Static traffic assignment with residual queues and spillback

Michiel C.J. Bliemer, University of Sydney / ETH Zurich
Mark P.H. Raadsen, University of Sydney

Conference paper STRC 2017

STRC

17th Swiss Transport Research Conference
Monte Verità / Ascona, May 17-19 2017

Static traffic assignment with residual queues and spillback

Michiel C.J. Bliemer
University of Sydney
Sydney

Phone: +61291141840
Fax: +61291141863
email: michiel.bliemer@sydney.edu.au

Mark P.H. Raadsen
University of Sydney
Sydney

Phone: +61291141891
Fax: +61291141863
email: mark.raadsen@sydney.edu.au

April 2017

Abstract

In this paper we mathematically derive a static version of a first order dynamic traffic assignment (TA) model by determining average inflow and outflow rates during a given time period under simplified temporal assumptions. The state-of-the-art dynamic TA model that we use as a starting point consists of a dynamic route choice component and a dynamic network loading component described by the continuous-time general link transmission model. The resulting static TA model adopts the same general formulation supporting any concave fundamental diagram and first order node model used in the dynamic model, but it does not need to explicitly describe time. The node model imposes hard capacity constraints on flows that may lead to queues, and finite storage constraints are imposed by the fundamental diagram that may result in spillback. As far as we are aware, this is the first analytical static model that is capable of describing residual queues and spillback in general transport networks. The model is formulated as a path-based variational inequality problem where network loading is embedded via a fixed point problem formulation. We describe an algorithm that can solve this problem, and illustrate feasibility by presenting numerical results on some example networks.

Keywords

Static traffic assignment – Residual queues – Spillback – Concave fundamental diagram – General link transmission model

1. Introduction

Static traffic assignment (TA) models are widely applied for strategic transport planning purposes. Traditional static TA models are referred to as capacity restrained, since they do not include hard capacity constraints on link flows but rather include link performance functions such that travel times simply increase rapidly when flows exceed capacity. It is well-known that in (heavily) congested networks, such an approach will lead to unrealistic networks flows and travel times.

There is a growing literature in advancing the state-of-the-art in static TA models, in particular by including *capacity constraints* such that link flows no longer exceeds physical link capacities. There exist two main approaches in order to achieve this (for a detailed overview, we refer to Bliemer et al., 2014). In the first approach, side constraints are added to the mathematical optimisation problem such that link flows (where no distinction is made between inflows and outflows) are smaller than or equal to link capacities. The resulting optimisation problem is typically solved using Lagrangian methods in which Lagrange multipliers are interpreted as additional queuing delays. The model implicitly assumes that there is sufficient network capacity such that flow can be rerouted without violating the capacity constraints. Therefore, this approach does not explicitly describe queues. Examples of this approach are Smith (1987), Yang and Yagar (1994, 1995), Larsson and Patriksson (1995) and Bell (1995). The second approach explicitly considers residual queues in which link inflows may exceed link outflows. Distinguishing between link inflows and link outflows is an important step away from traditional static TA models. It more closely resembles reality as well as allowing more flexibility in the model formulation. Examples of this approach are Bifulco and Crisalli (1998), Lam and Zhang (2000), Smith (2013), Smith et al. (2013), and Bliemer et al. (2014). These models are often also referred to as 'quasi-dynamic' TA models, since these static models produce outputs more similar to dynamic models with respect to the representation of queues and their approach in computing travel times. We believe that this term is somewhat confusing, since it is not the dynamic nature of models that allows the representation of queues (see our discussion later in this section). Therefore, in this paper we refer to such models as static models that account for capacity (and possibly storage) constraints.

Besides these analytical models there also exist software implementations that describe operational procedures that aim to better describe queues in static models, for example Bakker et al. (1994) and Bundshuh et al. (2006). Further, there also exist semi-dynamic approaches, such as Van Vliet (1982), and hybrid approaches that combine static and dynamic models, such as Bliemer et al. (2012). In this paper we restrict ourselves to analytical static models.

In general, two different ways of explicitly modelling (residual) queues exist. The more realistic assumption is to consider horizontal queues, also referred to as physical queues, which occupy space and therefore there is a limited storage capacity of vehicles on the link. When this storage capacity is exceeded, queues spill back onto upstream links. Alternatively, a more simplified assumption is to consider vertical queues that do not have a physical length, also referred to as point queues, which means that no such storage constraints exist and therefore not spillback occurs. In this paper we explicitly consider storage constraints our model formulation to account for spillback effects. While horizontal queues and storage constraints are fairly common in dynamic TA models, as far as we are aware, there exist no (analytical) static TA models that can account for storage constraints in general networks in order to describe spillback effects. One of the few studies that demonstrates how spillback could potentially be included is presented in Smith et al. (2013). They consider storage constraints in addition to capacity constraints, although their analysis is restricted to a five-link network and they focus mainly on the existence of an equilibrium in this simplified network. Adding spillback to a static model is not trivial and it is often (incorrectly) thought of as a feature that can only be described by a dynamic model.

In this paper we propose a static TA model that includes both capacity constraints as well as storage constraints on general transport networks and is therefore capable of describing residual queues and spillback. We could follow two different routes in developing such a new model, namely (i) extend an existing simpler model (i.e., the traditional static assignment model), or (ii) simplify an existing more advanced model (i.e., a first order dynamic TA model). All previous attempts have followed the first route, and it turns out to be very difficult to find the appropriate extension. In this paper we follow the second route and start with a model that already contains the desirable properties and create a simplified static version. This idea follows from Bliemer et al. (2017) who provide a qualitative assessment of TA models and use an analogy to genetics in biology to classify them. They distinguish three 'genes', namely spatial assumptions, temporal assumptions, and behavioural assumptions, each with their own 'nucleotides' that describe elemental characteristics that influence the functionality of the model. They conjecture that it is possible to use a dynamic model that is capable of describing residual queues and spillback as a starting point, and mathematically derive a consistent static version. In this paper we actually conduct such a mathematical derivation for the first time.

Table 1 lists the three main assumptions and the elemental characteristics as described in Bliemer et al. (2017). Assumptions in static models are typically only implicitly made, but it is important to make them explicit in order to understand their limitations. We first observe from Table 1 that the assumptions in the state-of-the-art first order dynamic TA model as described under model M1 are very different from the assumptions in traditional static capacity restrained models as described under model M3. Temporal assumptions (T.1)–(T.3) need to be different since these determine whether a model is static or dynamic (or semi-dynamic). Static models assume infinite vehicle propagation speed and wave speeds[1] in the hypocritical (uncongested) branch of the fundamental diagram. Static models also assume that residual traffic does not transfer from period to period.

Spatial assumptions (S.1)–(S.4) determine whether a model is capable of describing residual queues and spillback. Model M3 does not consider any capacity nor storage constraints and as such is not capable of representing queues (and spillback). The main point that Table 1 makes is that it is not necessary to make simplifying spatial assumptions in a static model as evidenced by model M2. This model makes exactly the same spatial assumptions as first order dynamic model M1, while only simplifying temporal assumptions. To be precise, model M2 includes the same concave fundamental diagram (that also describes bounds on capacity and queue storage) and the same node model as model M1. An important distinction between M2 and model M1 is its assumption of infinite backward wave speeds, this caters for spillback support while remaining a static model. Note that the vehicle propagation speeds in the hypercritical (congested) branch are set to vehicular speeds to support residual queues, this is discussed in detail in Section 4. In this paper we will formulate a model that is consistent with the assumptions in M2.

The main contribution of this paper is that we formulate a novel static TA model that can describe residual queues and spillback in general transport networks by deriving it from a first order dynamic TA model in which all spatial assumptions are maintained, including the fundamental diagram and node model. We show that the traditional capacity restrained TA model results as a special case if we remove capacity and storage constraints. Given that in particular the resulting network loading component is significantly more complex than in traditional static models, we explore the model structure and propose a solution algorithm making explicit use of forward and backward relationships between model variables. We demonstrate feasibility of the model on several small networks. While we believe that the novel formulation in this paper is a significant step forward in including spillback in static models, we consider it only as a first step in this direction in which we focus on model formulation and feasibility while we leave the investigation of solution properties and algorithm convergence for future research.

The paper outline is as follows. We start with an example in Section 2 to illustrate what the outcome of a static model with spillback could look like, as it requires quite a different way of thinking compared to traditional static models that most people are familiar with. In Section 3 a state-of-the-art first order dynamic TA model (conform M1) is described, which will be the starting point for our mathematical

[1] Wave speed denotes the speed at which flow state information travels across a link. In the hypocritical branch it travels downstream, while in congestion it travels in the upstream direction.

derivation. Then in Section 4 our novel static model (conform M2) is mathematically derived and presented as a main result. In particular our capacity and storage constrained network loading problem is significantly more complex than in other static models, while the route choice problem is not much different from other existing path-based static models. In Section 5 we explore the model structure and show that several existing models, including the traditional static model (conform M3), are special cases of our model. We propose a two-level algorithm in Section 6 that can find a solution to our network loading problem. In Section 7 we demonstrate our new static model by looking at several test networks. Finally, in Section 8 we provide a discussion on next steps to be taken in order to advance our novel static model towards a general tool for transport planning on large scale transport networks.

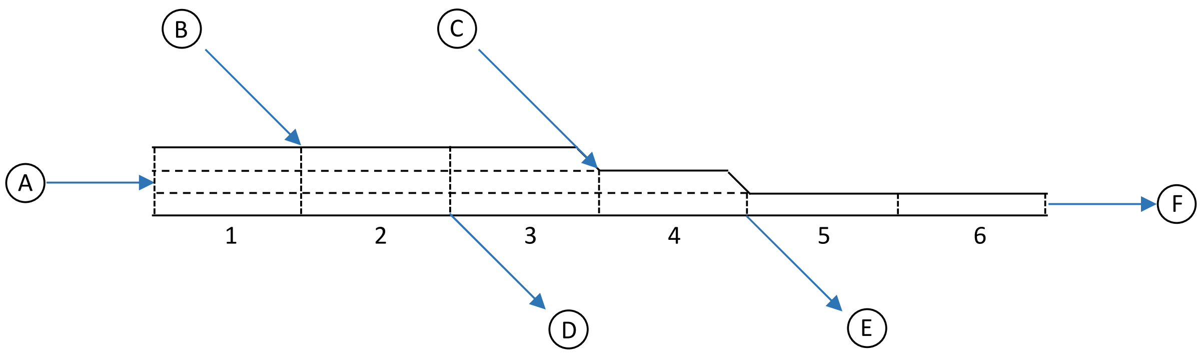
Table 1: Assumptions in different traffic assignment models

	M1	*M2*	*M3*
	Dynamic capacity and storage constrained	*Static capacity and storage constrained*	*Static capacity restrained*
Spatial assumptions			
(S.1) Shape fundamental diagram			
- Hypocritical branch	Concave	Concave	Concave
- Hypercritical branch	Concave	Concave	Not available
(S.2) Capacity constraints			
- Inflow	Constrained	Constrained	Unconstrained
- Outflow	Constrained	Constrained	Unconstrained
(S.3) Storage constraints	Constrained	Constrained	Unconstrained
(S.4) Turn flow restrictions	First order	First order	No restrictions
Temporal assumptions			
(T.1) Wave speeds			
- Hypocritical branch	Kinematic	Infinite	Infinite
- Hypercritical branch	Kinematic	Infinite	Not available
(T.2) Vehicle propagation speeds			
- Hypocritical branch	Vehicular	Infinite	Infinite
- Hypercritical branch	Vehicular	Vehicular	Not available
(T.3) Residual traffic transfer	Transfer	No Transfer	No transfer
Behavioural assumptions			
(B.1) Decision making			
- Rationality	Full	Full	Full
- Information	Imperfect	Imperfect	Perfect
(B.2) Travel time consideration	Experienced	Instantaneous	Instantaneous

2. Illustrative example

In order to demonstrate the main ideas behind the static model in this paper we consider a simple network as shown in Figure 2 in which we mainly focus on origin-destination (OD) pair (A,F). This network consists of 6 links, each having a length of 3 km. We assume a triangular fundamental diagram for each link with the following characteristics: (i) maximum speed is 90 km/h, (ii) capacity is 1800 veh/h per lane, and (iii) jam density is 180 veh/km per lane. The links only differ in the number of lanes; links 1, 2, and 3 have three lanes, link 4 has two lines, and links 5 and 6 have one lane. Therefore, links 4 and 5 are potential bottlenecks. Based on the mentioned link length and maximum speed, the free-flow link travel times are 2 minutes and the free-flow route travel time from A to F is 12 minutes.

Figure 1: Example corridor network



Consider a stationary travel demand from A to F of 4000 veh/h during one hour, and we assume that there is currently no travel demand on other OD pairs. Since the inflow of a two-lane road segment is limited to $2\times1800=3600$ vehicles and of a one-lane road to 1800 vehicles during this hour, queues will build up upstream the two bottlenecks. In Figure 2 we show the cumulative inflows and outflows over time consistent with simplified first order kinematic wave theory. We used the event-based algorithm proposed in Raadsen et al. (2016) to find an exact solution of the continuous-time link transmission model. After 6 minutes (360 s) a queue starts slowly building up at the end of link 3 (since the link inflow rate is 4000 veh/h while the outflow rate is 3600 veh/h), and after 8 minutes (480 s) a queue also starts rapidly building up at the end of link 4 (where the link inflow and outflow rates are 3600 and 1800 veh/h, respectively). After 24 minutes (1440 s) the queue on link 4 spills back onto link 3, which means that the queue on link 3 starts to grow more rapidly and spills back onto link 2 after 48.18 minutes (2891 s). The travel time of the first vehicle from A to F is 12 minutes (720 s), while the travel time of the last vehicle is 85.33 minutes ($8720-3600=5120$ s). Since the inflow rate is stationary at 4000 veh/h, and the outflow rate is stationary at 1800 veh/h, the average travel time from A to F is $\frac{1}{2}(12+85.33)=48.67$ minutes.

