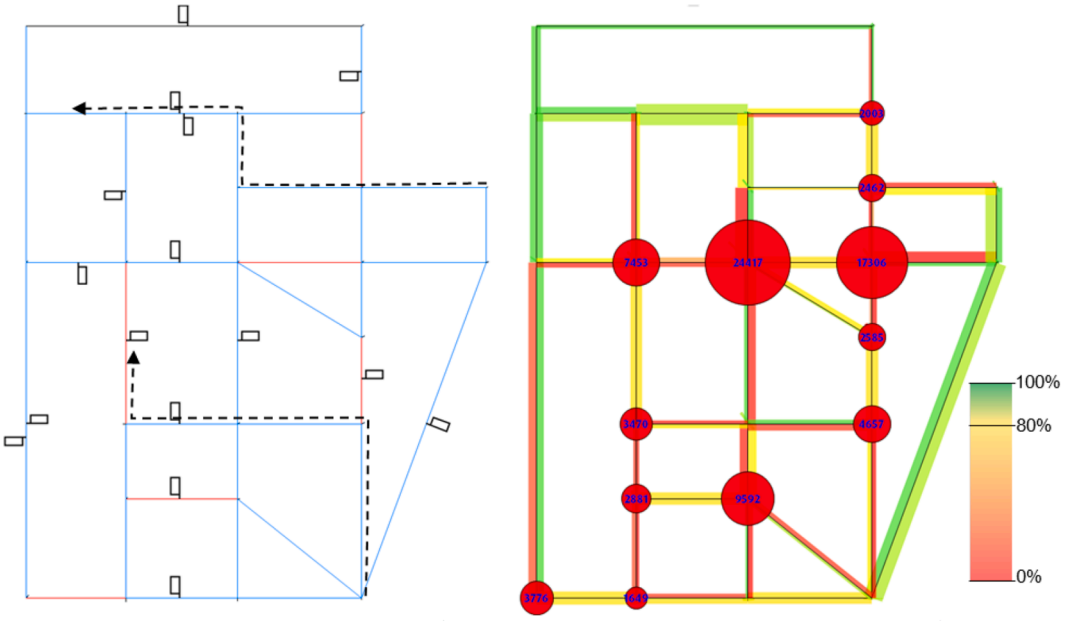


Fig. 4. Sioux Falls network: left: count locations (black flags next to links on the side of direction of travel), observed route definitions (dashed arrows) and congestion patterns (blocking links in red); right: assignment results of 'true' OD matrix (width: flow; color: speed as percentage of maximum speed; pie charts: number of vehicle hours lost in (vertical) queue)).

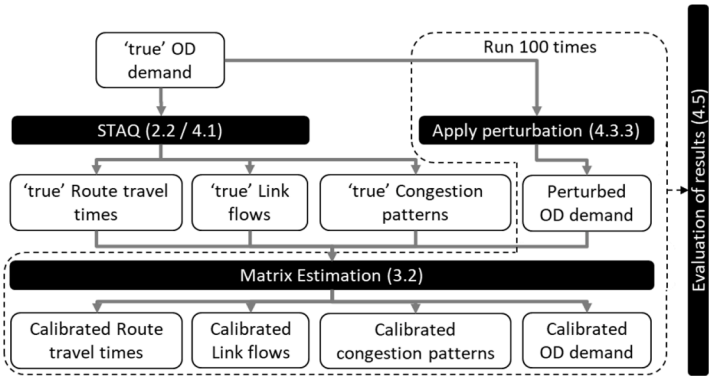


Fig. 5. Evaluation framework used for test case applications 3 and 4.

Note that in real world applications, observed flows, speeds, travel times and congestion patterns are observed in a more fine-grained time interval than a typical study period of SCCTA models, hence time-aggregation of observed values is required. In line with assumptions of static traffic assignment models, time- averaged values of observed flows, speeds and travel times should be used, and time average values for speeds or densities should be used to derive congestion patterns. It is however not possible to correct real world observed data to another assumption made in static traffic assignment models: the network is assumed empty before and after the study period. The solution to this problem would be to extend the approach to use a semi-dynamic capacity constrained traffic assignment model (SDCCTA) such that residual traffic is accounted for in both the model and the observations, which is therefore recommended in Section 6.2.

The (arbitrary selected) set of count locations and the set of constrained outlinks as well as the 2 selected routes with observed queuing delays (one traversing a single vertical queue, the other one traversing two vertical queues) are displayed on the left hand side of Fig. 4 whereas the prior assignment results (link flows, (vertical) queue sizes and link speeds as a percentage of maximum speeds) are displayed on the right hand side of Fig. 4.[5]

[5] Note that In Figure 4, to indicate the turning movement from which the queuing delay on a routes last node is included, the route is defined to end halfway onto the last nodes downstream link.

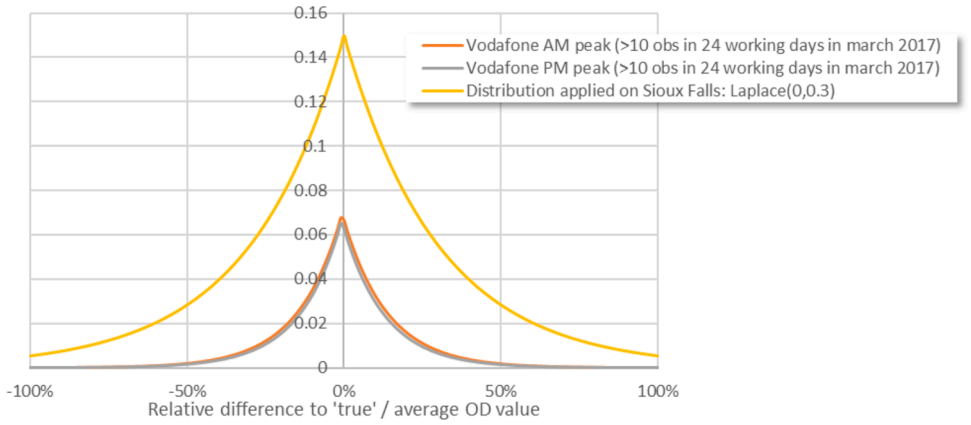


Fig. 6. Probability mass functions of the Laplace distributions fitted to (relative) variations in Vodafone data and the Laplace distribution applied on the Sioux Falls test cases in this section.

4.3. Evaluation framework

The performance of a matrix estimation methodology relates to the difficulty of the problem it needs to solve, which is related to the amount and (in)consistency and sensitivity of its input data (observed flows, congestion patterns, prior OD matrix and travel times).

Inconsistencies in input data force the solution methodology to (implicitly) choose or average between inconsistent datapoints which deteriorates the quality of the output and the speed and end-level of convergence. Sensitivity of the model also has influence on the convergence of the lower level, as very sensitive models (i.e.: high levels of congestion) force the use of small step sizes and the use of smart step size calculation methods, which both increase calculation time. Furthermore, high sensitivities amplify the negative effects on convergence due to differences between the true (1) and simplified (23) optimization problem.

To evaluate the performance of the matrix estimation methodology, the evaluation framework from Fig. 5 is used. For this synthetic application on the Sioux Falls network, a 'true' OD matrix ($\mathbf{D}_{true}$) is available, whereas in real model applications this is not the case. Therefore, $\mathbf{D}_{true}$ is perturbed and used as the prior OD matrix together with the 'true' observed link flows and travel times. Recognizing that inconsistencies are merely coincidences of inconsistent inputs each test case application is run repetitively 100 times with a differently perturbed prior OD matrix, thereby robustly evaluating the performance. Recognizing the effect of sensitivity of the model, the prior OD matrix (being perturbed around $\mathbf{D}_{true}$) as well as the other input represent a situation with much congestion (as shown in the right hand side from Fig. 4).[6]

4.3.1. Stop criteria and performance indicators

For all applications in this paper, the thresholds for the convergence criterion (defined in Section 3.2.6) are set to 1% for the average percentual deviation in link flow ($\varepsilon_{\mathbf{A}}$) and 5% for the average percentual route delay deviation ($\varepsilon_{\mathbf{P}}$). These values fall well below accuracy levels of observed link flows and route delay deviations in strategic transport models. The threshold of the stability criterion is set to zero, meaning that it is only met when the upper level yields the exact same OD matrix in two concurrent iterations. The maximum number of iterations is set to 10, because a solution method that would require that many iterations would be unsuitable for application on large scale networks due to computational requirements.

With respect to comparison of observed and calibrated link flows and route travel times, the convergence criterion effectively monitors both. Therefore, strictly spoken, there is no need to monitor these explicitly when evaluating application results. Instead, the number of iterations, the number of upper level function evaluations and the calculation time (on a machine with AMD Ryzen 9 3900X CPU (12 cores) @3.79 Ghz) required for convergence are monitored as the performance indicator for the match between link flows and route travel times. However, for clarity, average link flow and route delay deviations will also be included in the analysis of the results in Section 4.5.

With respect to comparison of OD demands, the evaluation framework from Fig. 5 allows to evaluate to what extent the matrix estimation method retrieves $\mathbf{D}_{true}$ when fed with congestion patterns, link flows and travel times that are consistent with it, but also how much deviation from the prior OD matrix it requires. The structural similarity index (SSIM, Djukic et al., 2013) is used to compare both the estimated and 'true' OD matrices as well as the estimated and prior OD matrices. More specifically, mean SSIM (mSSIM) values are used as performance indicators. Following suggestions by (Ros-Roca et al., 2018) mSSIM values are calculated by averaging SSIM values per row of the OD matrix considered. The mSSIM is an indicator for the similarity in matrix structure (by the definition from (Behara et al., 2020): the arrangement of the destinations from each origin). To also consider the actual differences in values per OD pair ((Behara et al., 2020) use the term 'mass'), for the comparison between estimated and prior OD matrices root mean squared

[6] Note that more inconsistencies could be introduced by also perturbing around the 'true' observed flow and travel time values. We leave this idea for further research.

Table 1
Distinctive properties of the four test case applications.

