WORKING PAPER

ITLS-WP-17-05(2)

Continuous-time general link transmission model with simplified fanning,

Part II: Event-based algorithm for networks

By
Mark P.H. Raadsen and Michiel C.J. Bliemer

Institute of Transport and Logistics Studies (ITLS),
The University of Sydney Business School, Sydney, Australia

March 2017

ISSN 1832-570X

INSTITUTE of TRANSPORT and LOGISTICS STUDIES
The Australian Key Centre in
Transport and Logistics Management

The University of Sydney
Established under the Australian Research Council's Key Centre Program.

NUMBER: Working Paper ITLS-WP-17-05(2)

TITLE:

Continuous-time general link transmission model with simplified fanning,

Part II: Event-based algorithm for networks

ABSTRACT:

In this paper a novel solution algorithm is proposed for solving general first order dynamic network loading (DNL) problems in general transport networks. This solution algorithm supports any smooth non-linear two regime concave fundamental diagram and adopts a simplified fanning scheme. It is termed eGLTM (event-based General Link Transmission Model) and is based on a continuous-time formulation of the kinematic wave model that adapts shockwave theory to simplify expansion fans. As the name suggests eGLTM is a generalisation of eLTM, which is a special case that solves the simplified first order model assuming a triangular fundamental diagram. We analyse the impact of modelling delay in the hypocritical branch of the fundamental diagram to assess the differences between the two models. In addition, we propose an additional stream of mixture events to propagate multi-commodity flow in event based macroscopic models, which makes both eLTM and eGLTM suitable for dynamic traffic assignment (DTA) applications. The proposed solution scheme can yield exact solutions as well as approximate solutions at a significantly lesser cost. The efficiency of the model is demonstrated in a number of case studies. Furthermore, different settings for our simplified fanning scheme are investigated as well as an extensive analysis on the effect of including route choice on the algorithms computational cost. Finally, a large scale case study is conducted to investigate the suitability of the model in a practical context and assess its efficiency compared to the simplified first order model.

KEY WORDS: ***Continuous-time grid free solution scheme, general link transmission model, macroscopic dynamic network loading, simplified fanning, event-based algorithm***

AUTHORS: **Raadsen and Bliemer**

Acknowledgements: We would like to thank Aleix Ruiz de Villa for his valuable input regarding the representation of path information within an implicit event based context. We thank DAT.Mobility for providing the StreamLine framework and OmniTRANS software, allowing us to implement the eGLTM prototype. We thank Veitch Lister Consultancy for providing the Gold Coast network. This research is partially funded by Australian Research Council Linkage Project LP130101048.

CONTACT: INSTITUTE OF TRANSPORT AND LOGISTICS STUDIES (H73)

The Australian Key Centre in Transport and Logistics Management

The University of Sydney NSW 2006 Australia

Telephone:	+612 9114 1824
E-mail:	business.itlsinfo@sydney.edu.au
Internet:	http://sydney.edu.au/business/itls

DATE: March 2017

1 Introduction

Each dynamic traffic assignment (DTA) model typically consists of two components. The first handles route, and possibly, departure time choice. The second is responsible for loading traffic onto the network conditional on the route and/or departure time choice. Often an iterative simulation-based approach is adopted to reach some form of (near) equilibrium state as the final result.

In this paper we focus mainly on the loading of traffic onto the network in a time dependent fashion known as dynamic network loading (DNL). Route choice is supported within the DNL formulation, but is assumed to be readily available in the form of path flows varying by departure time, see Figure 1.

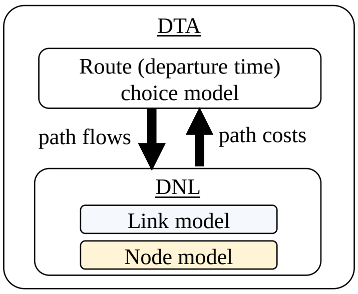


Figure 1: General framework for solving simulation based DTA.

Different types of DNL formulations exist that are typically classified as macroscopic, mesoscopic or microscopic (Peeta and Ziliaskopoulos, 2001). The latter two load individual discrete vehicles (or sometimes packets of vehicles) onto the network whereas a macroscopic DNL represents traffic flow as a continuum (often the analogy with a fluid is made). In this paper we focus on macroscopic models. Each DNL encompasses a link model and a node model. The link model propagates traffic flow across homogeneous road segments while the node model distributes flow across intersections where links compete for available capacity.