Instead of using a dynamic model, we would like to develop a static model that is able to describe average flow rates and route travel times similar to the dynamic model, based on the same capacity constraints and triangular fundamental diagram. Traditional capacity restrained models cannot describe residual queues and therefore are not suitable. Bliemer et al. (2014) proposed a static model with vertical residual queues that applies capacity constraints in the same way as the link transmission model through the application of a (first order) node model. Figure 3(a) shows the average link inflow rates and the residual queues at the end of the hour when we would apply this model. The first vehicle that departs does not experience any queues and therefore has a travel time of 12 minutes. At the end of the hour, the number of vehicles in a residual queue is 2200 in total, and with an outflow rate of 1800 veh/h it takes the last vehicle an additional 73.33 minutes to exit the network. Therefore, the average travel time from A to F is $12+\frac{1}{2}\cdot73.33=48.67$ minutes, the same as in the link transmission model. However, since the queues do not spill back upstream, a single vehicle that would travel from B to D would not encounter any queues, while a vehicle travelling from C to E would experience a queue that is too large.

Therefore, vertical queues can lead to significant under- and over-estimations of travel times on certain routes as a result of the lack of considering spillback effects.

Figure 2: Outcomes of the dynamic link transmission model on the corridor network

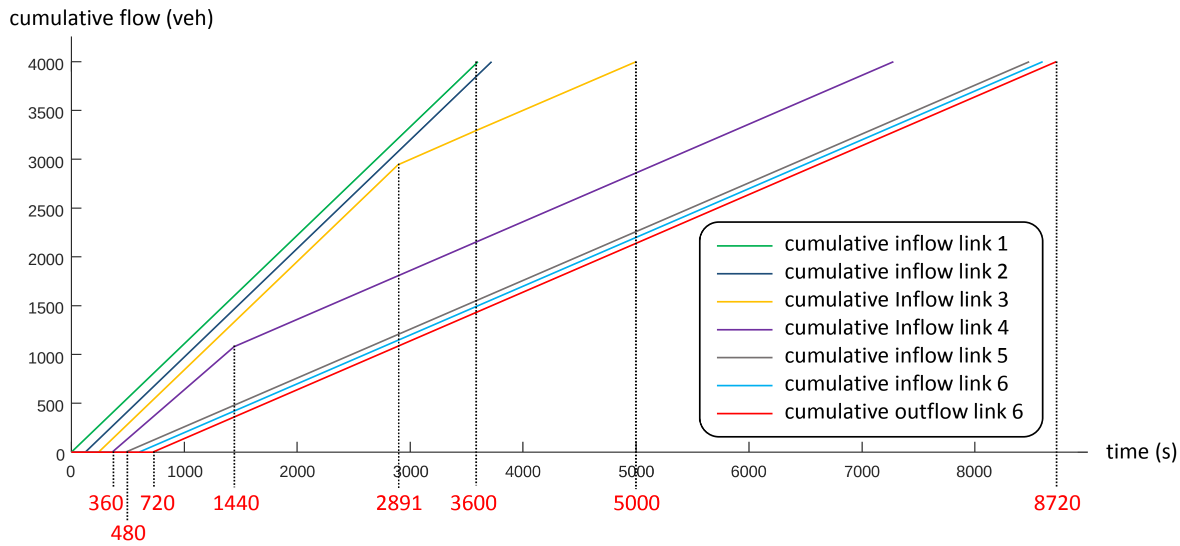
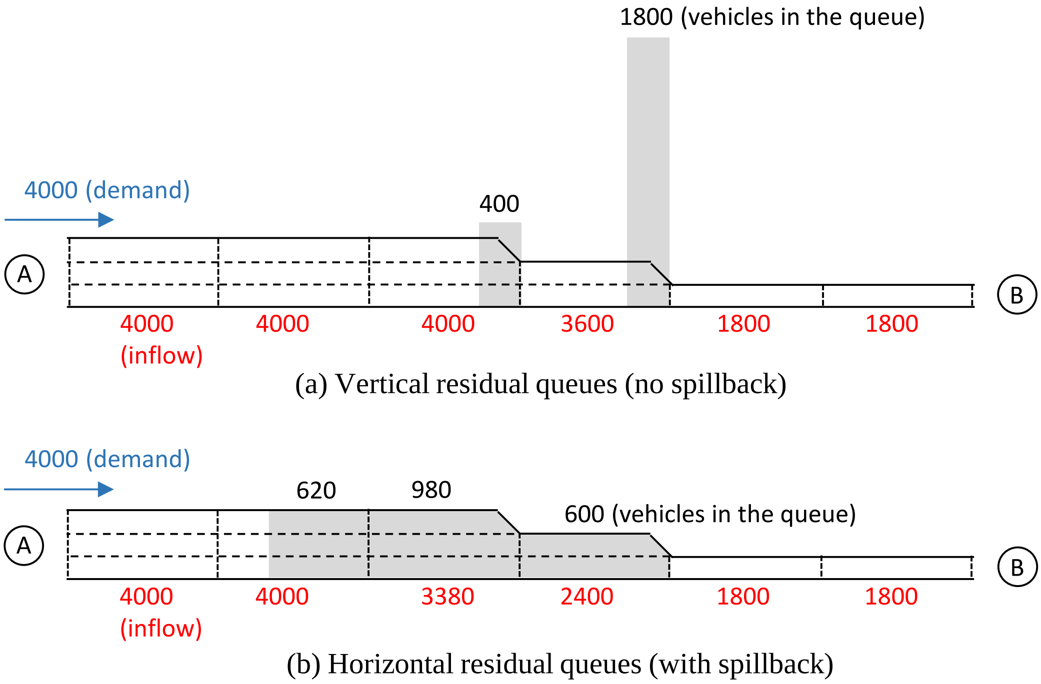


Figure 3: Residual queues in static model on the corridor network



In this paper we propose a static model with horizontal queues that will produce the outcome as presented in Figure 3(b). The same 2200 vehicles reside in a queue after one hour, however, the queues are now spilling back upstream and reach up to link 2. This is achieved by adjusting the total link inflow (or similarly, the average inflow rate) on links where spillback occurs. For example, link 4 has a total inflow of 2400 vehicles during one hour, which can be calculated as follows. For some period of time, the link is not yet in a spillback state and hence accepts an inflow rate of 3600 veh/h (the two-lane capacity). However, when spillback occurs, this rate drops to 1800 veh/h (the link outflow rate). In order to determine how long it takes for the queue to reach the beginning of the link, we need to look at the queueing density. With an outflow rate of 1800 veh/h, the triangular fundamental diagram will tell us that the queuing density is 200 veh/km, such that the link has a storage capacity of 600 vehicles (since the link is 3 km long). With an inflow rate of 3600 veh/h and an outflow rate of 1800 veh/h, this link reaches its storage capacity after 20 minutes. This means that the link accepts 3600 veh/h during 20 minutes, and 1800 veh/h during 40 minutes, hence on average 2400 vehicles during the hour. The average route travel time from A to F is again 48.67 minutes (since the same number of vehicles are in the residual queues and the exit rate out of the network remains 1800 veh/h). What is mainly different is the distribution of the queue over multiple links. Link 2 has no vehicles in a vertical queue, while it has 620 vehicles in the horizontal queue, hence a single vehicle from B to D will experience delays in case of horizontal queues. Further, link 4 has only 600 vehicles in the horizontal queue compared to 1800 vehicles in the vertical queue, hence a vehicle from C to E would experience much less delays in case of horizontal queues.

In the next section we first formulate a first order macroscopic dynamic TA model that serves as the starting point for our derivation of our static model. The resulting model no longer needs to take time explicitly into account while still allowing for residual queues and spillback.

3. First order macroscopic dynamic model

Let $\mathcal{G}=(\mathcal{N},\mathcal{A})$ be a given graph where $\mathcal{N}$ and $\mathcal{A}$ denote the set of nodes and directed links, respectively. Each node $n\in\mathcal{N}$ has a set of incoming links $\mathcal{A}_n^-$ and a set of outgoing links $\mathcal{A}_n^+$. Let $\mathcal{W}$ denote the set of origin-destination (OD) pairs, and let $\mathcal{I}$ denote the set of departure time intervals. Each departure time interval $i\in\mathcal{I}$ is defined by $[t^i,t^i+T^i)$, where T^i [h][2] indicates the length of the interval. For each OD relationship $w\in\mathcal{W}$ and each departure time interval $i\in\mathcal{I}$ we consider a given travel demand D^{wi} [veh]. Further, let $\mathcal{P}^w$ denote the set of relevant paths for OD pair *w*, and define $\mathcal{P}=\bigcup_w \mathcal{P}^w$.

Each link $a\in\mathcal{A}$ is described by a length L_a [km], maximum speed $\sigma_a^{\max}$ [km/h], capacity Q_a [veh/h], jam density K_a [veh/km], and critical density k_a^{crit} [veh/km]. Further, each link has an associated fundamental diagram described by $q=\Phi_a(k)$, where *q* is a flow rate [veh/h], *k* is a density [veh/km], and $\Phi_a(\cdot)$ is a two-regime concave flux function,

$$\Phi_a(k)=\begin{cases}\Phi_{I,a}(k), & \text{if } 0\le k\le k_a^{\text{crit}},\\ \Phi_{II,a}(k), & \text{if } k_a^{\text{crit}}\le k\le K_a,\end{cases}\qquad \forall a\in\mathcal{A}. \tag{1}$$

Function $\Phi_{I,a}(k)$ describes the strictly increasing hypocritical branch of the fundamental diagram, and $\Phi_{II,a}(k)$ describes the strictly decreasing hypercritical branch. Since these branches are strictly monotonous, we can define the inverse hypocritical and hypercritical flux functions $\Phi_{I,a}^{-1}(q)$ and $\Phi_{II,a}^{-1}(q)$. These inverse flux functions are useful in both dynamic and static models.

A macroscopic TA model consists of two components, namely (i) a route choice model, and (ii) a network loading model. The route choice model determines path flow rates based on perceived path travel times, while the network loading model simulates path flows on the network in order to determine traffic conditions on each link and resulting travel times.

3.1 Route choice model

Let $\mathbf{f}^i=[f_p^{wi}]_{w\in\mathcal{W},p\in\mathcal{P}^w}$ [veh/h] denote the vector of all average path flow rates during departure time interval *i*, and let $\mathbf{f}=[\mathbf{f}^i]_{i=1,\ldots,I}$. Let c_p^{wi} [h] denote the perceived average path travel times, which are assumed given by (see e.g. Chen, 1999)

$$c_p^{wi}=\tau_p^{wi}+\frac{1}{\mu}\log f_p^{wi},\qquad \forall p\in\mathcal{P}^w,\forall w\in\mathcal{W},\forall i\in\mathcal{I}, \tag{2}$$

where τ_p^{wi} [h] is the average experienced path travel time for vehicles that departed during time interval *i*, and where μ [-] is a positive parameter that indicates the level of perception error by travellers. Section 2.3 shows how to calculate average path travel times τ_p^{wi}, which depends on path flows **f**. Formulation (2) is consistent with logit-based route choice where μ represents the scale parameter, in other words it holds that

$$f_p^{wi}=\frac{\exp(\mu c_p^{wi})}{\sum_{p'\in\mathcal{P}^w}\exp(\mu c_{p'}^{wi})},\qquad \forall p\in\mathcal{P}^w,\forall w\in\mathcal{W},\forall i\in\mathcal{I}. \tag{3}$$

Conditional on path set $\mathcal{P}$, stochastic user equilibrium path flow rates $\overline{\mathbf{f}}$ can be determined by solving the following variational equality (VI) problem:

$$\sum_{i\in\mathcal{I}}\sum_{w\in\mathcal{W}}\sum_{p\in\mathcal{P}^w}c_p^{wi}\left(\overline{\mathbf{f}}\right)\left(f_p^{wi}-\overline{f}_p^{wi}\right)\ge 0,\quad \forall\mathbf{f}\in\Omega, \tag{4}$$

where the set of feasible path flow rates is given by $\Omega=\bigcup_{i=1,\ldots,I}\Omega^i$, with

[2] Since we believe it is important to understand the units of each variable, we indicate the unit when we first introduce a variable. Unit-less variables are indicated by [-].

$$\Omega^i = \left\{ \mathbf{f}^i \mid \sum_{p\in\mathcal{P}^w} f_p^{wi} = \frac{D^{wi}}{T_i}, \forall w \in \mathcal{W} ;\ f_p^{wi} \geq 0, \forall w \in \mathcal{W}, \forall p \in \mathcal{P}^w \right\}, \qquad \forall i \in \mathcal{I}. \tag{5}$$

3.2 Network loading model

3.2.1 Propagation of traffic state information

Let $u_a(t)$ [veh/h] and $v_a(t)$ [veh/h] denote the link inflow and outflow rates at time instant *t*. Then we can define the cumulative link inflows and outflows $U_a(t)$ [veh] and $V_a(t)$ [veh] at time instant *t* as

$$U_a(t) = \int_0^t u_a(\omega) d\omega, \quad \text{and} \quad V_a(t) = \int_0^t v_a(\omega) d\omega, \qquad \forall a \in \mathcal{A}. \tag{6}$$

Traffic state information travels upstream and downstream at kinematic wave speeds. We adopt the continuous-time formulation of the general link transmission model as described in Bliemer and Raadsen (2017) and Raadsen and Bliemer (2017). Let $\vec{u}_a(t)$ [veh/h] denote the potential outflow rate of out link *a* at time instant *t*, which will actually flow out if there are no outflow constraints. Similarly, let $\overleftarrow{v}_a(t)$ [veh/h] denote the potential inflow rate that link *a* can handle at time instant *t* if there is sufficient demand. The term 'potential' refers to the fact that these outflows and inflows will only be realised if there are no exit capacity or entry flow constraints, respectively.