Test case		Response functions		Congestion patterns				Prior		Flows		Queuing delays	
#	Referral	$\mathbf{y}(\mathbf{D})$	$\boldsymbol{\tau}(\mathbf{D})$	$\boldsymbol{\chi}$	δ_j (deficits)	δ_j (surpluses)	nudging	w_1	$\tilde{\mathbf{D}}$	w_2	$\tilde{\mathbf{y}}$	w_3	$\tilde{\tau}$
1	[REF]	Eq. (13)		∅	∅	∅		½	$\mathbf{D}_o$	½	from $\mathbf{D}_{true}$		∅
2	[+LS]	Eq. (13)		from $\mathbf{D}_{true}$	1.01	0.99		½	$\mathbf{D}_o$	½	from $\mathbf{D}_{true}$		∅
3	[+LS+S]	Eq. (15)		from $\mathbf{D}_{true}$	1.01	0.99		½	$\mathbf{D}_o$	½	from $\mathbf{D}_{true}$		∅
4	[+LS+S+QD]	Eq. (15)	Eq. (17)	from $\mathbf{D}_{true}$	1.01	0.99(94);0.9 (6)	(9)/100	1/ 3	$\mathbf{D}_o$	1/ 3	from $\mathbf{D}_{true}$	1/ 3	from $\mathbf{D}_{true}$

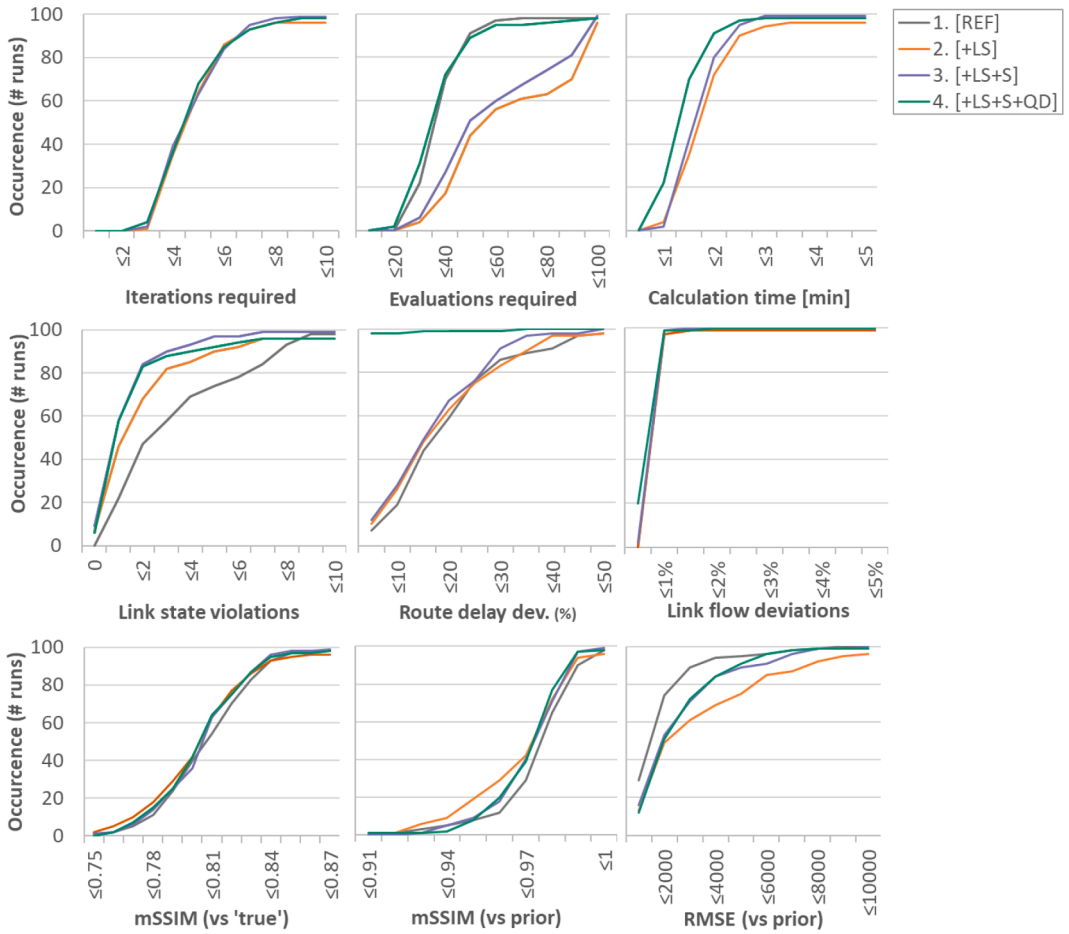


Fig. 7. cumulative distributions of performance indicators for all four test case applications. Note that 40 runs fall outside the range of the vertical axis of the mid upper graph for test case application 4. Therefore it is noted here that the 95th percentile of the number of evaluations required for this test case application is 214.

error (RMSE) values are also presented.

Differences between observed and estimated congestion patterns are enforced to be nonexistent in the upper level by the link state constraints. But because the problem that is solved in the upper level (23) is a simplified version of the true problem (1), this does not guarantee that all link state constraints are satisfied after application of the SCCTA model in the last iteration. Therefore, the number of link state violations in the final assignment results of the lower level are explicitly monitored as a performance indicator of the match with observed congestion patterns.

4.3.2. Generation of perturbed OD matrices

Treating entries in an OD matrix as exponentially distributed random variables, differences between different OD matrices would be best governed by a Laplace distribution (Kotz et al., 2001). Therefore, a Laplace distribution is used to randomly draw the cell-by-cell perturbations applied to the true OD matrix to generate the 100 OD matrices used as priors.

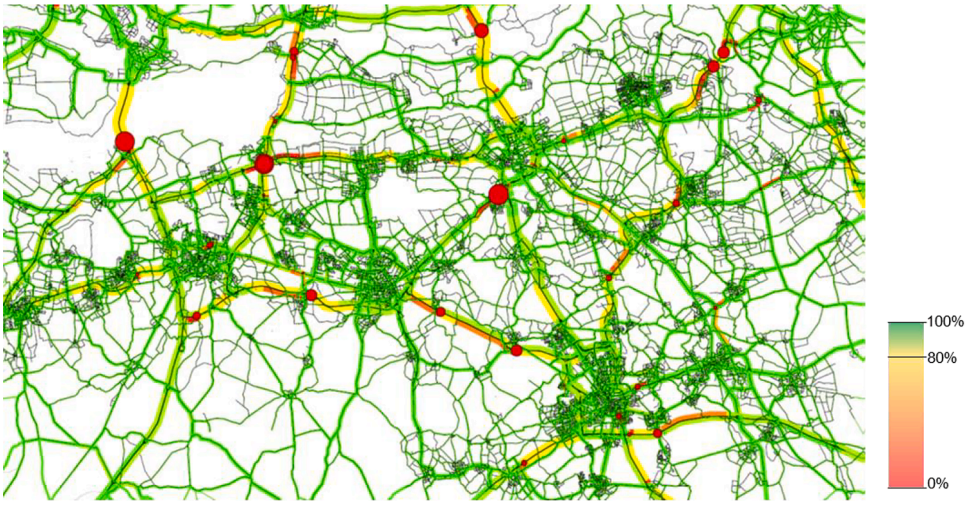


Fig. 8. (study area of the) BBMB network: assignment results of prior OD matrix (width: flow; color: speed as percentage of maximum speed; pie charts: number of vehicle hours spent in (vertical) queue)).

To determine the relevant size range of the perturbations, peak-hour OD matrices derived from observed Dutch mobile phone data from Vodafone for all non-holiday workdays in March 2017 where used. First, for each OD pair having more than 10 observations in the considered peak hour during the considered days, differences between the cell's values over the different days and its average value where calculated and expressed as percentual differences to the average. This yielded a dataset of about 4.5 million relative differences per peak hour on which Laplace distributions where fitted yielding location parameters close to 0 and scale parameters around 0.135 for both peak periods. The estimated distributions are displayed in Fig. 6 together with the distribution applied on the Sioux Falls test cases. To ensure that the test cases represent a worst-case scenario for the matrix estimation methods, the distribution applied on the Sioux Falls test cases uses a much higher and wider distribution by increasing the scale parameter to 0.3, which means that the structure of the OD matrix is severely changed by the perturbations, but the number of nonzero OD pairs remained 528 for all perturbed ODmatrices.[7] Furthermore, to make sure that the performance of the proposed solution method is tested in congested conditions, it was verified that the percentage of demand per observed link varied between 5% and 100% for all use cases.

Note that the fitted distributions imply that no structural bias is introduced during generation of perturbed OD matrices. Given the use case of the method (Section 1.1: to refine prior OD demand from a demand model with data on link and route level), this a deliberate choice for the test case applications, because if in practice there would be a structural bias between prior assignment results and observed demand levels, authors suggest using a global scaling factor to remove it before applying the matrix estimation method, such that the prior structure is kept in-tact as much as possible.

4.4. Test case applications

Four test case applications are defined that gradually add solution method features and support for additional types to the traditional approach used for SCRTA models until we arrive at the proposed solution method from Section 3. Distinctive properties of the four test case applications are summarized in Table 1. In all test case applications, the same route set with 1430 routes was used yielding an average of 2.71 routes per OD pair (since all perturbed OD matrices have the same number of nonzero OD pairs). Furthermore, in all test case applications, the same pre-generated route set (Section 4.1) is used.

The first test case application (referred to as [REF]) employs the approach used in (Brederode et al., 2017),weighing (normalized) deviations from prior demand and (normalized) deviations from observed link flows equally ($w_1 = w_2 = 1/2$). [REF] acts as a reference in which sensitivities and link state constraints are omitted, in which case it is not possible to include observed queuing delays nor congestion patterns. Therefore, in [REF], problem (23) simplifies into:

$$\begin{aligned} &\mathbf{D}^* = \underset{\mathbf{D}}{\operatorname{argmin}}\left(w_1 \sum (\mathbf{D}-\mathbf{D}_0)^2 + w_2 f_{2,N} \Big/ f_{1,N} \sum (\mathbf{y}(\mathbf{D})-\tilde{\mathbf{y}})^2\right) \\ &\text{Subjectto}: \ \mathbf{y}(\mathbf{D}) = \hat{\alpha}\psi\mathbf{D} \\ &0 \le \mathbf{D} \le \overline{\mathbf{D}} \end{aligned} \tag{44}$$

which resembles, apart from the traffic assignment used, any of the gradient based approaches described in (Abrahamsson, 1998).