Interestingly, most well-known macroscopic DNL approaches adopt the same underlying mathematical problem formulation, but apply different solution schemes (algorithms) to solve the problem. This paper considers the well-known kinematic wave model and proposes a novel general event-based solution approach. Our approach embeds multiple solution schemes within a single framework. Differences between the considered solution schemes are to be found regarding: (i) The adopted fundamental diagram (FD), (ii) modelling of expansion fans, (iii) exactness of the solution, and (iv) the tracking of flow composition, i.e. path information. Our solution approach is based on the continuous-time general link transmission model formulation proposed in Part I of this study (Bliemer and Raadsen, 2017). In order to include smooth nonlinear concave FDs, they propose two on-the-fly linearization techniques to simplify expansion fans that allow exact non-entropic solutions to the kinematic wave model. One of these approaches adopts and extends shockwave theory, which Bliemer and Raadsen argue has most potential for developing efficient event-based algorithms. In this paper we explore and propose such algorithms. The companion paper Bliemer and Raadsen (2017) will from here on forward be referred to as Part I.

Most approaches in simulation-based DNL rely on some form of discretisation, either in time, space, or both. The amount of error introduced is a result of the choices made on the level of detail of the solution grid. DNL algorithms capable of yielding exact solutions are often referred to as grid-free approaches because they have no pre-determined grid points in time and space. All variations of the solution scheme presented in this paper are considered grid-free. The implicit event-based approach proposed in this paper generalises and extends the procedure described in Raadsen et al. (2016).

1.1 Dynamic Network loading in the literature

Macroscopic DTA emerged in an effort to address limitations in existing static assignment methods. Adding a temporal component to the model brought it closer to reality while at the same time introducing new challenges. Like the first DTA model (Merchant and Nemhauser, 1978), most early dynamic models represented extensions to static assignment methods and focussed on finding analytical model representations (e.g., Carey, 1987; Friesz et al., 1989; Janson, 1991).

In the 1990s several significant advances were made with respect to addressing drawbacks of analytical methods, such as the inherent difficulty of modelling physical queues and spillback (Daganzo, 1995) as well as the large computational effort required to solve such models. This restricted model applications to small or medium sized networks. For this reason simulation-based schemes gained more traction, such as Messmer and Papageorgiou (1990) who aim to solve the model proposed by (Payne (1971) and the cell transmission model (CTM) proposed by Daganzo (1994, 1995), that solves the kinematic wave model proposed by Lighthill and Whitham (1955) and Richards (1956). These simulation-based methods can be applied to reasonably large networks and do, in most cases, include spillback effects.

The kinematic wave model underpins the majority of today's simulation-based solution schemes. Besides CTM that requires discretising space and time to find an approximate solution, link transmission models[1] (LTMs) have been proposed that only require time discretisation (Yperman et al., 2005; Yperman, 2007; Gentile, 2010) and as such can obtain a solution to the kinematic wave model more efficiently and accurately. Himpe et al. (2016) proposed a related iterative algorithm in which the discretisation scheme is less constrained, allowing to find an approximate solution more quickly.

While LTM has become a popular approach in practice, the properties of LTM remained largely unknown. In recent years this gap has been bridged. LTM can be directly derived from the Lax-Hopf solution to the Hamilton-Jacobi formulation of the kinematic wave model (Han et al., 2012; Jin, 2014; Bliemer and Raadsen, 2017). The Lax-Hopf formula can be used to construct analytical solutions to the kinematic wave model in continuous time for a single link (Daganzo, 2005; Claudel and Bayen, 2010; Mazaré et al., 2011) and as such LTM is now sometimes considered a time-discretised version of this semi-analytical continuous-time solution scheme.

1.2 Event-based dynamic network loading algorithms

Macroscopic event-based methods typically track changes in traffic states over space and/or time. We distinguish between two types of event-based methods. First, there are *explicit* event-based methods known as wave-front or front tracking algorithms (e.g., Henn, 2003; Coclite et al, 2005; Chen, 2009; Wu and Liu, 2011). We categorise these models as explicit, since they track each (shock)wave explicitly in space and time. Whenever a shockwave interacts with another shockwave new events are generated to propagate the newly formed shockwave(s) etc. Most explicit event-based algorithms described in the literature have not been applied in a network context but rather focus on a single link. Secondly, there are *implicit* event-based algorithms. This algorithm type relies on tracking flow rate changes on link boundaries only (Raadsen et al., 2016) and is based on cumulative vehicle flows. The concepts underpinning this type of algorithm were originally introduced by Newell (1993). This scheme is termed implicit since it no longer requires to track each individual (shock)waves over space, making it more efficient. The algorithm proposed in this paper is of the *implicit* event-based type.