We can identify three disjoint link states, namely (i) free-flow (in which both link boundaries are in a hypocritical state), (ii) spillback (in which both link boundaries are in a hypercritical state), or (iii) congestion (in which the upstream link boundary is in a hypocritical state while the downstream boundary is in a hypercritical state). If a link is in a free-flow state, then $\vec{u}_a(t)$ on the downstream link boundary is directly influenced by inflow rates $u_a(\cdot)$ some time earlier on the upstream link boundary. Similarly, if a link is in a spillback state, then $\overleftarrow{v}_a(t)$ on the upstream link boundary is directly influenced by outflow rates $v_a(\cdot)$ some time earlier on the downstream link boundary. These relationships are indicated by arrows on top of the variables.

Define the potential cumulative link outflows $\vec{U}_a(t)$ [veh] and potential cumulative link outflows $\overleftarrow{V}_a(t)$ [veh] as

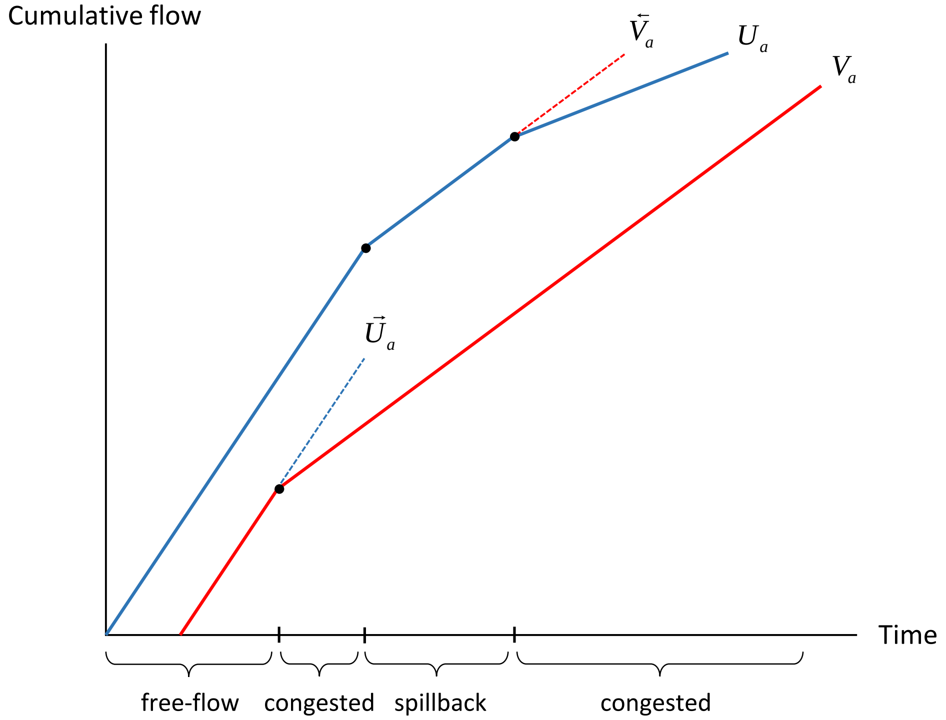
$$\vec{U}_a(t) = \int_0^t \vec{u}_a(\omega) d\omega, \quad \text{and} \quad \overleftarrow{V}_a(t) = \int_0^t \overleftarrow{v}_a(\omega) d\omega, \qquad \forall a \in \mathcal{A}. \tag{7}$$

A link is in a free-flow state if $\vec{U}_a(t) = V_a(t),$ that is, if the number of vehicles in a queue (the excess demand) defined by $\vec{U}_a(t) - V_a(t)$ equals zero. A link is in spillback state if $\overleftarrow{V}_a(t) = U_a(t),$ that is, if the number of additional vehicles that can still flow into the link (the excess supply) defined by $\overleftarrow{V}_a(t) - U_a(t)$ equals zero. Note that by definition $\vec{U}_a(t) \geq V_a(t)$ and $\overleftarrow{V}_a(t) \geq U_a(t).$ Figure 4 graphically illustrates the different link states and their relationship with cumulative (potential) link inflows and outflows.

In hypocritical conditions let us define $\sigma_{I,a}(q) = q / \Phi_{I,a}^{-1}(q)$ [km/h] as the vehicle speed and $\gamma_{I,a}(q) = 1 / \left(d\Phi_{I,a}^{-1}(q) / dq \right)$ [km/h] as the (kinematic) wave speed. Similarly, in hypercritical conditions let $\sigma_{II,a}(q) = q / \Phi_{II,a}^{-1}(q)$ [km/h] be the corresponding vehicle speed and $\gamma_{II,a}(q) = 1 / \left(d\Phi_{II,a}^{-1}(q) / dq \right)$ [km/h] the wave speed. Define $\xi_{I,a}(q)$ [veh] and $\xi_{II,a}(q)$ [veh] as the number of vehicles one passes if one travels at the same speed the hypocritical and hypercritical traffic states propagate, respectively, which can be computed as

$$\xi_{I,a}(q) = L_a q \left(\frac{1}{\gamma_{I,a}(q)} - \frac{1}{\sigma_{I,a}(q)} \right), \quad \text{and} \quad \xi_{II,a}(q) = L_a q \left(\frac{1}{\sigma_{II,a}(q)} - \frac{1}{\gamma_{II,a}(q)} \right), \qquad \forall a \in \mathcal{A}. \tag{8}$$

Figure 4: Cumulative (potential) link inflows and outflows in different link states



Then in case the upstream link boundary is in a hypocritical state, the potential cumulative outflow is defined as

$$\vec{U}_a(t) = \min_{q\in[0,Q]} \left\{ U_a\left(t - \frac{L_a}{\gamma_{I,a}(q)}\right) + \xi_{I,a}(q) \right\}, \qquad \forall a \in \mathcal{A}, \tag{9}$$

and we define $\vec{u}_a(t) \in \partial\vec{U}_a(t)$ as the potential outflow rate that minimises Eqn. (9). Similarly, if the downstream link is in a hypocritical state, then the potential cumulative inflow is defined as

$$\overleftarrow{V}_a(t) = \min_{q\in[0,Q]} \left\{ V_a\left(t + \frac{L_a}{\gamma_{II,a}(q)}\right) + \xi_{II,a}(q) \right\}, \qquad \forall a \in \mathcal{A}. \tag{10}$$

where $\overleftarrow{v}_a(t) \in \partial\overleftarrow{V}_a(t)$ is the potential inflow rate that minimises Eqn. (10).

3.2.2 Propagation of path information

In a path-based DNL, path information is preserved by tracking multi-commodity flow along each link and across links. Let $u_{ap}^w(t)$ [veh/h] and $v_{ap}^w(t)$ [veh/h] be the OD and path-specific link inflow and outflow rates at time instant *t*, respectively. By definition it holds that

$$u_a(t) = \sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}^w} u_{ap}^w(t), \quad \text{and} \quad v_a(t) = \sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}^w} v_{ap}^w(t), \qquad \forall a \in \mathcal{A}. \tag{11}$$

For conservation of path information *across* links we adopt the following constraint for all $b \in \mathcal{A}$, $p \in \mathcal{P}^w$, $w \in \mathcal{W}$,

$$u_{bp}^w(t) = \begin{cases} f_p^{wi}, & \text{if link } b \text{ is the first link on path } p; \\ v_{ap}^w(t), & \text{otherwise, where } a \text{ is the previous link on path } p, \end{cases} \qquad \text{for } t \in [t_i, t_i + T_i). \tag{12}$$

To ensure that path information is conserved *within* a link, we need to let upstream path proportions $u_{ap}^w(t)/u_a(t)$ travel at the speed with which vehicles traverse a link. Let $\tau_a(t)$ [h] denote the travel time that a vehicle entering link a at time instant t experiences. Section 2.3 discusses how to determine these link travel times. Since first-in-first-out (FIFO) is satisfied in the general link transmission model with a single vehicle class, the downstream path proportions $\rho_{ap}(t)$ [-] are given by

$$\rho_{ap}^{w}\left(t+\tau_{a}(t)\right)=\frac{u_{ap}^{w}(t)}{u_{a}(t)}, \qquad \forall a\in\mathcal{A}, \forall p\in\mathcal{P}^{w}, \forall w\in\mathcal{W}. \tag{13}$$

3.2.3 Determining turn flows through nodes

While the link model shows important dynamics on how traffic state information and path information travel along a link over time, the non-spatial node model that we adopt here on the other hand is time-invariant. A general node model for each node $n\in\mathcal{N}$ can be written in the form of an implicit function $\Gamma(\cdot)$ that takes the following inputs: sending flow rates $\mathbf{s}_{n}(t)=[s_{a}(t)]_{a\in\mathcal{A}_{n}^{-}}$ [veh/h], receiving flow rates $\mathbf{r}_{n}(t)=[r_{b}(t)]_{b\in\mathcal{A}_{n}^{+}}$ [veh/h], and turn flow proportions $\boldsymbol{\varphi}_{n}(t)=[\varphi_{ab}(t)]_{a\in\mathcal{A}_{n}^{-},b\in\mathcal{A}_{n}^{+}}$ [-]. Output of the node model are exit flow reduction factors for each incoming link that describes the proportion of sending flow that is able to flow out, denoted by $\boldsymbol{\lambda}_{n}(t)=[\lambda_{a}(t)]_{a\in\mathcal{A}_{n}^{-}}$ [-]. Hence, formally the node model is stated as

$$\boldsymbol{\lambda}_{n}(t)=\Gamma\left(\mathbf{s}_{n}(t),\mathbf{r}_{n}(t),\boldsymbol{\varphi}_{n}(t)\right), \qquad \forall n\in\mathcal{N}. \tag{14}$$

The inputs follow directly from the link model, namely

$$s_{a}(t)=\begin{cases}\vec{u}_{a}(t), & \text{if } \vec{U}_{a}(t)=V_{a}(t) \text{ (i.e., the link is in a free-flow state),}\\ Q_{a}, & \text{otherwise,}\end{cases} \qquad \forall a\in\mathcal{A}, \tag{15}$$

$$r_{a}(t)=\begin{cases}\bar{v}_{a}(t), & \text{if } \bar{V}_{a}(t)=U_{a}(t) \text{ (i.e., the link is in a spillback state),}\\ Q_{a}, & \text{otherwise,}\end{cases} \qquad \forall a\in\mathcal{A}, \tag{16}$$

$$\varphi_{ab}(t)=\sum_{w\in\mathcal{W}}\sum_{p\in\mathcal{P}^{w}}\delta_{bp}^{w}\rho_{ap}^{w}(t), \qquad \forall a\in\mathcal{A}_{n}^{-}, \forall b\in\mathcal{A}_{n}^{+}, \forall n\in\mathcal{N}. \tag{17}$$

Link outflow rates can be computed as

$$v_{a}(t)=\lambda_{a}(t)s_{a}(t), \qquad \forall a\in\mathcal{A}, \tag{18}$$

while path-specific outflow rates follow from

$$v_{ap}^{w}(t)=\rho_{ap}^{w}(t)v_{a}(t), \qquad \forall a\in\mathcal{A}, \forall p\in\mathcal{P}^{w}, \forall w\in\mathcal{W}. \tag{19}$$

Several formulations for function $\Gamma(\cdot)$ exist in the literature, but not all are considered first order node models. Tampère et al. (2011) formulated a series of requirements for first order node models, which includes that function $\Gamma(\cdot)$ satisfies the invariance principle and satisfies conservation of turn flows. These two properties will become useful in simplifying the model formulation later in this paper. From here on we assume that function $\Gamma(\cdot)$ satisfies all first order requirements.

3.3 Travel time calculation

The link travel time $\tau_{a}(t)$ for a vehicle entering link a at time instant t can be determined using cumulative link inflows and outflows. Because FIFO is satisfies, it holds that

$$U_{a}(t)=V_{a}\left(t+\tau_{a}(t)\right), \qquad \forall a\in\mathcal{A}. \tag{20}$$

Then we can compute $\tau_{a}(t)=V_{a}^{-1}\left(U_{a}(t)\right)-t.$ In general, a vehicle may encounter different hypocritical and hypercritical traffic states while traversing a link, such that $\tau_{a}(t)$ can only be calculated implicitly at the moment a vehicle exits the link.

However, if link inflow and outflow rates are stationary during a certain period of time, we can calculate these travel times explicitly, which becomes useful in deriving travel times in our quasi-dynamic model. Suppose that $u_a(t)=\bar{u}_a$ and $v_a(t)=\bar{v}_a$ during some period of time such that a vehicle entering during this time period does not encounter any traffic states other than $\bar{u}_a$ and $\bar{v}_a$ while traversing the link. Then we can calculate the travel time by splitting it into two components. A common way to do this is to distinguish travel time under free-flow (hypocritical) conditions and travel time under congested (hypercritical) conditions. However, this requires explicitly keeping track of the tail of the queue. Here we propose an alternative way that does not require information regarding traffic states within the link, but instead only uses information from potential cumulative outflow $\vec{U}_a(t)$ and potential cumulative inflow $\overleftarrow{V}_a(t)$.