[7] Note that during application, the distribution was truncated such that OD pairs for which perturbations of less than −100% would be applied where assigned a value of 1 (to not introduce new OD pairs with value 0) while OD pairs with a true value of more than 500 and a perturbation of more than 100% where truncated to that value.

The second test case application (referred to as [+LS]) adds link state constraints from observed congestion patterns derived from $\mathbf{D}_{true}$ by adding Eq. (18) to optimization problem (44). The minimum capacity surpluses δ_j (on non-constraining outlinks $\{j \in L|\chi_j = 1\}$) and deficits (on constraining outlinks $\{j \in L|\chi_j = -1\}$) that act as a buffer around discontinuities in $\boldsymbol{\alpha}_n(\mathbf{T}_n)$ are set to 0.99 and 1.01, respectively. By adding link state constraints, transitions between traffic regimes are avoided, which should improve convergence and reduce the link state violations. However, because exogenous congestion patterns (from $\mathbf{D}_{true}$) are used inconsistencies between the exogenous congestion patterns and the prior OD matrix may cause the objective function to become non-convex (as described in 2.3.3), hence reducing convergence in [+LS].

The third test case application (referred to as [+LS+S]) adds sensitivities (+S) to the response function for link flows to account for the sensitivity of the assignment matrix to changes in OD demand. To do so, the response function for link flows is restored to the first order Taylor approximation (Eq. (15)), effectively arriving at problem (23), but excluding observed queuing delays (i.e.: $w_3 = 0$). The inclusion of sensitivities should improve convergence as the upper level has more accurate information which should also lead to less link state violations and less unnecessary changes to the prior OD demand.

The fourth test case application (referred to as [+LS+S+QD]) adds observed queueing delays (+QD) derived from $\mathbf{D}_{true}$. This means that (normalized) route delay deviations are added to (23), weighted equally to both (normalized) OD demand deviations and (normalized) link flows ($w_1 = w_2 = w_3 = 1/3$). Because the queuing delays operate on the level of individual turning movements (instead of aggregations over inlinks (for link flow deviations) or outlinks (for link state constraints)), the solution candidates evaluated by the upper level contain relatively large changes to individual turn demands. This increases the likelihood that discontinuities in $\boldsymbol{\alpha}_n(\mathbf{T}_n)$ are crossed due to difference between the true function within the node model (8) and its linear approximate derived by finite differences (see Section 3.2.2) used in the upper level. This mechanism is described in more detail in Appendix C and it led to 9 runs that did not converge within the maximum number of (10) iterations. For these runs, a nudging iteration (Section 3.2.6) was prepended to reduce the chance that the mechanism occurs in the first iteration, whereas for 6 of these 9 runs it was also necessary to increase the buffer around discontinuities in $\boldsymbol{\alpha}_n(\mathbf{T}_n)$ for one or two of the non-constraining outlinks by lowering the minimum capacity surpluses δ_j from 0.99 to 0.9 to prevent the mechanism to occur in later iterations.

4.5. Results

Results of all four test case applications are displayed in Fig. 7. Because each test case is run a hundred times with a different prior demand matrix, all performance indicators are summarized as cumulative distributions of each indicator over the different runs. Recall from Section 4.3.1 that stop criteria are defined for link flow and route delay deviations. Note however, that for [REF], [+LS] and [+LS+S], the stop criterion on route delay deviation is ignored, as in these applications, route delays are not included in the optimization. This means that for these applications convergence is reached when only link flow deviations meet the stop criterion, whereas for [+LS+S+QD] convergence is only reached when both criteria are met.

Considering the level of convergence of the bi-level problem (upper left graph), the number of converging runs is read by looking at the value at iteration ≤ 10. This shows that in [REF] 98/100 runs converge. Addition of link state constraints [+LS] causes a reduction to 96/100 converging runs, which shows that (at least on this network), the positive effect of added stability is outweighed by the negative effect of (potential) additional data inconsistencies. As expected in Section 4.3.1, addition of sensitivities to the response function [+LS+S] increases the number of converging runs (to 99/100) as the upper level has more accurate information. Addition of queuing delays [+LS+S+QD] only slightly reduces the number of converging runs to 98/100. However, without the algorithmic enhancements (nudging and lowering the minimum capacity surpluses on specific link state constraints) the number of runs converged would have been 91/100, showing that addition of observed queuing delays without mitigating measures has the largest negative effect on the level of convergence. Note that the speed of convergence barely varies over the different test case applications; only the addition of queuing delays structurally lags about one iteration for runs requiring more than four iterations.

Note that for all four test case applications the cumulative distributions in the upper left graph indicate that no additional runs are converging beyond iteration 8 or 9. Additional test runs (not described in this paper) with the maximum number of iterations criterion increased beyond 10 confirm this observation. Analysis of individual non-converging data points within some of the non-converging runs point towards differences between the route choice probabilities from the 'true' and perturbed OD demand. These differences can cause certain combinations of observed link flows and/or route queuing delays to become inconsistent, causing the optimal, still feasible, solution to not satisfy the convergence criteria (Section 4.3.1). That differences due to route choice inconsistencies are indeed the cause is confirmed by the fact that additional test runs (not described in this paper) where route choice probabilities from $\mathbf{D}_{true}$ where kept fixed over iterations all converged within two to four iterations. Note that in practice, non-converging datapoints are easily detectable and may be resolved by increasing the difference tolerance on one or both datapoints or removing one of the datapoints.

From the number of upper level function evaluations required for convergence (upper mid graph) two mechanisms are derived. Firstly, adding data sources increases difficulty of the optimization problem, and thus requires more function evaluations, which is shown by comparison of [+LS] with [REF] for the effect of addition of congestion patterns and comparison of [+LS+S+QD] with [+LS+S] for the effect of addition of queuing delays. Secondly, enhancing the gradient information in the upper level by including sensitivities increases effectiveness of the upper level solver and thus reduces the number of function evaluations required, which is shown by comparison of [+LS+S] with [+LS].

Considering the calculation time required for convergence (upper right graph) in relation to the previous two graphs shows that relatively small differences in the number of iterations required and the relatively large differences in the number of function evaluations required translate into relatively small differences in calculation time. This reveals that most time is still spent in the lower level (and within the lower level the SCCTA run takes up most of the time), whilst the upper level is relatively fast.

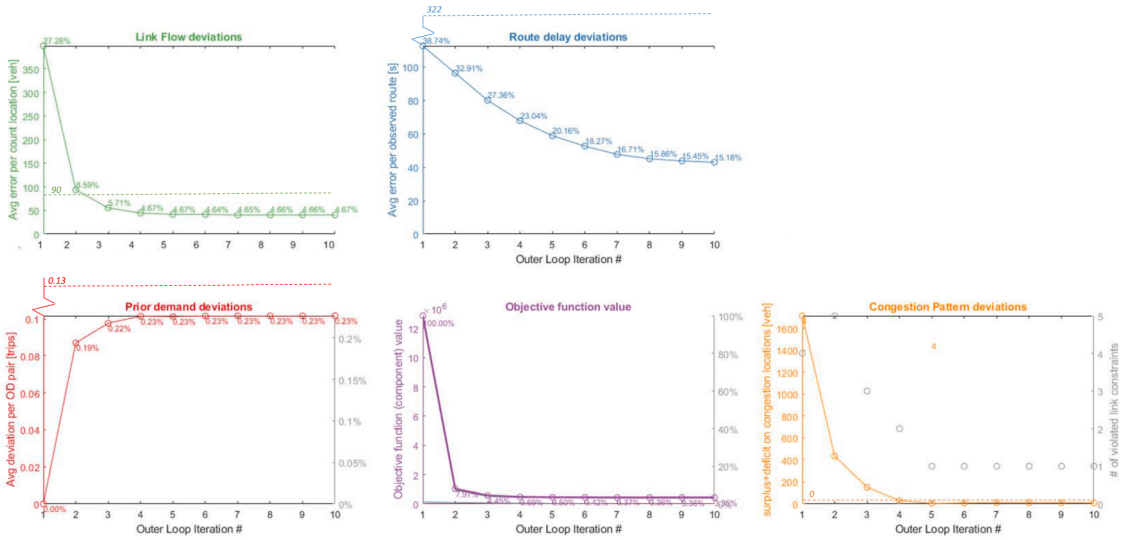


Fig. 9. convergence of proposed methodology on the BBMB model in terms of average link flow deviations (upper left graph), route delay deviations (upper mid graph), prior demand deviations (lower left graph), congestion pattern deviations (lower right graph) and objective function value (lower mid graph). Dashed lines indicate minimum deviations yielded by the software currently used for matrix estimation in the BBMB, which employs the [REF] method described in 4.4.

With respect to the number of link state violations (left graph on second row), comparison of [+LS] with [REF] shows the effectiveness of the link state constraints, whereas differences between these cumulative distributions and a (non-shown) vertical asymptote at 0 violations represent the number of violations caused by the difference between the simplified problem solved in the upper level and the true bi-level optimization problem. Results from test case application [+LS+*S*] compared to [+LS] show that addition of sensitivity information to the gradient decreases the number of link state violations, as expected in Section 4.3.1, while adding queuing delay information (compare [+LS+*S*+QD] with [+LS+*S*]) does not have a clear effect. The latter observation makes sense because in these test case applications, congestion patterns and queuing delays are fully consistent, as these are both derived from $\mathbf{D}_{true}$.

With respect to the average route delay deviations (mid graph on second row), comparison of [+LS] and [+LS+*S*] with [REF] shows that inclusion of link states and sensitivity information only slightly improves the fit on observed route delays, whereas the proposed method [LS+*S*+QD] is required to include observed queuing delays. The limited effect of adding link state and sensitivity information on the fit on observed route delays shows that there is indeed relatively limited correlation between model variables, suspectedly because temporal correlations are avoided as the solution method employs a static (hence time aggregated) assignment model (recall from Section 1).