1.3 Contribution

[1] We would like to point out that in our terminology, technically CTM and (G)LTM are not models, but rather solution schemes to solve the kinematic wave model.

In this paper we make a number of contributions. Firstly, we generalise the event-based LTM algorithm (referred to as eLTM) proposed in Raadsen et al. (2016). This algorithm only supports a triangular FD which is known to overestimate vehicle speeds and underestimate link travel times especially around the flow saturation point. We propose an event-based General LTM algorithm (referred to as eGLTM) in which any concave two-regime FD with smooth nonlinear branches is supported, which includes eLTM as a special case.

The model formulated in Part I only considers a single link and single commodity flow. . In this paper we extend Part I by making the DNL suitable for general networks and multi-commodity flow. As far as the authors are aware, there exists no implicit event-based methodology to support multi-commodity flow in macroscopic models. A framework is proposed that makes a clear distinction between flow propagation and mixture (multi-commodity) propagation on both the link and the node level. For the remainder of the paper we make the simplifying assumption that multi-commodity flow is not tracked by paths, but by departure time period. Raadsen et al. (2016) also distinguished flow by departure time period, but had to make the strong assumption that the path flows over the entire network vary proportionally over time, which does not support within-iteration route choice. This assumption will be relaxed in this paper, while still allowing multi-commodity flow to be modelled via splitting rates, which significantly reduces computation cost compared to path-based approaches in large scale strategic planning applications, albeit at the cost of (some) violation regarding path flow consistency. While we focus on models that describe route choice using splitting rates, we will demonstrate that the general theoretical framework is equally capable of incorporating path-based or destination-based approaches.

Thirdly, the exact multi-commodity eGLTM solution scheme is relaxed to allow for approximate solutions. The trade-off between information loss and computational cost can be defined in a flexible way by the user. Case studies are provided to demonstrate the efficiency of the approximate solutions, and we propose suitable settings based on these studies.

In order to achieve this, we make the following four additional simplifying assumptions in the remainder of this paper: (i) Paths flows are assumed to be temporarily stationary within each departure time period, (ii) a fixed set of (pre-generated) paths is available throughout the simulation/iteration, (iii) expansion fans are simplified using on-the-fly *D*-step inner linearization as presented in Part I (referred to as simplified fanning), and (iv) we consider only a single vehicle type.

1.4 Outline

This paper is organised as follows. Section 2 introduces notation on both topology and the FD. Section 3 discusses the link and node model formulation; the flow propagation formulation of Part I is rewritten by making explicit assumptions about the simplified fanning scheme. In addition the mixture propagation formulation is introduced to support a multi-commodity approach. Section 4 takes the model formulations of Section 3 and proposes their implicit-event based equivalents. Section 5 presents the grid-free solution algorithm based on the event-based formulations. Section 6 discusses numerical results to analyse the performance of the solution scheme and analyses the effects of choosing different settings for yielding approximate results. This is followed by conclusions and future research directions in Section 7.

2 Network and fundamental diagram definition

Notation largely follows Part I[2]. Consider general network $\mathcal{G}=(\mathcal{N},\mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ the set of directional links. Let $\mathcal{O}\subseteq\mathcal{N}$ be the set of origin nodes. Each node $n\in\mathcal{N}$ has a set of incoming links $\mathcal{A}_n^-$ and outgoing links $\mathcal{A}_n^+$. In addition to link length L_a (km), the following five

[2] We briefly discuss the main concepts described in Part I to introduce notation required to rewrite the original formulation.

calibrated parameters are assumed present for each link $a \in \mathcal{A}$: The maximum flow rate (i.e., capacity) $q_a^{\max}$ (veh/h). The maximum (jam) density $k_a^{\max}$ (veh/km). The critical density (i.e., the density at capacity) k_a^{crit} (veh/km). The minimum (backward) wave speed $\gamma_a^{\min}$ (km/h), and maximum (forward) wave speed $\gamma_a^{\max}$ (km/h). These parameters constrain the possible shapes of the FD. A possible shape is depicted in Figure 2(a). In general it holds that free speed $\sigma_a^{\max}$ (km/h) is equal to the maximum wave speed, $\sigma_a^{\max} = \gamma_a^{\max}$, and that speed at capacity equals $\sigma_a^{\text{crit}} = q_a^{\max} / k_a^{\text{crit}}$ (km/h). In the part that follows we omit sub-index *a* for notational convenience and readability.