First, we assume that the link is either in a free-flow or a congested state for which holds that $\bar{u}_a \geq \bar{v}_a$. Then we can write $\tau_a(t)=\tau^0_{I,a}+\tau^+_a(t+\tau^0_{I,a})$, where stationary minimum (hypocritical) travel time $\tau^0_{I,a}$ and additional (time-dependent) delay $\tau^+_a(t+\tau^0_{I,a})$ are defined through

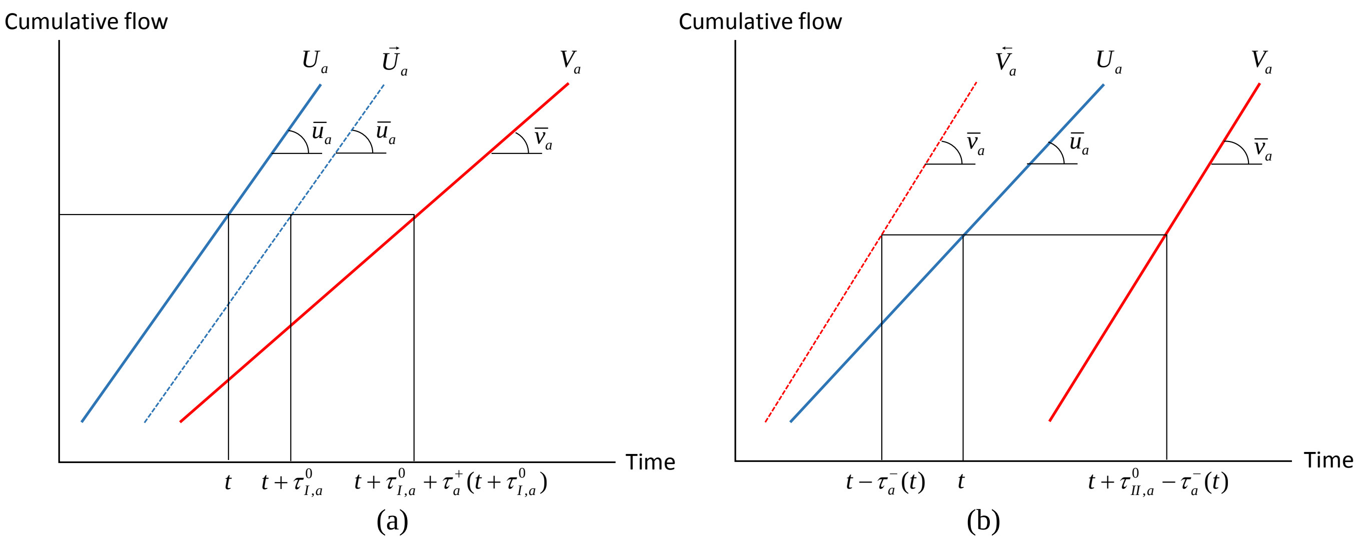
$$U_a(t)=\vec{U}_a\left(t+\tau^0_{I,a}\right)=V_a\left(t+\tau^0_{I,a}+\tau^+_a(t+\tau^0_{I,a})\right),\qquad \forall a\in\mathcal{A}. \tag{21}$$

This situation is graphically illustrated in Figure 5(a), and it follows that

$$\tau^0_{I,a}=\frac{L_a}{\sigma_{I,a}(\bar{u}_a)},\qquad \forall a\in\mathcal{A}, \tag{22}$$

$$\tau^+_a(t+\tau^0_a)=\frac{\vec{U}_a(t+\tau^0_a)-V_a(t+\tau^0_a)}{\bar{v}_a},\qquad \forall a\in\mathcal{A}. \tag{23}$$

Figure 5: Explicitly determining link travel times in case of stationary inflow and outflow rates



Now assume that the link is either in a spillback or a congested state for which holds that $\bar{u}_a \leq \bar{v}_a$. Then we can write $\tau_a(t)=\tau^0_{II,a}-\tau^-_a(t)$, where stationary maximum (hypercritical) travel time $\tau^0_{II,a}$ and (time-dependent) time saving $\tau^-_a(t)$ are defined through

$$\overleftarrow{V}_a\left(t-\tau^-_a(t)\right)=U_a(t)=V_a\left(t+\tau^0_{II,a}-\tau^-_a(t)\right),\qquad \forall a\in\mathcal{A}. \tag{24}$$

Referring to Figure 5(b), it holds that

$$\tau^0_{II,a}=\frac{L_a}{\sigma_{II,a}(\bar{v}_a)},\qquad \forall a\in\mathcal{A}, \tag{25}$$

$$\tau_a^-(t) = \frac{\overleftarrow{V}_a(t) - U_a(t)}{\overline{v}_a}, \qquad \forall a \in \mathcal{A}. \tag{26}$$

Now let us consider the calculation of path travel times. Let $U_{ap}^w(t)$ and $V_{ap}^w(t)$ denote the path-specific cumulative inflows and outflows, respectively. Since FIFO holds on the link level and we do not consider travel times inside a node, FIFO also holds on the path level. Then similar to Eqn. (20) we can define path travel time $\tau_p^w(t)$ for a vehicle departing at time instant t on path p implicitly through $U_{\underline{a}_p p}^w(t) = V_{\overline{a}_p p}^w\left(t + \tau_p^w(t)\right)$, where $\underline{a}_p$ and $\overline{a}_p$ denote the first and last link on path p, respectively. Therefore, $\tau_p^w(t) = V_{\overline{a}_p p}^{w\ -1}\left(U_{\underline{a}_p p}^w(t)\right) - t$, and average path travel times during departure time interval i can be calculated as

$$\tau_p^{wi} = \frac{1}{f_p^{wi} T^i} \int_{U_{\underline{a}_p}^w(t^i)}^{U_{\underline{a}_p}^w(t^i) + f_p^{wi} T^i} \left(V_{\overline{a}_p p}^{w\ -1}(y) - U_{\underline{a}_p p}^{w\ -1}(y)\right) dy, \qquad \forall p \in \mathcal{P}^w, w \in \mathcal{W}, i \in \mathcal{I}. \tag{27}$$

where $\underline{a}_p$ and $\overline{a}_p$ denote the first and last link on path p, respectively.

4. Static model with residual queues and spillback

In this section we provide our novel static TA model formulation. In order to keep the relevant equations together, we first show the final result, and then explain how the model can be derived from our first order dynamic TA model. We consider our model a time-averaged version of the dynamic model in which we consider representative flow rates equal to the average flow rates during each departure time interval. As such, continuous-time flow rate variables for each time instant *t* are replaced with discrete variables that describe averages over each time period *i*.

4.1 Model formulation

We can find static stochastic user equilibrium path flows $\overline{\mathbf{f}}^i$ for each departure time interval $i \in \mathcal{I}$ by solving the following static VI problem,

$$\sum_{w\in\mathcal{W}} \sum_{p\in\mathcal{P}^w} c_p^{wi}\left(\overline{\mathbf{f}}^i\right)\left(f_p^{wi} - \overline{f}_p^{wi}\right) \geq 0, \quad \forall \mathbf{f}^i \in \Omega^i. \tag{28}$$

The average perceived and experienced path travel times are calculated as

$$c_p^{wi} = \tau_p^{wi} + \frac{1}{\mu}\log f_p^{wi}, \qquad \forall p\in\mathcal{P}^w, \forall w\in\mathcal{W}, \forall i\in\mathcal{I}, \tag{29}$$

$$\tau_p^{wi} = \sum_{a\in p} \frac{L_a}{\sigma_{I.a}(u_a^i)} + \frac{T^i}{2}\left(\prod_{a\in p}\frac{1}{\lambda_a^i} - 1\right), \qquad \forall p\in\mathcal{P}^w, \forall w\in\mathcal{W}, \forall i\in\mathcal{I}. \tag{30}$$

Travel times depend on average link inflow rates u_a^i [veh/h] and average link exit flow reduction factors λ_a^i [-], which are obtained by solving the following system of equations that describe quasi-dynamic network loading:

(path-specific link inflow rate) $$u_{ap}^{wi} = \delta_{ap}^w f_p^{wi} \prod_{a'\in p_a} \lambda_{a'}^i, \qquad \forall a\in\mathcal{A}, \forall p\in\mathcal{P}^w, \forall w\in\mathcal{W}, \forall i\in\mathcal{I}, \tag{31}$$

(link inflow rate) $$u_a^i = \sum_{w\in\mathcal{W}}\sum_{p\in\mathcal{P}^w} u_{ap}^{wi}, \qquad \forall a\in\mathcal{A}, \forall i\in\mathcal{I}, \tag{32}$$

(splitting rate) $$\varphi_{ab}^i = \frac{1}{u_a^i}\sum_{w\in\mathcal{W}}\sum_{p\in\mathcal{P}^w} \delta_{bp}^w u_{ap}^{wi}, \qquad \forall a\in\mathcal{A}_n^-, \forall b\in\mathcal{A}_n^+, \forall n\in\mathcal{N}, \forall i\in\mathcal{I}, \tag{33}$$

(link outflow rate) $$v_a^i = \lambda_a^i u_a^i, \qquad \forall a\in\mathcal{A}, \forall i\in\mathcal{I}, \tag{34}$$

(receiving flow rate) $$r_a^i = \min\left\{v_a^i + \frac{L_a \Phi_{II}^{-1}(v_a^i)}{T^i}, Q_a\right\}, \quad \forall a\in\mathcal{A}, \forall i\in\mathcal{I}, \tag{35}$$

(exit flow reduction factor) $$\boldsymbol{\lambda}_n^i = \Gamma\left(\mathbf{u}_n^i, \mathbf{r}_n^i, \boldsymbol{\varphi}_n^i\right), \qquad \forall n\in\mathcal{N}, \forall i\in\mathcal{I}. \tag{36}$$

Eqn. (31) describes the propagation of average path flows f_p^{wi} through the network to obtain path-specific average link inflow rates u_{ap}^{wi} [veh/h], explicitly taking into account capacity constraints as well as storage constraints on downstream links. These constraints are jointly represented in link reduction factors λ_a^i. Sub-path p_a is defined as the set of all links in path *p* up to (but not including) link *a*. Eqn. (32) is a simple definition that states that average link inflow rates are the sum of all path-specific average inflow rates through that link. For each node, turn flow proportions are calculated in Eqn. (33), where δ_{ap}^w [-] is a link-path incidence indicator that is equal to one if link *a* is on path $p\in\mathcal{P}^w$, and is zero otherwise. Eqn. (34) states that the average link outflow rate v_a^i [veh/h] is equal to the average link inflow rate multiplied with the link reduction factor. Eqn. (35) states that the average receiving flow r_a^i [veh/h] is equal to Q_a if there is no spillback, while it is restricted through a function of the average link outflow rate v_a^i in case of spillback. Finally, Eqn. (36) determines link reduction factors through the node model, where $\boldsymbol{\lambda}_n^i = [\lambda_a^i]_{a\in\mathcal{A}_n^-}$, $\mathbf{u}_n^i = [u_a^i]_{a\in\mathcal{A}_n^-}$, $\mathbf{r}_n^i = [r_b^i]_{b\in\mathcal{A}_n^+}$, and $\boldsymbol{\varphi}_n^i = [\varphi_{ab}^i]_{a\in\mathcal{A}_n^-, b\in\mathcal{A}_n^+}$.

The remainder of this section is dedicated to mathematically deriving the presented model formulation from the first order dynamic model described in Section 2 using static assumptions mentioned in Table 1. This will be done in reverse order, namely starting with the quasi-dynamic network loading equations, then the average path travel time equations, and finally the route choice formulation.

4.2 Derivation of the quasi-dynamic network loading formulation

We start with the first order dynamic TA model described in Section 2. The first static assumption that we consider is "no residual traffic transfer" (T.3). Residual traffic refers to vehicles that are present on the network at the beginning a time interval as a result of a previous time period. These vehicles could not reach their final destination because of a travel time longer than the duration of the time period, which typically occurs with long distance trips and vehicles that are delayed in a residual queue. While dynamic models are perfectly capable of handling residual traffic, static models do not describe these dynamics and simply assume that there are no longer vehicles from a previous departure time period on the network, even if path travel times are actually longer than time period duration T_i.

From a model perspective this means that the network is considered empty at the beginning of each time interval *i*, i.e. residual traffic is given by $U_a(t^i) - V_a(t^i) = 0,$ and no interactions exist between traffic flows departing during different time periods. Without loss of generality we can set $U_a(t^i) = V_a(t^i) = 0.$

In this section we are looking for average inflow and outflow rates over each time interval *i*, which are defined as

$$u_a^i = \frac{U_a(t^i + T^i) - U_a(t^i)}{T^i} = \frac{U_a(t^i + T^i)}{T^i}, \qquad \forall a \in \mathcal{A}, \forall i \in \mathfrak{I}, \tag{37}$$

$$v_a^i = \frac{V_a(t^i + T^i) - V_a(t^i)}{T^i} = \frac{V_a(t^i + T^i)}{T^i}, \qquad \forall a \in \mathcal{A}, \forall i \in \mathfrak{I}. \tag{38}$$

Looking at the potential cumulative outflows and inflows in Eqns. (9)-(10), the first thing we note is that since we only consider a single inflow and outflow rate per link, there is no need to minimise over flow rates. Therefore, we simply obtain average potential flow rates as $\vec{u}_a^i = \vec{U}_a(t_i + T_i) / T_i$ [veh/h] and $\overleftarrow{v}_a^i = \overleftarrow{V}_a(t_i + T_i) / T_i$ [veh/h].

Now let us assume "infinite wave speeds" (T.1) in both the hypocritical and the hypercritical branches of the fundamental diagram, and "infinite vehicle propagation speed" (T.2) in the hypocritical branch. This means that $\gamma_{I,a}(u_a^i) \to \infty,$ $\gamma_{II,a}(v_a^i) \to -\infty,$ and $\sigma_{I,a}(u_a^i) \to \infty,$ which simplifies Eqn. (8) to $\xi_{I,a}(q) = 0$ and $\xi_{II,a}(q) = L_a q / \sigma_{II,a}(q),$ such that Eqns. (9) and (10) respectively result in

$$\vec{U}_a(t) = U_a(t), \quad \text{for} \quad t \in [t^i, t^i + T^i), \quad \forall a \in \mathcal{A}, \tag{39}$$

$$\overleftarrow{V}_a(t) = V_a(t) + \frac{L_a v_a^i}{\sigma_{II,a}(v_a^i)}, \quad \text{for} \quad t \in [t^i, t^i + T^i), \quad \forall a \in \mathcal{A}. \tag{40}$$

Since according to (37) it holds by definition that $U_a(t^i + T^i) = T^i u_a^i$ and $V_a(t^i + T^i) = T^i v_a^i,$ Eqns. (39) -(40) can respectively be rewritten as

$$\vec{U}_a(t^i + T^i) = T^i u_a^i, \quad \text{and} \quad \vec{u}_a^i = u_a^i, \qquad \forall a \in \mathcal{A}, \forall i \in \mathfrak{I}, \tag{41}$$

$$\overleftarrow{V}_a(t^i + T^i) = T^i v_a^i + \frac{L_a v_a^i}{\sigma_{II,a}(v_a^i)} = T^i v_a^i + L_a \Phi_{II,a}^{-1}(v_a^i), \quad \text{and} \quad \overleftarrow{v}_a^i = v_a^i + \frac{L_a \Phi_{II,a}^{-1}(v_a^i)}{T^i}, \quad \forall a \in \mathcal{A}, \forall i \in \mathfrak{I}. \tag{42}$$

The potential average outflow rate as shown in Eqn. (41) is simply the average inflow rate. More interesting is the potential inflow rate shown in Eqn. (42). Note that $\Phi_{II,a}^{-1}(v_a^i)$ is the queuing density at hypocritical flow rate v_a^i. Therefore, the total number of vehicles that can potentially flow into link *a* during time interval *i* is equal to the total number of vehicles exiting the link, $T^i v_a^i$, plus the available vehicle storage on the link described by $L_a \Phi_{II,a}^{-1}(v_a^i)$.