For the sake of completeness, Fig. 7 also includes a comparison of average link flow deviations (right graph on second row). This graph confirms that, except for the 7 non-converging runs already described above, the stop criterion of 1% average link flow deviations is met for all four test case applications.

Comparison of the estimated against the 'true' OD matrix (lower left graph) shows no notable differences between the different runs. Apparently, even though observed link flows, congestion patterns and queuing delays are all derived from $\mathbf{D}_{true}$, in all four hundred runs there is an abundant number of optimal solutions close to the prior. Additional test runs (not included in this paper) show that this remains the case, even when the search space is increased by excluding the prior demand component in the objective function (by setting w_1 to zero). This demonstrates that although the proposed solution method finds the global optimum to the simplified optimization problem for each iteration, this does not mean that it finds the global optimum (if it exists) to the true optimization problem.

Comparison of the estimated against the prior OD matrix (lower mid and right graphs) shows that adding congestion patterns leads to substantial larger deviations from the prior OD matrix compared to the results from test cases with added sensitivity information. This demonstrates the effect of the increased effectiveness of the upper level solver due to the sensitivity information on the quality of the estimated matrix. Compared to the mSSIM, the RMSE indicator shows larger differences, since the latter captures all differences, whereas the former only targets differences in matrix structure (Section 4.3.1).

5. Application on a large network

In this section, results of a large scale application of the proposed solution algorithm on (data from) the strategic transport model of the province of Noord-Brabant, the Netherlands (Heynicks et al., 2016) are presented.

Table 2
calculation times and related indicators from application of proposed and reference method on the BBMB model.

Iteration#	# of STAQ iterations	calculation time lower level	# of solver iterations	calculation time upper level	total calculation time
1	12	03:15:12	100	08:04:42	11:19:54
2	14	02:52:26	100	07:43:52	10:36:18
3	13	02:43:45	100	06:30:09	09:13:54
4	13	02:42:54	35	01:44:18	04:27:12
5	13	02:42:54	71	04:01:08	06:44:02
6	13	02:45:07	46	02:11:37	04:56:44
7	13	02:46:04	35	01:02:40	03:48:44
8	13	02:44:35	35	00:45:30	03:30:05
9	13	02:43:48	35	00:45:36	03:29:24
10	13	02:54:04	0	00:00:00	02:54:04
Proposed method (10 iterations)	**130**	**28:10:49**	**557**	**32:49:32**	**61:00:21**
Proposed method (3 iterations)	**39**	**8:51:23**	**300**	**22:18:43**	**31:10:06**
Reference method (4 iterations)	**40**	**15:06:35**	**–**	**04:15:31**	**19:22:06**

5.1. Transport model and observed input data

The network and prior OD demand for road traffic of the base year (2015, version S107) of the provincial model of Noord-Brabant (abbreviated in Dutch to 'BBMB') is used. This network contains 1425 centroids 145.269 links and 103.045 nodes. The prior OD matrix used describes the AM peak period (07:00–09:00) and contains 1.580.764 OD pairs with nonzero demand. During assignment 5.162.010 unique routes where generated and used, yielding 3.26 routes per OD pair on average.

With respect to observed input data, the full BBMB count-data set for the AM peak period is used, which contains observed link flows for 415 count locations, along with a set of observed travel times on 24 (highway) routes. Up until now, this set was only used for validation purposes, as the prevailing matrix estimation method of the BBMB-model is not capable of including observed queuing delays. Link state constraint values are derived from assignment results of the prior demand matrix. To reduce problem size, the upper bound on od demands (Section 2.3.5) is set to $\overline{\mathbf{D}} = 2\mathbf{D}_0$, yielding a set of relevant links ($J_{\overline{\mathbf{D}}}$) containing 21 constraining and 1583 non-constraining links (hence a reduction of 98.9% compared to the set L containing all links).

Note that the BBMB model employs junction modeling, which means that its node models do not only account for constraints due to limited supply on outlinks (in the form of link capacities), but also for the effect of limited supply due to conflict points on the junction itself (i.e. crossing flows; in the form of turn capacities). To support this in the context of the proposed matrix estimation method, turn capacities are calculated using the junction modeling component of OmniTRANS (Bezembinder and Brandt, 2016) and included as internal node constraints (Tampère et al., 2011) while running the SCCTA model (Section 3.2.1) and approximating sensitivities (Section 3.2.2).

5.2. Convergence and calculation time

Ten iterations of the proposed methodology where run on the BBMB model, after which all convergence indicators (solid lines in Fig. 9) seem to have stabilized. The minimum capacity surpluses and deficits added to prevent unintentional regime switches when running the lower level (Section 2.3.3) where both set to 1% (i.e.: $\delta_j = 0.99$ for non-constraining outlinks and $\delta_j = 1.01$ for constraining outlinks). The weighting parameters in the objective function (45) where set to w_1=0.01 (prior), w_2=0.12 (link flows) and w_3=0.87 (queuing delays), and directly applied (i.e.: normalization as described in 2.3.4 was omitted). Also in Fig. 9, dashed lines indicate minimum deviations yielded by the software currently used for matrix estimation in the BBMB, which employs the [REF] method described in 4.4.

Considering the average link flow deviation per count location, the upper left graph shows that these quickly reduce from 27% to around 5% and that it outperforms the reference methodology (which averages on 90 vehicles per count location) in iteration three. This graph also shows that for link flow deviations, to save calculation time, the algorithm could be stopped after iteration four, as results hardly improve afterwards.

The average route delay deviations (upper mid graph) show a reduction from around 39 to around 15 percent, which translates to a reduction from 112 s (in a range from 13 up to 241 s) to 43 s (in a range from 1 up to 114 s). Note that from iteration 7 onwards, the fit on link flows slightly deteriorates while the route delay deviations keep improving. During these iterations, the objective function keeps improving, which demonstrates the weighting of objective function components. Further note that the reference method does not consider route delay deviations, which causes an average deviation of 322 s per observed route, which is (much) larger than the average route delay deviation when assigning the prior demand.

Considering the congestion patterns (lower right graph) deviations reduce from a deficit of more than 1700 vehicles on four different locations in the first iteration to zero vehicles from the fifth iterations onwards. The reference method also converges to zero vehicles. Note that from the fifth iteration onwards, the graph still reports one location on which the congestion pattern is not matched.

Table 3
turn based flow acceptance factors and sensitivity of link flows on corridor network for different demand intervals.

Demand interval	Flow acceptance factors		Sensitivity of link flows		
D_{rs}	α_{23}	α_{12}	$\frac{\partial y_3(D_{rs})}{\partial D_{rs}}$	$\frac{\partial y_2(D_{rs})}{\partial D_{rs}}$	$\frac{\partial y_1(D_{rs})}{\partial D_{rs}}$
$D_{rs} \leq 1000$	1	1	>0	>0	>0
$1000 < D_{rs} \leq 2000$	($^1/_2$,1)	1	0	>0	>0
$2000 < D_{rs} \leq 3000$	½	($^2/_3$,1)	0	0	>0

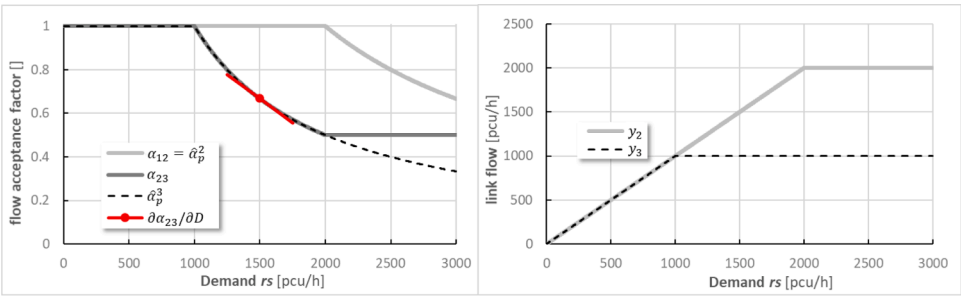


Fig. 10. Corridor network (top), flow acceptance factors (left) and link flows as function of demand (right).

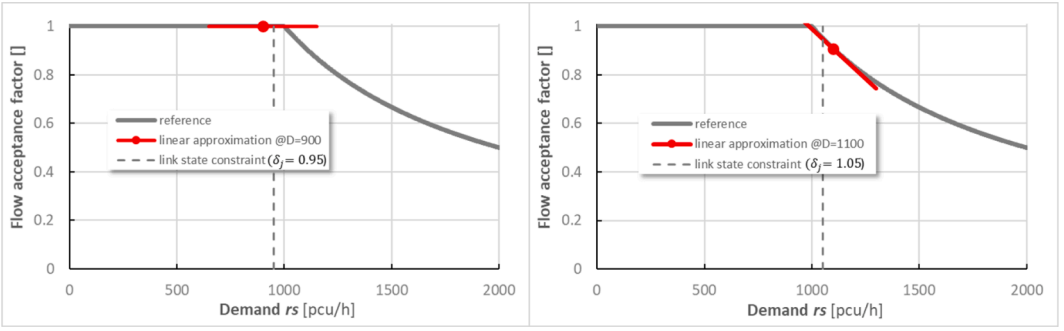


Fig. 11. $\alpha_{23}(D_{rs})$, sensitivity approximations and link state constraints for not constraining (left) and constraining (right) cases in reference situation with effective turn capacity equals 1000 pcu/h.

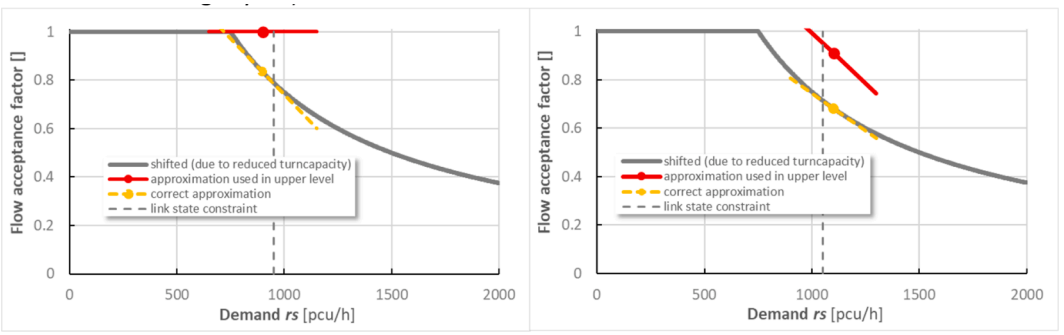


Fig. 12. $\alpha_{23}(D_{rs})$, sensitivity approximations and link state constraints for not constraining (left) and constraining (right) cases when effective turn capacity is reduced to 750 pcu/h.