Each FD is split into two regimes, a hypocritical branch representing all uncongested states and a hypercritical branch for all congested states (see e.g., Cascetta, 2009). For the hypocritical regime we define strictly increasing concave flux function $\Phi_I : k \to q$ and its inverse $\Phi_I^{-1} : q \to k$, with density interval $[0, k^{\text{crit}}]$ and flow rate interval $[0, q^{\max}]$. For the hypercritical regime we define strictly decreasing concave flux function $\Phi_{II} : k \to q$ and its inverse $\Phi_{II}^{-1} : q \to k$, with density interval $[k^{\text{crit}}, k^{\max}]$ and flow rate interval $[0, q^{\max}]$. Since LTM formulations are traditionally expressed in terms of flow rates instead of densities, we will use the inverse flux functions. Wave speed *w* (km/h) represents the slope of the FD at each point. A (steady state) flow rate propagates with this (characteristic) wave speed and is vital in creating our general solution scheme. Note that when a triangular FD is adopted only a single wave speed exists per regime. In the general case this no longer holds, i.e. the range of wave speeds depends on the shape of the FD. To accommodate this we define functions $\gamma_I : q \to w$ and $\gamma_{II} : q \to w$ to denote wave speeds on the hypocritical and hypercritical branch, respectively. If the hypo- and hypercritical branches of the flux function are continuously differentiable, then $\gamma_I(q) = 1 / \left(d\Phi_I^{-1}(q) / dq \right)$ and $\gamma_{II}(q) = 1 / \left(d\Phi_{II}^{-1}(q) / dq \right)$.

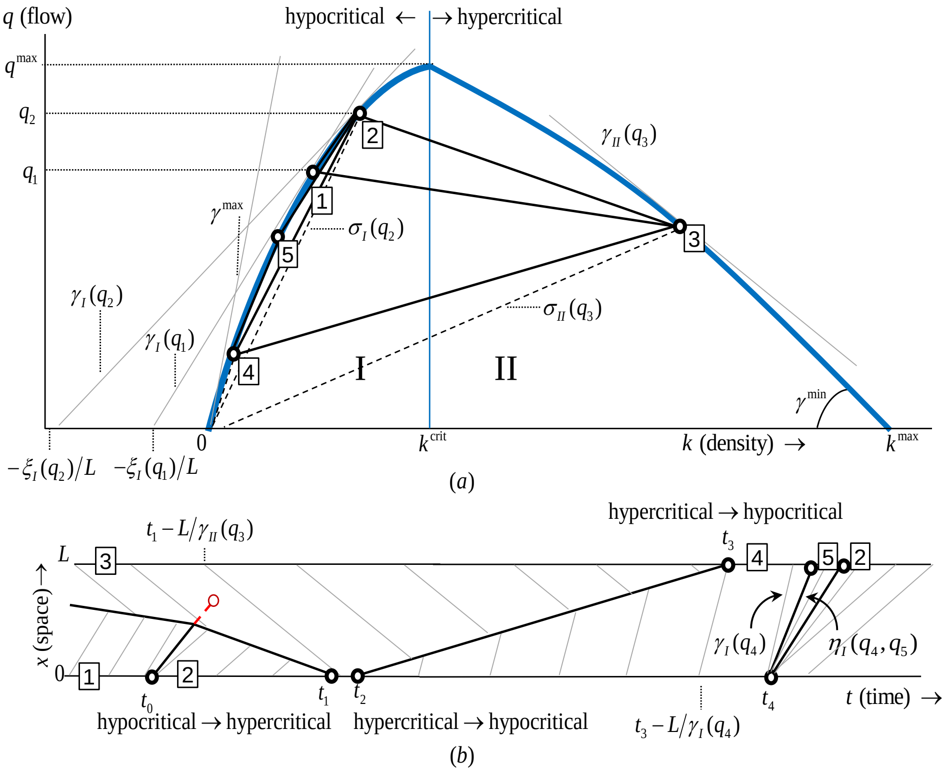


Figure 2: (a) Schematic two regime FD with three flow states, (b) example space-time diagram of flow states realised on a link.