Using (41) and (42) we can reformulate Eqns. (15) and (16) respectively as

$$s_a^i = \begin{cases} u_a^i, & \text{if } u_a^i = v_a^i, \\ Q_a, & \text{if } u_a^i > v_a^i, \end{cases} \qquad \forall a \in \mathcal{A}, \tag{43}$$

$$r_a^i = \begin{cases} v_a^i + \dfrac{L_a \Phi_{II,a}^{-1}(v_a^i)}{T^i}, & \text{if } v_a^i + \dfrac{L_a \Phi_{II,a}^{-1}(v_a^i)}{T^i} = u_a^i, \\ Q_a, & \text{if } v_a^i + \dfrac{L_a \Phi_{II,a}^{-1}(v_a^i)}{T^i} > u_a^i, \end{cases} \qquad \forall a \in \mathcal{A}. \tag{44}$$

Since (43) holds for each $v_a^i \in [0, Q_a]$ and (44) holds for each $u_a^i \in [0, Q_a]$, these average sending and receiving flow rates can be simplified to

$$s_a^i = \min\left\{u_a^i, Q_a\right\} = u_a^i, \qquad \forall a \in \mathcal{A}, \tag{45}$$

$$r_a^i = \min\left\{v_a^i + \frac{L_a \Phi_{II,a}^{-1}(v_a^i)}{T^i}, Q_a\right\}, \qquad \forall a \in \mathcal{A}. \tag{46}$$

Eqn. (45) does not mean that no congestion occurs, but rather that it is not necessary to use a sending flow rate equal to capacity. This can also be seen from the invariance principle of the node model which states that if a link is in congestion, then increasing the sending flow on that link does not change the link outflow rates. Therefore, in case link *a* is congested, then using $s_a^i = u_a^i$ yields the same result as using $s_a^i = Q_a$. Eqn. (46) provides our first result that enters the static model as Eqn. (35).

Next, we consider average path-specific link inflow rates u_{ap}^{wi}, and by the same definition as in Eqn. (11) it holds that

$$u_a^i = \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}^w} u_{ap}^{wi}, \qquad \forall a \in \mathcal{A}, \forall i \in \mathcal{I}, \tag{47}$$

which shows our second result conform Eqn. (32).

Based on Eqn. (13) the average downstream flow proportions ρ_{ap}^{wi} [-] become

$$\rho_{ap}^{wi} = \frac{u_{ap}^{wi}}{u_a^i}, \qquad \forall b \in \mathcal{A}, \forall p \in \mathcal{P}^w, \forall w \in \mathcal{W}, \forall i \in \mathcal{I}. \tag{48}$$

Note that travel time $\tau_a(t)$ disappeared in this equation. In order to calculate travel times, we make the simplifying assumption that link inflow and outflow rates are stationary at the average flow rates. Since we start with an empty network, it holds that $u_a^i \geq v_a^i$ such that we can apply Eqns. (22) and (23) to obtain link travel times

$$\tau_a(t) = \frac{L_a}{\sigma_{I,a}(u_a^i)} + \left(t + \frac{L_a}{\sigma_{I,a}(u_a^i)}\right)\frac{u_a^i - v_a^i}{v_a^i}, \qquad \text{for} \quad t \in [t^i, t^i + T^i), \quad \forall a \in \mathcal{A}. \tag{49}$$

Since we assumed "infinite vehicle propagation speeds" (T.2), $\sigma_{I,a}(u_a^i)$ is set to infinity such that $\tau_a(t^i) = 0$ and $\tau_a(t^i + T^i) = T^i(u_a^i - v_a^i)/v_a^i$. In other words, the flow proportions are fixed during the entire time period *i*, but are also kept to the same fixed rate for some time after the end of time period *i* to allow residual traffic to leave the network.

Path proportions in Eqn. (48) can be used to calculate average turn proportions φ_{ab}^i in a similar way to Eqn. (17),

$$\varphi_{ab}^{i} = \frac{1}{u_a^i} \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}^w} \delta_{bp}^{w} u_{ap}^{wi}, \qquad \forall a \in \mathcal{A}_n^-, \forall b \in \mathcal{A}_n^+, \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \tag{50}$$

This provides our third result in the static model, see Eqn. (33).

Since the node model provided in Eqn. (14) is time invariant and we assume stationary inflow and outflow rates per time period, together with the fact that $s_a^i = u_a^i$ as stated in Eqn. (45), average link reduction factors can be determined as

$$\boldsymbol{\lambda}_n^i = \Gamma\left(\mathbf{u}_n^i, \mathbf{r}_n^i, \boldsymbol{\varphi}_n^i\right), \qquad \forall n \in \mathcal{N}, \forall i \in \mathcal{I}, \tag{51}$$

which establishes our fourth result conform Eqn. (36).

Knowing the link reduction factors, we can readily determine link outflow rates similar to Eqn. (18) as

$$v_a^i = \lambda_a^i u_a^i, \qquad \forall a \in \mathcal{A}, \forall i \in \mathcal{I}, \tag{52}$$

and leads to our fifth result as stated in Eqn. (34).

Finally, we look at the propagation of path information. The time averaged equivalent of Eqn. (12) is

$$u_{bp}^{wi} = \begin{cases} f_p^{wi}, & \text{if link } b \text{ is the first link on path } p; \\ v_{ap}^{wi}, & \text{otherwise, where } a \text{ is the previous link on path } p, \end{cases} \qquad \forall b \in \mathcal{A}, \forall p \in \mathcal{P}^w, \forall w \in \mathcal{W}, \forall i \in \mathcal{I}. \tag{53}$$

Applying this equation where link b is not the first link on path p gives us $u_{bp}^{wi} = v_{ap}^{wi} = \rho_{ap}^{wi} v_a^i = \rho_{ap}^{wi} \lambda_a^i s_a^i = \rho_{ap}^{wi} \lambda_a^i u_a^i = \lambda_a^i u_{ap}^{wi}$, and applying it recursively back to the first link on that path yields

$$u_{ap}^{wi} = \delta_{ap}^{w} f_p^{wi} \prod_{a' \in p_a} \lambda_{a'}^i, \qquad \forall a \in \mathcal{A}, \forall p \in \mathcal{P}^w, \forall w \in \mathcal{W}, \forall i \in \mathcal{I}. \tag{54}$$

This concludes our final result of the network loading equations conform Eqn. (31).

4.3 Derivation of the average (perceived) path travel time

Average experienced path travel times τ_p^{wi} can be calculated conform Eqn. (27). Since for the travel time calculations we assume that all links have stationary inflow and outflow rates, similar to writing link travel times as a minimum travel time plus an additional delay we can write path travel times as $\tau_p^{wi} = \tau_{I,p}^{wi,0} + \tau_p^{wi,+}$, where $\tau_{I,p}^{wi,0}$ is a fixed minimum hypocritical travel time on path *p* during time interval *i* (ignoring congestion) that is defined as

$$\tau_{I,p}^{w,0} = \sum_{a \in p} \frac{L_a}{\sigma_{I,a}(u_a^i)}, \qquad \forall p \in \mathcal{P}^w, \forall w \in \mathcal{W}, \forall i \in \mathcal{I}, \tag{55}$$

and where average additional path delay during interval *i*, $\tau_p^{wi,+}$, can be calculated as

$$\tau_p^{wi,+} = \frac{1}{f_p^{wi} T^i} \int_{U_{\underline{a}p}^w(t^i)}^{U_{\underline{a}p}^w(t^i) + f_p^{wi} T^i} \left(V_{\bar{a}_p p}^{w\ -1}(y) - \vec{U}_{\underline{a}_p p}^{w\ -1}(y) \right) dy. \tag{56}$$

Note that we have replaced $U_{\underline{a}p}^w(t)$ in Eqn. (27) with $\vec{U}_{\underline{a}p}^w(t)$ because in our static assignment it holds that $\vec{U}_{\underline{a}p}^w(t) = U_{\underline{a}p}^w(t)$ (since hypocritical vehicle flow propagation is considered instantaneous), therefore Eqn. (56) only considers the additional queuing delays. It holds that $U_{\underline{a}p}^w(t) = t f_p^{wi}$ and $V_{\bar{a}_p p}^w(t) = t f_p^{wi} \prod_{a' \in p} \lambda_{a'}^i$, such that Eqn. (56) simplifies to

$$\begin{aligned}\tau_p^{wi,+} &= \frac{1}{f_p^{wi}T^i}\int_0^{f_p^{wi}T^i}\left(\frac{y}{f_p^{wi}\prod_{a'\in p}\lambda_{a'}^i}-\frac{y}{f_p^{wi}}\right)dy \\ &= \frac{1}{f_p^{wi}T^i}\left[\frac{y^2}{2f_p^{wi}\prod_{a'\in p}\lambda_{a'}^i}-\frac{y^2}{2f_p^{wi}}\right]_{y=0}^{f_p^{wi}T^i} \\ &= \frac{T^i}{2}\left(\frac{1}{\prod_{a'\in p}\lambda_{a'}^i}-1\right).\end{aligned} \tag{57}$$

Therefore, average experienced path travel times are the sum of Eqns. (55) and (57), which yields the result expressed in Eqn. (30). It should be noted that this expression is identical to the expression derived in Bliemer et al. (2014), even though it has been derived in a different way. Finally, perceived average path travel times c_p^{wi} in Eqn. (29) are calculated in the same way as in Eqn. (2).

4.4 Derivation of the route choice formulation

It is clear that due to the assumption of no residual traffic transfer, average path travel times τ_p^{wi} only depend on path flow rates $\mathbf{f}^i$ and are independent of flow rates that depart during time periods other than *i*. Since τ_p^{wi} only depends on $\mathbf{f}^i$ instead of $\mathbf{f}$, also c_p^{wi} only depends on $\mathbf{f}^i$.

Since also the set of feasible path flows Ω is separable by departure time interval *i* (see Section 3.1), the dynamic VI problem in (4) simplifies to a series of independent static VI problems shown in (28). It is easy to see that if (28) holds for each time period *i*, then also (4) holds. Hence, the static formulation is special case of the dynamic formulation.

Since the static model for each time period *i* can be solved independently, for notational convenience we omit superscript *i* in all variables from here onwards.

5. Model structure and special cases

The quasi-dynamic network loading problem as described by Eqns. (31)–(36) is not trivial to solve due to many interactions between variables on the transport network. Exit flow reduction factors $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_n]_{n\in\mathcal{N}}$ depend link inflow rates $\mathbf{u} = [\mathbf{u}_n]_{n\in\mathcal{N}}$ and turn proportions $\boldsymbol{\varphi} = [\boldsymbol{\varphi}_n]_{n\in\mathcal{N}}$, which both depend on path flow rates $\mathbf{f}$, and also depend on receiving flow rates $\mathbf{r} = [\mathbf{r}_n]_{n\in\mathcal{N}}$ that in turn depend on link outflow rates $\mathbf{v} = [\mathbf{v}_n]_{n\in\mathcal{N}}$. Further, both $\mathbf{u}$ and $\mathbf{v}$ depend on $\boldsymbol{\lambda}$. In this section we explore these relationships and also show that several existing models are special cases of our model formulation.

5.1 Disentangling exit flow reduction factors

The exit flow reduction factors are the result of both capacity constraints as well as (queue) storage constraints. In other words, if $\lambda_a < 1,$ then we do not know whether the exit flow is reduced because of capacity constraints, or because of storage constraints, or by both. In order to get a better understanding of these exit flow reduction factors, which is also important in designing an algorithm that can find a solution to our quasi-dynamic network loading problem, we disentangle λ_a into two components. To be specific, we define $\lambda_a = \alpha_a \beta_a$, where α_a [-] is the exit flow reduction factor of link *a* due to capacity constraints on a downstream link (i.e., ignoring spillback), and β_a [-] is the additional exit flow reduction factor of link *a* due to storage constraints on a downstream link (i.e., the impact of spillback). Note that a storage constraint is always due to a capacity constraint further downstream. For each node we can determine $\boldsymbol{\alpha}_n = [\alpha_n]_{a\in\mathcal{A}_n^-}$ as

$$\boldsymbol{\alpha}_n = \Gamma\left(\mathbf{u}_n, \mathbf{Q}_n, \boldsymbol{\varphi}_n\right), \qquad \forall n \in \mathcal{N}, \tag{58}$$

where $\mathbf{Q}_n = [Q_b]_{b\in\mathcal{A}_n^+}$ [veh/h] is the vector of fixed physical capacities of outgoing links. Therefore, $\boldsymbol{\alpha}_n$ focuses on the impact of the sending flows (that are equal to link inflows) and turn proportions, keeping receiving flows fixed to capacity. We can calculate $\boldsymbol{\beta}_n = [\beta_n]_{a\in\mathcal{A}_n^-}$ by first calculating $\boldsymbol{\lambda}_n$ via Eqn. (36) and then simply define

$$\beta_a = \frac{\lambda_a}{\alpha_a}, \qquad \forall a \in \mathcal{A}. \tag{59}$$

Therefore, $\boldsymbol{\beta}_n$ focuses on the impact of a decrease in receiving flows due to spillback, keeping sending flows and turn proportions fixed. We add Eqns. (58) and (59) to our system of equations, (31)–(36). Since (58) and (59) are purely definitions, adding these equations will not alter the solution.