This is because the minimum capacity surpluses and deficits added to prevent unintentional regime switches when running the lower level (Section 2.3.3) are included in this graph. Inspection of the assignment results showed that the concerning location is a constraining outlink for which the demand is indeed higher than its capacity, but lower than the capacity multiplied by δ_j.

Considering deviations to the prior OD demand, the lower right graph shows that, in correspondence to the link flow deviations,

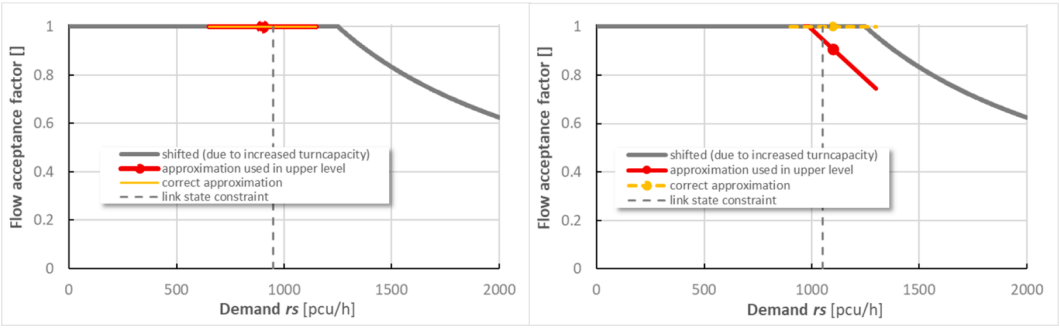


Fig. 13. $\alpha_{23}(D_{rs})$, sensitivity approximations and link state constraints for not constraining (left) and constraining (right) cases when effective turn capacity is increased to 1250 pcu/h.

most changes to the OD matrix are done in the first four iterations. This is also confirmed by the objective function values (lower mid graph). The proposed method requires less than 0.1 trips per OD pair on average, which is less than the reference method, which required a change of 0.13 trips per OD pair on average.

Calculation times per iteration of the proposed solution method on an AMD Ryzen 9 3900X CPU (12 cores) @3.79 Ghz are displayed in Table 2, along with the total calculation time of the reference method.[8] The total calculation time of the proposed solution method (10 outer loop iterations) amounts 61 h, of which 46% is spent in the lower and 54% is spent in the upper level. Apart from the first iteration (in which the route set and mappings between OD-, route-, turn- and link level are generated), lower level calculation times show limited variation at around 2:45 h per iteration. This is explained by realizing that most of it is spent during application of the SCCTA model (STAQ) whilst the number of STAQ iterations only varies between 12 and 14 iterations per (outer loop) iteration, translating to 12:19 up to 12:46 min per STAQ iteration. Upper level calculation times vary extensively between 45 min and 8 h per (outer loop) iteration, translating to 1:18 up to 4:51 min per solver iteration. This means that not only the number of solver iterations, but also the calculation time per solver iteration varies extensively.

Both the upper level calculation times as well as the number of solver iterations indicate that most effort is put in the first three (outer loop) iterations which corresponds to the reductions of the objective function value per iteration and the amount of change in the OD matrix, which are largest in these first three iterations. As the default settings for the solver (Waltz et al., 2006) where used in this application, the number of solver iterations per (outer loop) iteration could probably be reduced and stabilized over iterations by tuning its parameters and stop criteria, but we leave this for future research.

Fig. 9 shows that, compared to the minimum deviations from the reference method (dashed lines), the proposed method (solid lines) attains lower deviations on all objective function components from iteration three onwards. We therefore compare the calculation times of the reference method (19 h) with the calculation time spent in the first three iterations of the proposed method (31 h). This comparison shows that the proposed method spends 61% more time, but yields lower link flow and prior demand deviations and much lower route delay deviations, whilst performing equally on congestion pattern deviations.

5.3. Findings on large network application

The most important finding from the application on the BBMB model is that the proposed solution method is indeed applicable to large scale transport models. The proposed method clearly outperforms the reference method and does so within feasible calculation times, but only because of the use of the following three problem size reducing features of the solution method.

Firstly, recall from Section 4.5 that the added value of including sensitivities for observed link flows and link states proved very limited, whereas inclusion of sensitivities proved to be a requirement for observed route queuing delays. It is suspected that this is caused by the flow maximization property of the node model, which is one of the seven requirements for first order macroscopic node models (Tampère et al., 2011). This property causes that reductions in turn flow towards supply constrained outlinks (the source of all sensitivities) due to reduced demand are compensated for by increases of flow on other turns towards that outlink. This yields stable link flows on constrained outlinks, composed by unstable flows on turns towards the outlink. Therefore, being the only data source dependent of flows on turn instead of link level, observed queuing delays require inclusion of sensitivities, whereas other (link-level) data sources do not. This led to the insight that sensitivities for observed link flows and link states may be omitted altogether, which reduced the number of the required evaluations of Eq. (29) for application on the BBMB model by more than 95%.

Secondly, reducing the problem sizes for steps 2 through 6 in Section 3.2, upper bounds on od demands are set to $\overline{\mathbf{D}} = 2\mathbf{D}_0$ (Section 5.1), reducing the number of links considered by 98.9%. Although this reduction proved sufficient for the application presented, the solution space may be widened, and/or the problem size number be further reduced by setting the upper bounds per OD pair allowing to combine absolute values and values relative to the prior demand.

[8] Note that the reference method does not use a solver, but a heuristic approach in the upper level, which explains why the number of solver iterations is left empty for this method.

Thirdly, not indicated earlier, the problem size in the upper level (steps 3 through 6 in Section 3.2) is reduced by only including paths and OD pairs that use links with observed flow, state or queuing delay data. For the application on the BBMB model, this reduces the number of OD pairs considered from 1.58 million to 1.51 million (a reduction of 4%).

6. Conclusions, discussion and further research

In this paper, an efficient solution method for the matrix estimation problem is presented using a SCCTA model which combines the favorable properties of SCRTA and DCSTA models. The solution method allows for inclusion of route queuing delays and congestion patterns besides the traditional link flows and prior demand matrix, which is novel to the best of our knowledge.

The proposed solution method uses response functions constructed from sensitivities on node level to solve a series of simplified optimization problems in the upper level, thereby avoiding costly additional assignment model runs of the lower level. Link state constraints are added to prevent usage of approximations outside their valid range as well as to include observed congestion patterns. The proposed solution method is robust, tractable and reliable because conditions under which a solution to the simplified optimization problem exist are known and because the problem is convex and has a smooth objective function.

Four test case applications on the small Sioux Falls model where conducted, each consisting of 100 runs with varied prior OD demands for robustness. These applications demonstrate the inclusion of observed congestion patterns and that adding sensitivities to the response function leads to more accurate results and slightly less computation time required.

Addition of queuing delays proved to be the biggest challenge, as these operate on the level of individual turning movements, and are therefore, contrary to link flows and link demands, not stabilized by the flow maximization property of the node model. This increases the likelihood that discontinuities in $\boldsymbol{\alpha}_n(\mathbf{T}_n)$ are crossed during estimation. This caused 9/100 runs with queuing delays to not converge, but algorithmic enhancements (nudging and lowering the minimum capacity surpluses on specific link state constraints) resolved this problem, decreasing the number of non-converging runs to only 1.

The proposed solution method yields 99 converging runs whereas analysis shows that the single non-converging run is caused by a prior OD Demand matrix from which the SUE route choice probabilities cause incompatibility between two datapoints. In practice such inconsistencies can easily be detected and removed.

In addition to the Sioux Falls results, a large scale application on the BBMB model was conducted. Results show that when using its problem size reducing features, the solution method is indeed capable of solving large scale problems within feasible calculation time and while doing so, it attains lower deviations on all objective function components compared to the reference method.

6.1. Discussion

Although this paper shows the potential for SCCTA model in the context of travel demand matrix estimation, use of this type of assignment model is still very limited. Apart from STAQ (used in this paper), all SCCTA models of which the authors are aware of (Bakker et al., 1994; Bell, 1995; Bifulco and Chrisalli, 1998; Bundschuh et al., 2006; Köhler and Strehler, 2012; Lam and Zhang, 2000; Smith, 2012, 1987) are not directly suitable for two reasons. Firstly, they use link exit capacities that constrain flow through a link only at the downstream end of the link, thereby unrealistically modeling queues inside the bottleneck links contrary to upstream of the bottleneck. Secondly, all these models lack a node model satisfying the requirements posed by (Tampère et al., 2011). Fortunately, research on SCCTA models and its more advanced sibling SCSCTA models that adds storage constraints is still ongoing (Bliemer and Raadsen, 2020; Raadsen and Bliemer, 2018).

The proposed solution approach is currently only applicable to SCCTA and SCRTA models as it solves a matrix estimation problem in which temporal correlations between model variables do not exist. Authors believe that the proposed solution approach would be extendible to the semi-dynamic capacity constrained case, where multiple time periods, each with its own stationary travel demand, are modelled and residual traffic is transferred in between (Bliemer et al., 2017) but do not believe that this approach is easily extendible to DTA context, as such models introduce temporal correlations.

Because the way in which the proposed solution derives and uses the assignment matrix ($\widehat{\alpha}(\mathbf{D})$) and its sensitivity ($\partial\widehat{\alpha}\ /\partial\mathbf{D}$) the approach is not dependent on conditions (if any) to which the assignment model results adhere. This was confirmed by test runs of the proposed method (not described in this paper for brevity) in which the SCCTA model was run for only one route choice iteration and test runs where the SCCTA model was run to equilibrium using the paired combinatorial instead of multinomial logit route choice model. This makes the approach suitable for application in a wide range of SCTTA applications spanning from operational (disequilibrium) to strategic (equilibrium) and using any first order node model (see Smits et al., 2015 for an overview of different node models proposed in literature).