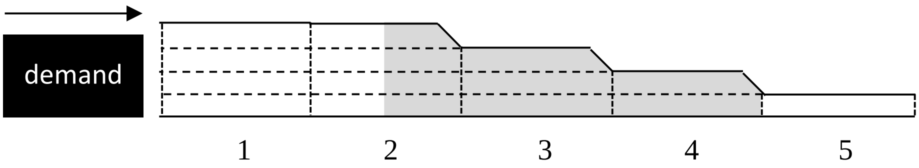
In order to illustrate, consider the five link corridor link Figure 6. We can distinguish four cases as defined in Table 2. Since $\alpha_4 < 1$ and $\beta_4 = 1,$ link 4 is only constrained by the capacity of the downstream link, making link 5 an active bottleneck. The queue on link 4 spills back onto link 3 and link 2. As such, the downstream storage constraint dominates the capacity constraint on the link directly downstream and hence $\alpha_3 = 1$ and $\beta_3 < 1.$ Therefore, link 4 is not an active bottleneck, but rather link 3 is constrained by a bottleneck further downstream (link 5). For link 2 is holds that $\alpha_2 < 1$ and $\beta_2 < 1,$ which means that link 2 is constrained by the capacity of link 3 (which is therefore an active bottleneck) as well as by the storage on link 3.

The set of links for which $\alpha_a < 1$ indicates where queues originate. We call a link with $\alpha_a < 1$ a *spillback root*, indicating the location where a queue originates. Denote the set of all spillback roots by set $\mathcal{A}^*$, which is defined as

$$\mathcal{A}^* = \left\{a \in \mathcal{A} :\ \alpha_a < 1\right\}. \tag{60}$$

In order to determine the upstream propagation of queues, we always start with the spillback roots in set $\mathcal{A}^*$.

Figure 6: Exit flow reduction factors due to capacity and storage constraints



1	2	3	4	5
$\alpha_1 = 1$	$\alpha_2 < 1$	$\alpha_3 = 1$	$\alpha_4 < 1$	$\alpha_5 = 1$
$\beta_1 = 1$	$\beta_2 < 1$	$\beta_3 < 1$	$\beta_4 = 1$	$\beta_5 = 1$
$\lambda_1 = 1$	$\lambda_2 < 1$	$\lambda_3 < 1$	$\lambda_4 < 1$	$\lambda_5 = 1$

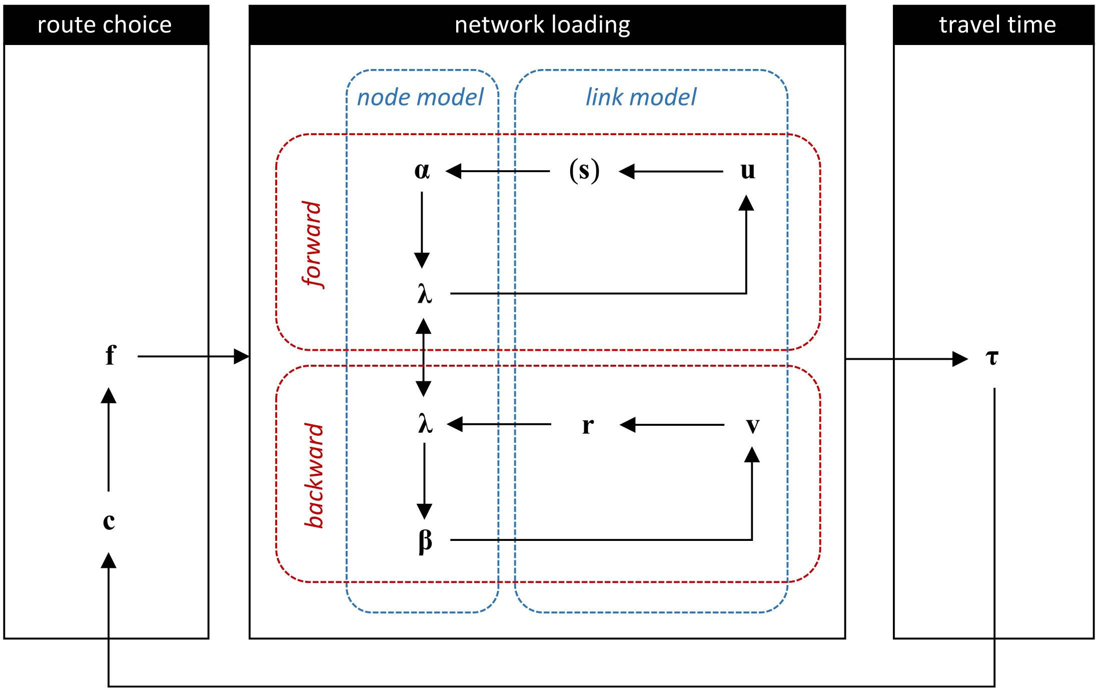
Table 2: Different cases of capacity and storage constraints

	$\beta_a = 1$	$\beta_a < 1$
$\alpha_a = 1$	There are no capacity or storage constraints active. The link is in a free-flow state.	There is only a storage constraint active on a downstream link. The link is in a spillback state.
$\alpha_a < 1$	There is only a capacity constraint active on a downstream link. The link is in a congested state.	There is a capacity constraint active as well as a storage constraint active on a downstream link. The link is in a congested state.

5.2 Forward and backward flow updates

With the introduction of $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_n]_{n\in\mathcal{N}}$ and $\boldsymbol{\beta} = [\boldsymbol{\beta}_n]_{n\in\mathcal{N}}$ we are able to define relationships between variables that either defined as *forward* or *backward*. A forward relationship looks at how path flows affect downstream link inflows and hence sending flows, which in turn affect $\boldsymbol{\alpha}$. A backward relationship looks at how a bottleneck affects outflows rates and hence receiving flows, which in turn affect $\boldsymbol{\beta}$. These relationships are schematically depicted in Figure 7 where we show the entire model structure. The network loading part consists of a node model and a link model. The node model calculates $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The link model calculates the forward propagation of link inflows $\mathbf{u}$ resulting in sending flows $\mathbf{s}$ and the backward propagation of link outflows $\mathbf{v}$ resulting in receiving flows $\mathbf{r}$. In Figure 7 we have put $\mathbf{s}$ in brackets since it holds that $\mathbf{s} = \mathbf{u}$, hence $\mathbf{s}$ can be omitted. We can identify two circular relationships that are interrelated through the node model. The circular forward relationship is essentially a fixed point problem in which we need to find $\boldsymbol{\alpha}$ and $\mathbf{u}$ that are consistent at the network level. This problem is described in Bliemer et al. (2014) as an extension of the work of Bifulco and Crisalli (1998) who were the first to formulate the network loading problem as a fixed point problem. The circular backward relationship is new and makes the problem more challenging. This relationship is not a fixed point problem, however, it can be seen as merely an update of receiving flows along a spillback tree. This observation is important since it provides a clue for designing an algorithm to find a solution to the quasi-dynamic network loading problem; the forward relationship will require an iterative algorithm and is best placed in an inner loop, while the backward relationship will require a recursive algorithm that is best placed in an outer loop.

Figure 7: Model structure



5.3 Special cases

Our capacity and (queue) storage constrained static TA model described by Eqns. (28)–(36) describes a model capable of describing residual queues and spillback by incorporating a full fundamental diagram and a proper node model. Many other static models that have been proposed in the literature and that are used in practice can be seen as a special case of this model.

The capacity constrained model proposed by Bliemer et al. (2014) results if vertical queues are considered instead of horizontal queues. As argued in Bliemer et al. (2017), this model can be derived by assuming that $\gamma_{II,a}(v_a) \to -0$ instead of $\gamma_{II,a}(v_a) \to -\infty$, which results in $\beta_a = 1$ for all $a \in \mathcal{A}$. This replaces λ_a with α_a in travel time calculation (30) and simplifies network loading problem (31)–(36) to the following set of equations:

$$u_{ap}^{w} = \delta_{ap}^{w} f_p^{w} \prod_{a' \in p_a} \alpha_{a'}, \qquad \forall a \in \mathcal{A}, \forall p \in \mathcal{P}^w, \forall w \in \mathcal{W}, \tag{61}$$

$$u_a = \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}^w} u_{ap}^{w}, \qquad \forall a \in \mathcal{A}, \tag{62}$$

$$\varphi_{ab} = \frac{1}{u_a} \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}^w} \delta_{bp}^{w} u_{ap}^{w}, \qquad \forall a \in \mathcal{A}_n^-, \forall b \in \mathcal{A}_n^+, \forall n \in \mathcal{N}, \tag{63}$$

$$\boldsymbol{\alpha}_n = \Gamma\left(\mathbf{u}_n, \mathbf{Q}_n, \boldsymbol{\varphi}_n\right), \qquad \forall n \in \mathcal{N}. \tag{64}$$

The model proposed by Bifulco and Crisalli (1998) is a special case of the network loading problem above. Instead of considering a proper node model as in Eqn. (64), they use the following expression that caps link outflows to the physical link capacity,

$$\alpha_a = \min\left\{1, \frac{Q_a}{u_a}\right\}, \qquad \forall a \in \mathcal{A}, \tag{65}$$

which no longer requires calculating turn flow proportions in Eqn. (63).

Traditional capacity restrained models result if we also ignore capacity constraints (and hence residual queues). Such a model can be derived if we additionally assume that $Q_a \to \infty$ (i.e., no capacity constraints), which results in $\alpha_a = 1$ for all $a \in \mathcal{A}$. Then the average experienced path travel times simplify to

$$\tau_p^w = \sum_{a \in p} \frac{L_a}{\sigma_{I.a}(u_a)}, \qquad \forall p \in \mathcal{P}^w, \forall w \in \mathcal{W}, \tag{66}$$

where $L_a / \sigma_{I.a}(u_a)$ represents the link performance function. For example, the well-known Bureau of Public Roads (BPR) link travel time function follows if we choose the following speed-flow function:

$$\sigma_{I.a}(u_a) = \sigma_a^{\max}\left(1 + \omega_a \left(\frac{u_a}{C_a}\right)^{\zeta_a}\right)^{-1}, \qquad \forall p \in \mathcal{P}^w, \forall a \in \mathcal{A}, \tag{67}$$

where C_a [veh/h] is the (soft) link capacity and ω_a and ζ_a are positive calibration parameters. With $\alpha_a = 1$, network loading problem (61)–(64) further simplifies to the following familiar equation,

$$u_a = \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}^w} \delta_{ap}^w f_p^w, \qquad \forall a \in \mathcal{A}. \tag{68}$$

This equation provides the simplest form of network loading without imposing any capacity or storage constraints. Since link travel times in Eqn. (66) are separable (i.e., link travel times only depend on flow on its own link), VI problem formulation (28) together with Eqn. (29) can be replaced by the following minimisation problem introduced by Fisk (1980),

$$\min_{\mathbf{u},\mathbf{f}} \sum_{a \in \mathcal{A}} \int_{x=0}^{u_a} \frac{L_a}{\sigma_{I,a}(x)} dx + \frac{1}{\mu} \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}^w} f_p^w \log f_p^w, \tag{69}$$

subject to network loading constraints (68) and path flow constraints denoted by $\mathbf{f} \in \Omega$. In case of a deterministic user equilibrium, $\mu \to \infty$ and the original formulation of Beckmann et al. (1956) results in which the second term in Eqn. (67) vanishes.

Note that the hypercritical branch of the fundamental diagram plays no role in any of the special cases discussed here. As far as we are aware, our capacity and storage constrained TA model is the first to take the hypercritical branch of the fundamental diagram into account (via Eqn.(35)).

6. Solution algorithm

Our model consists of a route choice equilibrium problem and of a network loading problem. We first briefly present the algorithm for finding a stochastic user equilibrium solution, and then discuss in more detail the network loading algorithm.

6.1 Route choice equilibrium algorithm

Since the route choice problem formulation is a typical logit-based model similar to existing models, one can readily adopt existing route choice algorithms for path-based models. Algorithm A determines a route choice equilibrium and is adopted from Bliemer et al. (2014).

Algorithm A

Input: A-priori path set $\mathcal{P}^w$ and travel demand D^w for each OD pair *w*, assignment map $\boldsymbol{\delta} = [\delta_{ap}^w]$, and inverse hypocritical flux function $\Phi_{I,a}^{-1}(q)$ for each link $a \in \mathcal{A}$.

Step 0: *Initialisation*. Assume an empty network in which $\mathbf{u}^{(0)} = \mathbf{0}$ and $\mathbf{f}^{(0)} = \mathbf{0}$. Set $c_p^w = \tau_p^w = \sum_{a \in p} L_a / \sigma_a^{\max}$ and set $h := 1$.

Step 1: *Determine intermediate route flows*. Compute the intermediate path flows $\tilde{\mathbf{f}}$ using Equation (3).

Step 2: *Route flow averaging*. Compute the new averaged route flows using the method of successive averages, $\mathbf{f}^{(h)} = \mathbf{f}^{(h-1)} + h^{-g}(\tilde{\mathbf{f}} - \mathbf{f}^{(h-1)})$, for a given $g \in [0,1]$.

Step 3: *Network loading*. Use Algorithm B to conduct network loading, yielding exit flow reduction factors due to capacity and storage constraints, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, from which we can calculate exit flow reduction factors $\boldsymbol{\lambda}$ and link flows $\mathbf{u}$.

Step 4: *Travel cost calculation*. Compute (perceived) average path travel times $\mathbf{c}$ and $\boldsymbol{\tau}$ using Equations (29) and (30).