The link state constraints in this paper effectively stabilize the solution method by maintaining the state of potential bottleneck links. However, constraints in the node model actually operate on the turn level, and any constraint state switch causes a discontinuity in $\boldsymbol{\alpha}_n(\mathbf{T}_n)$ (Brederode et al., 2014). This means that the link state constraints from 2.3.3 are too simplistic as they do not specify the normative turning movement. However, refining link state constraints is not pursued any further for three reasons. Firstly, during development of the solution method, tests with constraints on turning movement have been conducted, showing that the turn constraints in combination with fixed route choice probabilities constrain the simplified optimization problem too much, causing this version of the solution method to perform very poorly. Secondly, it has not proven to be a problem in any of the test case applications. Thirdly, deriving link state constraint values from observed data currently is already a challenge and deriving normative turning movements from some suitable data source would be even harder.

To increase robustness of the test results, an almost infinite number of additional test applications could be defined by introducing different inconsistencies and variations in the data by e.g. perturbing around the 'true' observed flow and travel time values, change the set of observed links, congestion patterns and routes, change the level of demand in $\mathbf{D}_{true}$, and vary the combinations of different observed types. This has not been described in this paper for two reasons. Firstly, authors feel that it is more useful to switch to true empirical data first (as done in Section 5), to be sure that the correct range of input data is tested. Secondly, a multitude of test cases have been conducted in preparation of this paper, but due to limits to the length of a scientific paper only insights from those runs relevant to the four test case applications and the large network application have been used in Sections 4.5 and 5.3.

Section 2.1.2 stated that, although incorrect in theory, omitting sensitivities in response functions appears to not be a problem in practice, which seems to withhold practitioners from using methods that include sensitivities. Authors suspect that this is the case because the sensitivities are relatively very small compared to the direct effect of changing OD demand. This is especially the case when an SCRTA assignment model is used assuming SUE conditions, as in such models only route choice sensitivities exist (the lack of capacity constraints implies flow acceptance factors are non-existent) and traffic is spread out over routes the most, dampening any effects of changed route choice.

This same mechanism but then applied to flow acceptance factors is suspected to only partly explain why the added value of inclusion of sensitivities in the SCCTA context is very limited (compare results from testcases [LS+S] to [LS]). The other part of the explanation in SCCTA context was already discussed in Section 5.2.3: the flow maximization property of the node model stabilizes link flows but does not stabilize turn flows. These hypotheses could be checked by comparing the effects of omitting sensitivities for the different types of observed data using an assignment model assuming stochastic user equilibrium conditions with an assignment model assuming all-or-nothing route choice behavior for both the SCRTA and SCCTA cases, but we leave this idea for further research.

6.2. Future research

In this section, recommendations for further research are described, in order of priority from the authors' point of view.

This paper shows that the proposed solution method converged in 99/100 test runs conducted on Sioux Falls, and that non-convergence occurs due to route choice probabilities being incompatible with some of the datapoints describing link flows and route queuing delays. This advocates for an extension to the solution method that accounts for sensitivities of the response function of route choice probabilities -just as the current method does for the response functions for link flows, queuing delays and congestion patterns- such that it can actively steer away from situations where route choice probabilities cause datapoints to become inconsistent.

Although not yet applicable to real sized transport model networks, authors believe that in time static models that take both capacity and storage constraints into account (Bliemer and Raadsen, 2020) will replace the role of SCCTA models in strategic transport model systems. Therefore, on the longer term, research into extension of the proposed solution algorithm to support storage constraints is desired.

The mSSIM performance indicator used for OD matrix comparison in this paper is known to be sensitive to the way it is averaged. Based on literature, in this paper, averaging per matrix row was chosen. Other aggregates might reveal different insights. Given this sensitivity, use of another performance indicator for OD matrix comparison is recommended. The mean normalized Levenshtein distance as proposed by (Behara et al., 2020) seems a promising alternative.

CRediT authorship contribution statement

Luuk Brederode: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Adam Pel:** Conceptualization, Supervision, Writing – review & editing. **Luc Wismans:** Supervision, Writing – review & editing. **Bernike Rijksen:** Investigation. **Serge Hoogendoorn:** Supervision, Project administration.

Acknowledgments

The application on the large network in Section 5 was partially funded by the province of Noord Brabant.

Appendix A. Convexity

A.1. Convexity and uniqueness of first part of the objective function

To consider the first part of the objective function we look at problem (23) with $w_1 = 1$, $w_2 = 0$, $w_3 = 0$. To prove convexity of the first part of the objective function, we look at the Hessian (second partial derivatives to the OD demands), which is given by:

$$\frac{\partial}{\partial D_{rs'}}\frac{\partial f_1}{\partial D_{rs}} = \frac{\partial^2\left(\sum_{rs\in RS}\left(D_{rs}-D^0_{rs}\right)\right)}{\partial D_{rs}\partial D_{rs'}} = \begin{cases} 0 & \forall rs' \neq rs \\ 2 & \forall rs' = rs \end{cases} \forall rs', rs \in RS \tag{46}$$

Hence, in this case, the Hessian matrix $|RS|x|RS|$ has value two on all elements of its diagonal and zeroes in all other cells, which means it is positive definite. Therefore, the first part of the objective function is strictly convex and since all constraints are linear inequalities, and as such form a closed convex set, problem (23) always has a unique solution when $w_1 = 1$, $w_2 = 0$, $w_3 = 0$).

A.2. Convexity of second part of objective function

To consider the second part of the objective function we look at problem (23) with $w_1 = 0, w_2 = 1, w_3 = 0$. To prove convexity of the second part of the objective function, we look at the first order Taylor approximations of the link flows $y_l(\mathbf{D})$ which are linear functions of the form:

$$y_l(\mathbf{D}) = c_l + b_l^T D \text{ with } c_l \in \mathrm{R},\ b_l \in \mathrm{R}^{|RS|}. \tag{47}$$

As such, $(y_l(\mathbf{D}) - \widetilde{y}_l)^2$ is a quadratic function:

$$(y_l(\mathbf{D}) - \widetilde{y}_l)^2 = \left(b_l^T D\right)^2 - 2b_l^T D(\widetilde{y}_l - c_l)^2 \tag{48}$$

With corresponding Hessian matrix $\nabla^2(b_l^T D)^2 = 2b_l b_l^T$ which is positive semidefinite. Indeed:

$$D^T b_l b_l^T D = \left(b_l^T D\right)^2 \geq 0\ \forall D \in \mathrm{R}^{|RS|}, \tag{49}$$

which means that the second part of the objective function in (23) is convex. This means that there is no unique solution for problem (23) when $w_1 = 0$, $w_2 = 1$, $w_3 = 0$, but all local solutions are global minimizers.

A.3. Convexity of third part of objective function

To consider the third part of the objective function we look at problem (23) with $w_1 = 0, w_2 = 0, w_3 = 1$. To prove convexity of the third part of the objective function, we look at the first order Taylor approximations of the queuing delays on route level τ_p. Using the same reasoning as A.2 it can be proven that its corresponding Hessian matrix is positive semidefinite which means that also the third part of the objective function in (23) is convex.

Appendix B. Discontinuities in flow acceptance factor function

Because the node model adheres to strict link capacity constraints, a discontinuity in the relationship between OD demand and flow acceptance factors occurs whenever a change in demand causes a bottleneck to switch from an inactive to an active state or vice versa. To illustrate this, consider the corridor network displayed in Fig. 9 (top) where $C_1 = 3000$, $C_2 = 2000$ and $C_3 = 1000$ and the corresponding relations between demand on rs and flow acceptance factors (bottom left) and between demand on rs and link flows (bottom right). It is clearly visible that functions α_{23} and y_3 are discontinuous at $D_{rs} = 1000$ (link 3 switches state) whereas functions α_{12} and y_2 are discontinuous at $D_{rs} = 2000$ (link 2 switches state). The switch of state of link 2 at $D_{rs} = 2000$ causes a second discontinuity of $\alpha_{23}(D_{rs})$ due to the active bottleneck upstream, but the route based flow acceptance factor at link 3 ($\widehat{\alpha}_p^3 = \alpha_{12}\alpha_{23}$) and thus y_3 do not have such a discontinuity.

From Fig. 9 (right), it becomes apparent that elements in response function (4) become unresponsive to changes in demand whenever one or more upstream bottlenecks are active on the considered OD pair. This means that, in the upper level, these OD pairs cannot be used to directly influence flows downstream links, and as such become irrelevant. We therefore look at the sensitivity of link flows for different demand intervals in Table 3 and conclude that link 3 can only be influenced when $D_{rs} \leq 1000$, whereas link 2 can only be influenced when $D_{rs} \leq 2000$ and link 1 remains sensitive as long as $D_{rs} \leq 3000$.

Fig. 9 (left) illustrates approximation for $\partial\alpha_{23}/\partial D_{rs}$ evaluated in point $D_{rs} = 1500$. Notice that because in this example and case $\alpha_{12} = 1$, the turn demand T_{23} is equal to OD demand D_{rs}, so translation from turn to route and OD level (Section 3.2.3) is omitted in this example. Further note that the results in this example seem trivial, as they can be deduced by simply analysing the network, but this example is given as an introduction to the solution scheme described in Section 3.2. A more complex case based on the numerical example described in (Tampère et al., 2011) is given in (Brederode et al., 2014).

Appendix C. Approximated sensitivities on turn level breaking convergence

This appendix describes the mechanism that causes discontinuities in $\boldsymbol{\alpha}_n(\mathbf{T}_n)$ to be crossed during application of the proposed solution method. This may happen due to difference between the true function within the node model (8) and its linear approximate derived by finite differences (see Section 3.2.2) used in simplified optimization problem (23) applied in the upper level.

Continuing the numeric example from Appendix B, but now the domain of interest is limited to $D_{rs} \leq C_2$. In this case $\alpha_{23}(\mathbf{T}_n) = \alpha_{23}(D_{rs})$ and the link state constraint (as defined in (18)) on link 3 may be written as:

$$\chi_3(D_{rs} - \delta_3 C_3) \leq 0. \tag{50}$$

The two graphs in Fig. 10 display $\alpha_{23}(D_{rs})$ for the case where link 3 is not constraining (left; $D_{rs} = 900$) and constraining (right; $D_{rs} = 1100$), along with its linear sensitivity-approximations $\frac{\partial\alpha_{23}}{\partial D_{rs}}|_{500}$ and $\frac{\partial\alpha_{23}}{\partial D_{rs}}|_{1500}$ and link state constraints assuming $\delta_3 = 0.95$ (left) and $\delta_3 = 1.05$ (right).

Consider a situation where this corridor network is just a part of a general network where demand on other OD pairs in the network has influence on the distribution of available supply of the outlinks of the node between links 2 and 3. Assume that the upper level

solver changes the OD demand matrix such that the distribution of supply on the considered node in altered reducing the effective capacity of the turn from link 2 to link 3 to 750 pch/h. As displayed in Fig. 11, this yields a shifted $\alpha_{23}(D_{rs})$ and different sensitivities, whereas the approximated sensitivities used in the simplified optimization problem are not updated. This means that in both cases the link state constraint is still satisfied, but

1. In the non-constraining case (Fig. 11, left), the upper level uses an approximated sensitivity of zero, whereas it has become negative. This occurs in 6 runs of test case [+LS+S+QD] and breaks convergence. In the test case applications in Section 4.4 this is prevented by lowering δ_j to 0.9 on the considered outlinks. In this theoretical example δ_j should be lowered to 0.75 or lower;
2. In the constraining case (Fig. 11, right), the upper level assumes a slightly more negative sensitivity than its non-approximated counterpart. This probably occurs in some runs in some test cases in Section 4.4, but does not break convergence, as the gradient information correctly assumes the sensitivity to remain smaller than zero (albeit that the approximate value is slightly off).

Now assume that the upper level solver changes the OD demand matrix such that the distribution of supply on the considered node in altered increasing the effective capacity of the turn from link 2 to link 3 to 1250 pch/h.[9] As in cases 1 and 2, this yields a shifted $\alpha_{23}(D_{rs})$ whilst still satisfying the link state constraint, but

1. In the non-constraining case (Fig. 12, left), the approximated sensitivity of zero used in the upper level remains correct. This probably occurs in some runs in some test cases in Section 4.4, but it does not break convergence.
2. In the constraining case (Fig. 12, right), the upper level assumes a negative sensitivity, whereas it has become zero. This apparently does not occur in the test cases in Section 4.4, or at least not to the extent that it breaks convergence. However, this mechanism is likely to break convergence.

Currently, it is unknown why the first case breaks convergence in the test case applications whereas the fourth test case does not (or does not occur). This is left for further research.

References

Zijpp, Van der, N.J., 1996. Dynamic Origin-Destination Matrix Estimation on Motorway Networks. Delft. https://repository.tudelft.nl/islandora/object/uuid:08b4b2b9-f9b0-47ee-b565-30082779ddbc?collection=research.

Abrahamsson, T., 1998. Estimation of origin-destination matrices using traffic counts-a literature survey.

Alpcan, T., 2013. A framework for optimization under limited information. J. Glob. Optim. 55, 681–706. https://doi.org/10.1007/s10898-012-9942-z.

Antoniou, C., Azevedo, C.L., Lu, L., Pereira, F., Ben-Akiva, M., 2015. W–SPSA in practice: approximation of weight matrices and calibration of traffic simulation models. Transp. Res. Procedia 7, 233–253. https://doi.org/10.1016/j.trpro.2015.06.013.

Bakker, D., Mijjer, P.H., Hofman, F., 1994. QBLOK: an assignment technique for modelling the dependency between bottlenecks and the prediction of grid lock. In: Proceedings of Colloquium Vervoersplanologisch Speurwerk. Delft, pp. 313–332.

Behara, K.N.S., Bhaskar, A., Chung, E., 2020. A novel approach for the structural comparison of origin-destination matrices: Levenshtein distance. Transp. Res. C Emerg. Technol. 111, 513–530. https://doi.org/10.1016/j.trc.2020.01.005.

Bell, M.G.H., 1995. Stochastic user equilibrium assignment in networks with queues. Transp. Res. B Methodol. 29, 125–137. https://doi.org/10.1016/0191-2615(94)00030-4.

Ben-Akiva, M.E., Ramming, S., 1998. Lecture notes: discrete choice models of traveler behavior in networks.

Bezembinder, E.M., Brandt, F., 2016. Junction Modelling in OmniTRANS.

Bifulco, G., Chrisalli, U., 1998. Stochastic user equilibrium and link capacity constraints: formulation and theoretical evidences. In: Paper Presented at the Proceedings of the European Transport Conference. Loughborough, UK, pp. 85–96. Proceedings of the European Transport Conference. Presented at the European transport conference, Loughborough, UK, pp. 85–96.

Bliemer, M.C.J., Raadsen, M.P.H., 2020. Static traffic assignment with residual queues and spillback. Transportation Research Part B: Methodological 132 (23rd International Symposium on Transportation and Traffic Theory (ISTTT 23)), 303–319. https://doi.org/10.1016/j.trb.2019.02.010.

Bliemer, M.C.J., Raadsen, M.P.H., Brederode, L.J.N., Bell, M.G.H., Wismans, L.J.J., Smith, M.J., 2017. Genetics of traffic assignment models for strategic transport planning. Transp. Rev. 37, 56–78. https://doi.org/10.1080/01441647.2016.1207211.

Bliemer, M.C.J., Raadsen, M.P.H., De Romph, E., Smits, E.-S., 2013. Requirements for traffic assignment models for strategic transport planning: a critical assessment. In: Paper Presented at: Proceedings of the 36th Australasian Transport Research Forum 2013, ATRF, Brisbane, Australia, 2-4 October 2013. Australasian Transport Research Forum.

Bliemer, M.C.J., Raadsen, M.P.H., Smits, E.-S., Zhou, B., Bell, M.G.H., 2014. Quasi-dynamic traffic assignment with residual point queues incorporating a first order node model. Transp. Res. B Methodol. 68, 363–384.

Boyce, D., Ralevic-Dekic, B., Bar-Gera, H., 2004. Convergence of traffic assignments: how much is enough? J. Transp. Eng. 130, 49–55. https://doi.org/10.1061/(ASCE)0733-947X(2004)130:1(49).

Brederode, L., Heynicks, M., Koopal, R., 2016. Quasi dynamic assignment on the large scale congested network of Noord-Brabant. In: Presented at the European Transport Conference, AET 2016 and Contributors. Barcelona, p. 17.

Brederode, L., Pel, A., Wismans, L., de Romph, E., Hoogendoorn, S., 2019. Static traffic assignment with queuing: model properties and applications. Transp. Transp. Sci. 15, 179–214. https://doi.org/10.1080/23249935.2018.1453561.

Brederode, L., Verlinden, K., 2019. Travel demand matrix estimation methods integrating the full richness of observed traffic flow data from congested networks. Transp. Res. In: Procedia, Modeling and Assessing Future Mobility Scenarios Selected Proceedings of the 46th European Transport Conference 2018, ETC 2018, pp. 19–31. https://doi.org/10.1016/j.trpro.2019.12.003, 42.

[9] This is not possible in the corridor network, because the turn capacity would be higher than the capacity of its outlink, but in a general network, the effective turn capacity can increase due to a change in the distribution of supply on a node, which is the mechanism that this example illustrates.