Step 5: *Convergence check*. Calculate gap *G*,

$$G = \frac{\sum_w \sum_{p \in \mathcal{P}^w} f_p^{w,(h)} (c_p^w + \mu^{-1} \log f_p^{w,(h)} - \pi^{w,(h)})}{\sum_w D^w \pi^{w,(h)}}, \quad \text{where } \pi^{w,(h)} = \min_{p \in \mathcal{P}^w} \left\{ c_p^w + \mu^{-1} \log f_p^{w,(h)} \right\}.$$

If $G < \varepsilon_1$ for some pre-determined small $\varepsilon_1 > 0$, then stop. Otherwise, we set $h := h + 1$ and return to Step 1.

Algorithm B that is called in Step 3 is presented in the next section.

6.2 Network loading algorithm

Now we focus on solving the network loading problem. As discussed in Section 5.2, we have two interrelated problems, namely a problem that describes the forward relationship in which capacity constraints are applied to link inflows, and another problem that describes the backward relationships in which storage constraints are applied to link outflows. The first problem describes relationships on the entire network and is most time consuming to solve, while the second problem describes only local relationships in case of spillback and only makes minor adjustments to the problem. Therefore, it is natural to consider a two-level solution approach in which the forward relationships are considered on the lower (inner) level, while the backward relationships are considered on the upper (outer) level.

With respect to the inner level problem, given path flows $\mathbf{f}$ and given (temporarily fixed) exit flow reduction factors due to storage constraints, $\boldsymbol{\beta}$, we are looking to find link inflows $\mathbf{u}$ and exit flow reduction factors due to capacity constraints, $\boldsymbol{\alpha}$, that solve the following system of equations:

$$u_{ap}^{w} = \delta_{ap}^{w} f_{p}^{w} \prod_{a' \in p_a} \lambda_{a'}, \qquad \forall a \in \mathcal{A}, \forall p \in \mathcal{P}^{w}, \forall w \in \mathcal{W}, \tag{70}$$

$$u_a = \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}^{w}} u_{ap}^{w}, \qquad \forall a \in \mathcal{A}, \tag{71}$$

$$\varphi_{ab} = \frac{1}{u_a} \sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}^{w}} \delta_{bp}^{w} u_{ap}^{w}, \qquad \forall a \in \mathcal{A}_n^{-}, \forall b \in \mathcal{A}_n^{+}, \forall n \in \mathcal{N}, \tag{72}$$

$$\boldsymbol{\alpha}_n = \Gamma\left(\mathbf{u}_n, \mathbf{Q}_n, \boldsymbol{\varphi}_n\right), \qquad \forall n \in \mathcal{N}, \tag{73}$$

$$\lambda_a = \alpha_a \beta_a, \qquad \forall a \in \mathcal{A}. \tag{74}$$

If we set $\beta_a = 1$ for all $a \in \mathcal{A}$, then this set of equations is identical to the network loading problem in Bliemer et al. (2014) and is known to constitute a fixed point problem. Bliemer et al. (2014) present an algorithm can that efficiently solve this fixed point problem on large networks, noting that it is easy to determine potential bottlenecks (namely, links for which holds that $\sum_{w \in \mathcal{W}} \sum_{p \in \mathcal{P}^{w}} \delta_{ap} f_p^w > Q_a$) and hence identify potential spillback roots (links with $\alpha_a < 1$) while all other links and nodes do not need to be considered in finding the fixed point solution. Observe that in case $\beta_a < 1$ for some links, the nature of the problem does not change and therefore this inner level problem can still be readily solved in the exact same fashion.

With respect to the outer level problem, we are looking to update $\boldsymbol{\beta}$ given inflow rates $\mathbf{u}$, turn proportions $\boldsymbol{\varphi}$, and exit flow reduction factors due to capacity constraints, $\boldsymbol{\alpha}$, obtained from the inner level problem. The values for $\boldsymbol{\beta}$ can be found by solving the following system of equations:

$$v_a = \lambda_a u_a, \qquad \forall a \in \mathcal{A}, \tag{75}$$

$$r_a = \min\left\{ v_a + \frac{L_a \Phi_{II}^{-1}(v_a)}{T}, Q_a \right\}, \qquad \forall a \in \mathcal{A}, \tag{76}$$

$$\boldsymbol{\lambda}_n = \Gamma\left(\mathbf{u}_n, \mathbf{r}_n, \boldsymbol{\varphi}_n\right), \qquad \forall n \in \mathcal{N}, \tag{77}$$

$$\beta_a = \frac{\lambda_a}{\alpha_a}, \qquad \forall a \in \mathcal{A}. \tag{78}$$

This problem is not a fixed point problem, but rather can be solved recursively by tracking how far the queues that start at the spillback roots in set $\mathcal{A}^*$ spill back upstream. This is therefore a relatively easy problem to solve in which only a small proportion of the network needs to be considered.

Network loading Algorithm B determines the exit flow reduction factors due to capacity and storage constraints.

Algorithm B

Input: Path flows $\mathbf{f}^{(h)}$, and inverse hypercritical flux function $\Phi_{II,a}^{-1}(q)$ for each link $a \in \mathcal{A}$.

Step 0: *Initialization*. Initialise exit flow reduction due to storage constraints factors $\boldsymbol{\beta}^{(0)} = \mathbf{1}$. Set $j := 1$.

Step 1: *Determine exit flow reduction factors due to capacity constraints*.
- (a) Initialise exit flow reduction factors due to capacity constraints, $\boldsymbol{\alpha}^{(0)} = \mathbf{1}$. Set $m := 1$.
- (b) Calculate exit flow reduction factors $\boldsymbol{\lambda}$ using Eqn. (74).
- (c) Calculate path-specific link inflows using Eqn. (70).
- (d) Calculate link inflows $\mathbf{u}$ using Eqn. (71).
- (e) Calculate turn proportions $\boldsymbol{\varphi}$ using Eqn. (72).
- (f) Calculate exit flow reduction factors due to capacity constraints, $\boldsymbol{\alpha}^{(m)}$, using the node model in Eqn. (73).

(g) If $\frac{1}{|\mathcal{A}|}\left\|\boldsymbol{\alpha}^{(m)}-\boldsymbol{\alpha}^{(m-1)}\right\|<\varepsilon_2$ for some small $\varepsilon_2>0,$ then we have converged to a fixed point with $\boldsymbol{\alpha}=\boldsymbol{\alpha}^{(m)}$. Otherwise, set $m:=m+1$ and return to Step 1(b).

Step 2: *Determine exit flow reduction factors due to storage constraints.*

(a) Denote the set of links to be checked for spillback by $\mathcal{A}',$ where $\mathcal{A}':=\mathcal{A}^*,$ where $\mathcal{A}^*$ is defined in Eqn. (60). Set intermediate reduction factors $\tilde{\boldsymbol{\beta}}=\mathbf{1}$.

(b) Select an $a'\in\mathcal{A}'$.

(c) Calculate outflow $v_{a'}$ using Eqn. (75).

(d) Calculate receiving flow $r_{a'}$ using Eqn. (76).

(e) For upstream node $n',$ determine exit flow reduction factors $\boldsymbol{\lambda}_{n'}$ using Eqn. (77).

(f) For each $a''\in\mathcal{A}_{n'}^{\text{in}},$ check if $\lambda_{a''}<\alpha_{a''}\tilde{\beta}_{a''}$. If so, then the queue on link a' spills back onto link a'' and we set $\tilde{\beta}_{a''}:=\lambda_{a''}/\alpha_{a''}$ and add link a'' to the set of links that needs to be checked for spillback, i.e. $\mathcal{A}':=\mathcal{A}'\cup\{a''\}$.

(g) Link a' has been checked and can be removed from the set, i.e. $\mathcal{A}':=\mathcal{A}'\setminus\{a'\}$.

(h) If $\mathcal{A}'=\varnothing$ then continue with Step 3, otherwise return to Step 2(b).

Step 3: *Reduction factor averaging.* Compute the new averaged exit reduction factors due to stoagra constraints using the method of successive averages, $\boldsymbol{\beta}^{(j)}=\boldsymbol{\beta}^{(j-1)}+j^{-z}(\tilde{\boldsymbol{\beta}}-\boldsymbol{\beta}^{(j-1)}),$ for a given $z\in[0,1]$.

Step 4: *Convergence check.* If $\frac{1}{|\mathcal{A}|}\left\|\boldsymbol{\beta}^{(j)}-\boldsymbol{\beta}^{(j-1)}\right\|<\varepsilon_2$ for some $\varepsilon_2>0,$ then $\boldsymbol{\beta}=\boldsymbol{\beta}^{(j)},$ we calculate $\lambda_a=\alpha_a\beta_a$ for all $a\in\mathcal{A},$ and terminate the algorithm. Otherwise, set $j:=j+1$ and return to Step 1.

In Algorithm B, Step 1 describes the procedure to solve the inner level problem (the ‘forward’ update) in which $\boldsymbol{\alpha}$ is updated and follows the network loading algorithm with vertical queues as outlined in Bliemer et al. (2014). Step 1 is embedded in a new outer level algorithm that updates $\boldsymbol{\beta}$ to account for spillback. Intermediate values $\tilde{\boldsymbol{\beta}}$ are determined in Step 2 by applying storage constraints upstream (i.e., the ‘backward’ update). Note that it does not matter in which order the links in set $\mathcal{A}'$ are processed since we are only interested in the most restricting exit flow constraint. Further note that it is possible that queues can spillback towards links where the queue came from, thereby creating a circular relationship in which outflow restrictions are further decreased. While this may sound like a gridlock situation in which outflows become zero, such a circular relationship is actually not necessarily problematic since we are dealing with average outflow rates. Since the storage capacity $L_a\Phi_{II}^{-1}(v_a)$ on a link is always positive, the (average) receiving flows of a link can never become zero, and hence exit flow reduction factors are always positive. Such a situation will be illustrated with an example in Section 7.2.

Finally, we would like to point out that the averaging in Step 3 is not necessary in all situations. In many cases Algorithm B converges much faster when setting $z=0,$ such that $\boldsymbol{\beta}^{(j)}=\tilde{\boldsymbol{\beta}}$ (i.e., no averaging is applied). However, there may be cases in which such averaging is necessary in order for the algorithm to converge as we will show in an example in Section 7.3.

7. Test cases

In this section we consider several simple example networks to demonstrate our static model with residual queues and spillback. In all case studies we use the node model proposed in Tampère et al. (2011) as function $\Gamma(\cdot)$, and for simplicity we use a triangular fundamental diagram for all links (although any concave shape could be used).

7.1 Corridor network

To illustrate Algorithm B, we show in Table 3 the calculations for the small corridor example (in which there is no route choice) as presented in Section 2. In this example there is no need to apply the averaging in Step 3 of the algorithm and hence we omit this step.

In the first 'forward' update, which considers downstream propagation of sending flows, we solve the fixed point problem between $\boldsymbol{\alpha}$ and $\mathbf{u}$ assuming that there is no spillback yet (since $\boldsymbol{\beta} = \mathbf{1}$) and the resulting solution therefore considers vertical queues. Two exit flow reduction factors due to capacity constraints are smaller than one, indicating that there is a queue on links 3 and 4. This solution constitutes the vertical queuing outcome consistent with the model proposed in Bliemer et al. (2014).

Next, a 'backward' update, which considers upstream propagation of receiving flows, is performed starting from links 3 and 4. The queue on link 3 does not spill back since the receiving flow (4260) is sufficient to accommodate the outflow on link 2 (4000). The queue on link 4 on the other hand will spill back, since the receiving flow (2400) is smaller than the outflow on the upstream link (3600), hence β_3 is reduced to $2400/3600 = 0.667$. This means that the outflow of link 3 will only be 2400, which reduces the receiving flow on this link to 3380, which again is smaller than the outflow on the previous link (4000), hence spillback occurs to link 2 and β_2 is reduced to 0.845. The resulting receiving flow on link 2 (4097) is sufficient to let 4000 flow in, hence there is no spillback onto link 1.

The second 'forward' update ensures that $\boldsymbol{\alpha}$ becomes consistent with $\boldsymbol{\beta}$, which is this case means that only link 5 is an active bottleneck such that there is only a queue that originates from link 4. As a result, in the next 'backward' update only spillback for this queue needs to be considered, which updates the value for β_3. This updated value leads to an updated value for α_4 in the next 'forward' update. The final 'backward' update shows that $\boldsymbol{\beta}$ no longer changes and hence a solution has been found.

The average travel time from A to F can now easily be determined by applying Eqn. (30),

$$\tau^{AF} = \sum_{a=1}^{6} \frac{3}{90} + \frac{1}{2}\left(\frac{1}{0.845} \frac{1}{0.710} \frac{1}{0.750} - 1 \right) = 0.2 + 0.611 = 0.811 \text{ h} = 48.67 \text{ min.} \quad (79)$$

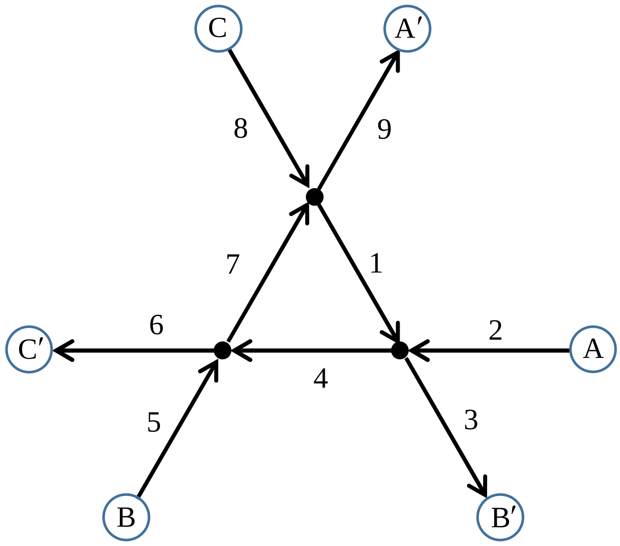
7.2 Network with circular relationships

Next we consider a simple nine-link network as shown in Figure 8. This network is adapted from Bliemer et al. (2014) and illustrates how multiple routes interact. There are three OD pairs, namely (A,A'), (B,B'), and (C,C'), each having only a single route and a demand of 2000 veh/h during one hour. Links 1, 4, and 7 are assumed to have a length of 1 km, while the links coming from the origins are assumed to be sufficiently long that queues do not spill back into the origin. Each link is assumed to be represented by a triangular fundamental diagram with a maximum speed of 100 km/h, a capacity of 2000 veh/h, and a jam density of 180 veh/km. We apply Algorithm B without averaging Step 3.