Brederode, L.J.N., Hofman, F., van Grol, R., 2017. Testing of a demand matrix estimation method Incorporating observed speeds and congestion patterns on the Dutch strategic model system using an assignment model with hard capacity constraints. In: Presented at the European Transport Conference, AET 2017 and contributors.
Brederode, L.J.N., Pel, A.J., Hoogendoorn, S.P., 2014. Matrix estimation for static traffic assignment models with queuing. In: HEART 2014 - 3rd Symposium of the European Association for Research in Transportation. Leeds UK.
Bundschuh, M., Vortisch, P., Van Vuuren, T., 2006. Modelling queues in static traffic assignment. In: Proceedings of the European Transport Conference, p. 2006.
Cantelmo, G., Viti, F., Cipriani, E., Nigro, M., 2017. A utility-based dynamic demand estimation model that explicitly accounts for activity scheduling and duration. In: Papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA 24-26 July 2017, pp. 440–459. https://doi.org/10.1016/j.trpro.2017.05.025, 23.
Cascetta, E., 2009. Transportation Systems Analysis. Optimization and Its Applications. Springer US, Boston, MA.
Cascetta, E., Nuzzolo, A., Russo, F., Vitetta, A., 1996. A modified logit route choice model overcoming path overlapping problems. Specification and some calibration results for interurban networks. Presented at the transportation and traffic theory. In: Proceedings of the 13th International Symposium on Transportation and Traffic Theory, Lyon, France, 24-26 July 1996.
Castiglione, M., Cantelmo, G., Qurashi, M., Nigro, M., Antoniou, C., 2021. Assignment matrix free algorithms for on-line estimation of dynamic origin-destination matrices. Frontiers in Future Transportation 2. https://doi.org/10.3389/ffutr.2021.640570.
Chakirov, A., Fourie, P.J., 2014. Enriched Sioux Falls scenario with dynamic and disaggregate demand 39 p. 10.3929/ETHZ-B-000080996.
Chu, C., 1989. A paired combinatorial logit model for travel demand analysis. In: Presented at the Transport Policy, Management & Technology Towards 2001: Selected Proceedings of the Fifth World Conference on Transport Research.
Cipriani, E., Gemma, A., Nigro, M., 2013. A bi-level gradient approximation method for dynamic traffic demand estimation: sensitivity analysis and adaptive approach. In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). Presented at the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pp. 2100–2105. https://doi.org/10.1109/ITSC.2013.6728539.
Cipriani, E., Florian, M., Mahut, M., Nigro, M., 2011. A gradient approximation approach for adjusting temporal origin–destination matrices. Transp. Res. C Emerg. Technol. 19, 270–282. https://doi.org/10.1016/j.trc.2010.05.013.
Correa, J.R., Schulz, A.S., Stier-Moses, N.E., 2004. Selfish routing in capacitated networks. Math. Oper. Res. 29, 961–976. https://doi.org/10.1287/moor.1040.0098.
Djukic, T., Hoogendoorn, S., Van Lint, H., 2013. Reliability assessment of dynamic OD estimation methods based on structural similarity index. In: Presented at the Transportation Research Board 92nd Annual Meeting Transportation Research Board.
Djukic, T., Masip, D., Breen, M., Perarnau, J., Casas, J., 2017. Modified bi-level framework for dynamic OD demand estimation in the congested networks. In: Presented at the 97th Annual Meeting of the Transportation Research Board. Washington, D.C.
Fafieanie, M.E., 2009. Calibrating Route Set Generation by Map Matching GPS Data. Twente University, Enschede. https://essay.utwente.nl/59341/1/scriptie_M_Fafieanie.pdf.
Fiorenzo-Catalano, M.S., 2007. Choice Set Generation in Multi-Modal Transportation Networks. TRAIL. https://repository.tudelft.nl/islandora/object/uuid:ef3b9c22-b979-4f46-9b02-110c82d67535?collection=research.
Fisk, C., 1980. Some developments in equilibrium traffic assignment. Transp. Res. B Methodol. 14, 243–255. https://doi.org/10.1016/0191-2615(80)90004-1.
Flötteröd, G., Rohde, J., 2011. Operational macroscopic modeling of complex urban road intersections. Transp. Res. B Methodol. 45, 903–922. https://doi.org/10.1016/j.trb.2011.04.001.
Frederix, R., 2012. Dynamic Origin-Destination Matrix Estimation in Large-Scale Congested Networks (Schatting van dynamische herkomst-bestemmingsmatrices in grootschalige, congestiegevoelige netwerken).
Frederix, R., Viti, F., Tampère, C.M.J., 2013. Dynamic origin–destination estimation in congested networks: theoretical findings and implications in practice. Transp. Transp. Sci. 9, 494–513. https://doi.org/10.1080/18128602.2011.619587.
Han, K., Friesz, T.L., Szeto, W.Y., Liu, H., 2015. Elastic demand dynamic network user equilibrium: formulation, existence and computation. Transportation Research Part B: Methodological 81 (1), 183–209. https://doi.org/10.1016/j.trb.2015.07.008.
Heynicks, M., Koopal, R., Zantema, K., 2016. The approach of traffic modelling in Noord-Brabant. In: Presented at the European Transport Conference. Barcelona.
Huang, W., Xu, G., Lo, H.K., 2020. Pareto-optimal sustainable transportation network design under spatial queuing. Netw. Spat. Econ. 20, 637–673. https://doi.org/10.1007/s11067-020-09494-6.
Köhler, E., Strehler, M., 2012. Combining static and dynamic models for traffic signal optimization inherent load-dependent travel times in a cyclically time-expanded network model. Procedia Soc. Behav. Sci. 54, 1125–1134.
Kotz, S., Kozubowski, T., Podgorski, K., 2001. The Laplace Distribution and Generalizations: a Revisit with Applications to Communications, Economics, Engineering, and Finance. Birkhäuser, Basel. https://link.springer.com/book/10.1007/978-1-4612-0173-1.
Lam, W.H.K., Zhang, Y., 2000. Capacity-constrained traffic assignment in networks with residual queues. J. Transp. Eng. 126, 121–128. https://doi.org/10.1061/(ASCE)0733-947X(2000)126:2(121).
Larsson, T., Patriksson, M., 1999. Side constrained traffic equilibrium models—analysis, computation and applications. Transp. Res. B Methodol. 33, 233–264. https://doi.org/10.1016/S0191-2615(98)00024-1.
Liu, H.X., He, X., He, B., 2009. Method of Successive Weighted Averages (MSWA) and self-regulated averaging schemes for solving stochastic user equilibrium problem. Netw. Spat. Econ. 9, 485–503. https://doi.org/10.1007/s11067-007-9023-x.
Ma, W., Pi, X., Qian, S., 2020. Estimating multi-class dynamic origin-destination demand through a forward-backward algorithm on computational graphs. Transp. Res. C Emerg. Technol. 119, 102747 https://doi.org/10.1016/j.trc.2020.102747.
Maher, M.J., Zhang, X., Vliet, D.V., 2001. A bi-level programming approach for trip matrix estimation and traffic control problems with stochastic user equilibrium link flows. Transp. Res. B Methodol. 35, 23–40. https://doi.org/10.1016/S0191-2615(00)00017-5.
Marzano, V., Papola, A., Simonelli, F., Papageorgiou, M., 2018. A Kalman filter for quasi-dynamic OD flow estimation/updating. IEEE Trans. Intell. Transp. Syst. 19, 3604–3612. https://doi.org/10.1109/TITS.2018.2865610.
Murty, K.G., 1991. Linear Programming. Wiley.
Nie, Y., Zhang, H.M., Lee, D.-H., 2004. Models and algorithms for the traffic assignment problem with link capacity constraints. Transp. Res. B Methodol. 38, 285–312. https://doi.org/10.1016/S0191-2615(03)00010-9.
Osorio, C., 2019a. Dynamic origin-destination matrix calibration for large-scale network simulators. Transp. Res. C Emerg. Technol. 98, 186–206. https://doi.org/10.1016/j.trc.2018.09.023.
Osorio, C., 2019b. High-dimensional offline origin-destination (OD) demand calibration for stochastic traffic simulators of large-scale road networks. Transp. Res. B Methodol. 124, 18–43. https://doi.org/10.1016/j.trb.2019.01.005.
Patil, P.N., Ross, K.C., Boyles, S.D., 2021. Convergence behavior for traffic assignment characterization metrics. Transp. Transp. Sci. 17, 1244–1271. https://doi.org/10.1080/23249935.2020.1857883.
Qurashi, M., Ma, T., Chaniotakis, E., Antoniou, C., 2020. PC–SPSA: employing dimensionality reduction to limit SPSA search noise in DTA model calibration. IEEE Trans. Intell. Transp. Syst. 21, 1635–1645. https://doi.org/10.1109/TITS.2019.2915273.
Raadsen, M., Bliemer, M., 2018. General solution scheme for the static link transmission model.
Ros-Roca, X., Montero Mercadé, L., Barceló Bugeda, J., 2018. Notes on the measure of the structural similarity of OD matrices.
Shafiei, S., Saberi, M., Zockaie, A., Sarvi, M., 2017. Sensitivity-based linear approximation method to estimate time-dependent origin–destination demand in congested networks. Transp. Res. Rec. J. Transp. Res. Board 2669, 72–79. https://doi.org/10.3141/2669-08.
Smith, M., 2012. Traffic control and route choice: modelling and optimisation, in: JCT Symposium, University of Warwick (September 21, 2012).
Smith, M.J., 2013. A link-based elastic demand equilibrium model with capacity constraints and queueing delays. Transp. Res. C Emerg. Technol. 29, 131–147. https://doi.org/10.1016/j.trc.2012.04.011.

Smith, M.J., 1987. Traffic control and traffic assignment in a signal-controlled network with queueing. In: Presented at the 10th International Symposium on Transportation and Traffic Theory (ISTTT), Boston, MA.
Smits, E.-S., Bliemer, M.C.J., Pel, A.J., van Arem, B., 2015. A family of macroscopic node models. Transp. Res. B Methodol. 74, 20–39. https://doi.org/10.1016/j.trb.2015.01.002.
Tajtehranifard, H., 2017. Incident Duration Modelling and System Optimal Traffic Re-Routing. Queensland University of Technology. https://eprints.qut.edu.au/110525.
Tampère, C.M.J., Corthout, R., Cattrysse, R., Immers, L.H., 2011. A generic class of first order node models for dynamic macroscopic simulation of traffic flows. Transp. Res. B Methodol. 45, 289–309. https://doi.org/10.1016/j.trb.2010.06.004.
Toledo, T., Kolechkina, T., 2013. Estimation of dynamic origin–destination matrices using linear assignment matrix approximations. IEEE Trans. Intell. Transp. Syst. 14, 618–626.
Toledo, T., Kolechkina, T., Wagner, P., Ciuffo, B., Azevedo, C., Marzano, V., Flötteröd, G., 2015. Network model calibration studies. In: Daamen, W., Buisson, C., Hoogendoorn, S.P. (Eds.), Traffic Simulation and Data: Validation Methods and Applications. CRC Press Taylor & Francis London, pp. 141–162.
Transportation Networks for Research Core Team, 2019. Transportation Networks for Research. [WWW Document]. URL https://github.com/bstabler/TransportationNetworks (accessed 8.26.19).
Tsanakas, N., Ekström, J., Olstam, J., 2020. Estimating emissions from static traffic models: problems and solutions. J. Adv. Transp. 2020, 1–17. https://doi.org/10.1155/2020/5401792.
Tympakianaki, A., Koutsopoulos, H.N., Jenelius, E., 2015. c-SPSA: cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin–destination matrix estimation. Transp. Res. C Emerg. Technol. 55, 231–245. https://doi.org/10.1016/j.trc.2015.01.016.
Waltz, R.A., Morales, J.L., Nocedal, J., Orban, D., 2006. An interior algorithm for nonlinear optimization that combines line search and trust region steps. Math. Program. 107, 391–408. https://doi.org/10.1007/s10107-004-0560-5.
Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: a forward and backward propagation algorithmic framework on a layered computational graph. Transp. Res. C Emerg. Technol. 96, 321–346. https://doi.org/10.1016/j.trc.2018.09.021.
Yang, H., 1995. Heuristic algorithms for the bilevel origin-destination matrix estimation problem. Transp. Res. B Methodol. 29, 231–242.
Yang, H., Yagar, S., 1994. Traffic assignment and traffic control in general freeway-arterial corridor systems. Transp. Res. B Methodol. 28, 463–486. https://doi.org/10.1016/0191-2615(94)90015-9.