Table 3: Iterations for solving the corridor network

	a	1	2	3	4	5	6
Initialisation	β_a	1.000	1.000	1.000	1.000	1.000	1.000
	α_a	1.000	1.000	1.000	1.000	1.000	1.000
Forward (1)	u_a	4000	4000	4000	4000	4000	4000
	α_a	1.000	1.000	0.900	0.450	0.450	1.000
	u_a	4000	4000	4000	3600	1620	729
	α_a	1.000	1.000	0.900	0.500	1.000	1.000
	u_a	4000	4000	4000	3600	1800	1800
	α_a	1.000	1.000	**0.900**	**0.500**	1.000	1.000
Backward (1)	v_a			3600			
	r_a			4260			
	β_a		1.000				
	v_a				1800		
	r_a				2400		
	β_a			**0.667**			
	v_a			2400			
	r_a			3380			
	β_a		**0.845**				
	v_a		3380				
	r_a		4097				
	β_a	1.000					
Forward (2)	u_a	4000	4000	3380	2028	1014	1014
	α_a	1.000	1.000	1.000	0.887	1.000	1.000
	u_a	4000	4000	3380	2253	2000	2000
	α_a	1.000	1.000	1.000	0.799	0.900	1.000
	u_a	4000	4000	3380	2253	1800	1620
	α_a	1.000	1.000	1.000	0.799	1.000	1.000
	u_a	4000	4000	3380	2253	1800	1800
	α_a	1.000	1.000	1.000	**0.799**	1.000	1.000
Backward (2)	v_a				1800		
	r_a				2400		
	β_a			**0.710**			
	r_a			2400			
	v_a			3380			
	β_a		**0.845**				
	r_a		3380				
	v_a		4097				
	β_a	1.000					
Forward (3)	u_a	4000	4000	3380	2400	1918	1918
	α_a	1.000	1.000	1.000	0.750	0.938	1.000
	u_a	4000	4000	3380	2400	1800	1689
	α_a	1.000	1.000	1.000	0.750	1.000	1.000
	u_a	4000	4000	3380	2400	1800	1800
	α_a	1.000	1.000	1.000	**0.750**	1.000	1.000
Backward (3)	v_a				1800		
	r_a				2400		
	β_a			**0.710**			
	v_a			2400			
	r_a			3380			
	β_a		**0.845**				
	v_a		3380				
	r_a		4097				
	β_a	1.000					

Figure 8: Network with circular relationships



The bottlenecks are the links 1, 4, and 7 where there is a travel demand of 4000 veh/h while the capacity is only 2000 veh/h. This means that queues form on each of these links, which eventually spill back onto links 2, 5, and 8. Further, this queue spillback also reduces the inflow into these links, and as a result also reduces the outflow, creating a circular relationship in which flows are reduced until the process stabilises. This leads to an average outflow rate on links 3, 6, and 9 is 98.5 veh/h, while the average outflow rate on all other links is 257.9 veh/h.

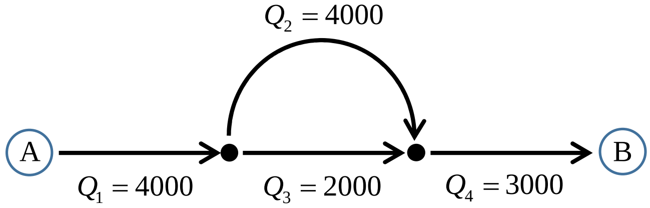
7.3 Example with unstable queues

In the previous examples we showed that Algorithm B converges to a stable network loading solution without the need for averaging Step 3 in a simple corridor case as well as in a heavily congested situation in which queues spill back in a circular fashion. Now consider the network in Figure 9 in which we have indicated the capacities of each link. Suppose that the path flows on both the top and bottom route are 2000 veh/h. It is immediately clear that the link 4 is a potential bottleneck, and link 3 will experience a queue. We assume that link 3 has a length of 2 km, a maximum speed of 100 km/h, and a jam density of 180 veh/km. The characteristics of the other links are not relevant in this example.

At the start of the demand period the path flows (in total 4000 veh/h) will be able to exit link 1, such that 2000 veh/h will flow into links 2 and 3. Since link 2 has twice the capacity of link 3, all 2000 veh/h will be able to exit link 2, while only 1000 veh/h flow from link 3 can exit (since the node model that we consider allocates priorities according to capacity), i.e. $\alpha_3 = 0.5$. As a result, a queue will start building up on link 3. The storage capacity with an outflow rate of 1000 veh/h equals 200 veh, such that the average receiving flow rate into link 3 is 1200 veh/h, which is larger than the 2000 veh/h that would like to enter link 3 from link 1. Therefore, after a short period of time the queue will spill back onto link 1. This queue will limit the flow entering link 2 from link 1, namely in the same ratio of 1200/2000, such that only 1200 veh/h enter link 2 on average. Therefore, once the flow spills back, the flows on links 2 and 3 are both reduced to 1200 veh/h ($\beta_1 = 0.6$), which can enter link 4 without any problem, and hence no queue exists on link 3. In other words, spillback can only occur if link 3 has a queue, but if there is spillback, link 3 cannot have a queue. Therefore, without averaging Step 3 in Algorithm B, there will be flip-flopping between a solution in which $\alpha_3 = 0.5$ and $\beta_1 = 1$, and another solution in which $\alpha_3 = 1$ and $\beta_1 = 0.6$ (while all other reduction factors are equal to one).

In reality, link 2 will form a queue, and once the queue spills back onto link 1, the inflow into link 2 will be constraint but the queue will not immediately disappear (since it takes some time until vehicles have traversed link 2). After some period of time, the queue on link 3 starts shrinking and will no longer spill back onto link 1. At this moment the flow into link 2 will increase again and the cycle repeats itself. So in a dynamic context the queue actually increases and decreases over time repeatedly. In the static case, we would like to find the average flow rates. The average inflow rates can be found by applying the averaging Step 3 in Algorithm B, which converges after 15 iterations to a stable solution with $\alpha_3 = 0.916$ and $\beta_1 = 0.783$. In other words, as expected both links 1 and 3 have reduced exit flows in the static solution.

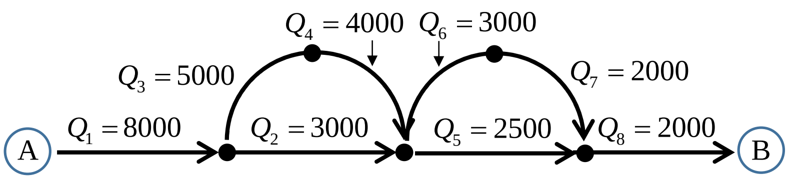
Figure 9: Network with unstable queues



7.4 Simple network with route choice

The example in Figure 10 shows a more general network in which we consider four routes between origin A and destination B with a travel demand of 8000 veh/h during one hour. These routes interact via one diverge node, one cross-node, and one merge node. Each link has a length of 2 km, a maximum speed of 100 km/h, and has a capacity as indicated in Figure 10. Further, links 5, 7, and 8 have a jam density of 180 veh/km, while other links have a jam density of 360 veh/h except for link 1 that is assumed to have a jam density of 1800 veh/h (which is large enough to ensure that the queue does not spillback into the origin).

Figure 10: Simple network with route choice



A scale parameter of $\mu = 5$ was used in the perceived travel time function, and a convergence threshold in Algorithm A of $\varepsilon_1 = 10^{-5}$ was chosen. Table 4 lists the equilibrium path flows and (perceived) travel times, as well as the associated link inflow, outflow rates, and exit reduction factors. The only two links that do not have a queue are links 6 and 8. Path travel times can again be computed using Eqn. (30). For example, path 1-2-5-8 has a travel time of

$$\tau^{AB}_{1-2-5-8} = \sum_{a=1}^{4} \frac{2}{100} + \frac{1}{2}\left(\frac{1}{0.436}\frac{1}{0.677}\frac{1}{0.833}\frac{1}{1.000} - 1 \right) = 0.080 + 1.530 = 1.610 \text{ h.} \quad (80)$$

The perceived travel time c_p^w can be computed using Eqn. (29)

Table 4: Equilibrium network outcomes of the simple network with route choice

a	u_a	v_a	λ_a	p	f_p	τ_p^w	c_p^w
1	8000	3491	0.436	1-2-5-8	1935	1.610	3.124
2	1526	1034	0.677	1-3-4-5-8	2487	1.582	3.146
3	1965	1878	0.956	1-2-6-7-8	1562	1.653	3.124
4	1978	1378	0.734	1-3-4-6-7-8	2015	1.602	3.124
5	1333	1111	0.833				
6	1079	1079	1.000				
7	1079	889	0.824				
8	2000	2000	1.000				

8. Conclusions and discussion

In this paper we have established a path-based static TA model that can represent residual queues and spillback on general networks. This was achieved by taking a first order dynamic TA model as a starting point and make static temporal assumptions to obtain average link inflow and outflow rates over the duration of the travel demand. The model takes the same (concave) fundamental diagram and node model as input as state-of-the-art dynamic counterparts, and as such guarantees maximum consistency with respect to spatial interaction assumptions.

The model has a concise mathematical problem formulation, and we have shown that several existing static models are special cases of this rather general formulation. A key challenge is the development of an algorithm that can solve the problem, mainly due to the fact that there are not only downstream interactions (i.e., path flows that influence link flows downstream), but also upstream interactions (i.e., queues spilling back to upstream links). In order to solve the problem, we separated these downstream and upstream interactions into separate variables that we solve by temporarily keeping either the upstream or downstream impacts fixed. We also showed that there may be cases of unstable queues in which there is a need to use an averaging scheme in order to find the average flow rates on the network that represent queues that grown and shrink over the demand period.

While we believe that our model is an important step towards developing static models that can more realistically describe queues and associated travel times on the network, we only see it as a first step. There are still many unanswered questions with respect to existence and uniqueness of the solution, and convergence of the algorithm. Furthermore, the algorithm that we proposed has not been optimised for efficiency. While the algorithm is relatively fast and we believe it is applicable to medium scale networks, for large scale transportation networks more research is needed to reducing the computation time.

References

Bakker, D., P.H. Mijjer, and F. Hofman (1994) QBLOK: An assignment technique for modeling the dependency between bottlenecks and the prediction of 'grid lock'. *Proceedings of Colloquium Vervoersplanologisch Speurwerk*, Delft, pp. 313-332.

Beckmann, M., C. McGuire, and C. Winsten (1956) *Studies in the Economics of Transportation*. Yale University Press, Connecticut, USA.

Bell, M.G.H. (1995) Stochastic user equilibrium in networks with queues. *Transportation Research Part B*, Vol. 29, No. 2, pp. 115–112.

Bifulco, G., and U. Crisalli (1998) Stochastic user equilibrium and link capacity constraints: formulation and theoretical evidences. *Proceedings of the European Transport Conference.*

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

Bliemer, M.C.J., M.P.H. Raadsen, L.J.N. Brederode, M.G.H. Bell, L.J.J. Wismans, and M.J. Smith (2017) Genetics of traffic assignment models for strategic transport planning. *Transport Reviews*, Vol. 37(1), pp. 56-78.

Bliemer, M.C.J., M.P.H. Raadsen, E.S. Smits, B. Zhou, and M.G.H. Bell (2014) Quasi-dynamic traffic assignment with residual point queues incorporating a first order node model. *Transportation Research Part B*, Vol. 68, pp. 363-384.

Bundschuh, M., P. Vortisch, and T. Van Vuuren (2006) Modelling queues in static traffic assignment. *Proceedings of the European Transport Conference.*

Bureau of Public Roads (1964) *Traffic Assignment Manual*. U.S. Dept. of Commerce, Urban Planning Division, Washington D.C.

Chen, H.-K. (1999) *Dynamic travel choice models: a variational inequality approach*. Springer-Verlag Berlin-Heidelberg.

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

Lam, W.H.K., and Y. Zhang (2000) Capacity-constrained traffic assignment in networks with residual queues. *Journal of Transportation Engineering*, Vol. 126, No. 2, pp. 121–128.

Larsson, T., and M. Patriksson (1995) An augmented Lagrangean dual algorithm for link capacity side constrained traffic assignment problems. *Transportation Research Part B*, Vol. 29, pp. 433-455.

Smith, M.J. (1987) Traffic control and traffic assignment in a signal-controlled network with queueing. *Presented at the 10th International Symposium on Transportation and Traffic Theory (ISTTT)*, Boston, MA.

Smith M.J. (2013) A link-based elastic demand equilibrium model with capacity constraints and queueing delays. *Transportation Research Part C*, Vol. 29, pp. 131-147.

Smith, M., W. Huang, and F. Viti (2013) Equilibrium in capacitated network models with queueing delays, queue-storage, blocking back and control. Presented at the 20th International Symposium on Transportation and Traffic Theory (ISTTT), Noordwijk, the Netherlands, *Procedia – Social and Behavioral Sciences*, Vol. 80, pp. 860-879.

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

Yang, H., and S. Yagar (1994) Traffic assignment and traffic control in general freeway- arterial. *Transportation Research Part B*, Vol. 28, No. 4, pp. 463-486.

Yang, H., and S. Yagar (1995) Traffic assignment and signal control in situated road networks. *Transportation Research Part A*, Vol. 29, No. 2, pp. 125-139.

Van Vliet, D. (1982) SATURN – a modern assignment model. *Traffic Engineering and Control*, Vol. 23, No. 12, pp. 578-581.