Shockwaves separate regions of two flow rates where their kinematic waves meet. See for example the shockwave in Figure 2(b) between t_2 and t_3: the shockwave separates traffic states with flow rates $q_{3,}$ and q_4 and the shockwave speed corresponds to the slope of the line one can draw on the FD between these same two flow rates. In general we denote the speed of the shockwave between two hypocritical flow rates as $\eta_I(q', q'')$ and two hypercritical flow rates through $\eta_{II}(q', q'')$, which are defined as

$$\eta_I(q',q'') = \frac{q''-q'}{\Phi_I^{-1}(q'')-\Phi_I^{-1}(q')}, \quad \eta_{II}(q',q'') = \frac{q''-q'}{\Phi_{II}^{-1}(q'')-\Phi_{II}^{-1}(q')}, \quad q'' \neq q', \quad q',q'' \in [0,q^{\max}]. \tag{1}$$

In case of an expansion fan, which only appears in case of a flow increase, shockwaves are absent. However, modelling such fans is cumbersome due to the infinite number of waves, and therefore flow rates, they generate. Here, we adopt simplified fanning through on-the-fly inner linearization as described in more detail in Part I, which keeps the original flux functions as input. This approach is more flexible than replacing the original FD with a fixed piecewise linear or triangular FD. Our approach can approximate the true FD shape to any level of accuracy defined by the user and if needed in a dynamic fashion. Simplified fanning boils down to decoupling the direct link between a flow rate and its characteristic wave in the expansion fan and replace this with one or more "representative" waves carrying the flow rate change(s). The number of "representative" waves can be defined by the user (the more, the closer the true shape of the FD will be followed). Consider time instant t_4 in the space-time example of Figure 2(b) in which the flow rate changes from traffic state 4 to traffic state 2. Here, the expansion fan is replaced with two representative waves that model the transition via intermediate traffic state 5. These two representative waves have the same speed as shockwave speeds $\eta_I(q_4,q_5)$ and $\eta_I(q_5,q_2)$, respectively. This is an example of simplified fanning with a 2-step approach. This approach simplifies solving the model considerably while still including fanning effects.

Finally, we denote the discrepancy between wave pace $1/\gamma$ and vehicle pace $1/\sigma$ through hypocritical function $\xi_I(q)$ and hypercritical function $\xi_{II}(q)$, they represent the total number of vehicles a characteristic wave would pass while traversing a link.

$$\xi_I(q) = Lq\left(\frac{1}{\gamma_I(q)} - \frac{1}{\sigma_I(q)}\right), \quad \xi_{II}(q) = Lq\left(\frac{1}{\sigma_{II}(q)} - \frac{1}{\gamma_{II}(q)}\right), \tag{2}$$

with hypocritical and hypercritical speed functions $\sigma_I(q) = q/\Phi_I^{-1}(q)$, and $\sigma_{II}(q) = q/\Phi_{II}^{-1}(q)$, respectively.

3 Continuous-time DNL formulation

Part I established a continuous-time DNL formulation for the propagation of single-commodity flow on a link. In this section we extend this formulation in two ways. First, we extend the model to a multi-commodity approach by introducing the concept of mixture propagation. It introduces a mixture propagation model on the link level and a mixture distribution model on the node level. Figure 3 depicts the DNL framework that accommodates this, making the model suitable for general networks.

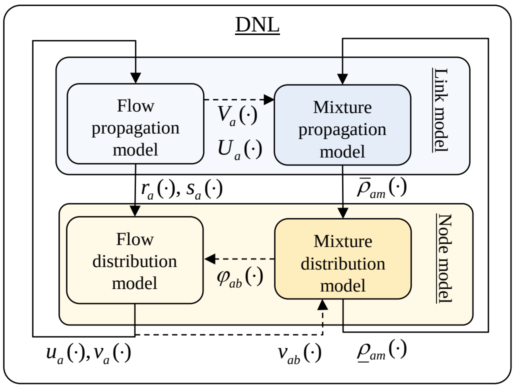


Figure 3: General model framework for implicit-event based multi-commodity DNL solution schemes.

It separates flow based and mixture based components by design. We define flow mixture as "what the flow rate is composed of", for example: The percentage of the flow departed during a certain time interval or, the percentage of flow following a particular path. The dashed lines in Figure 3 indicate a dependency between flow and mixture components while solid lines show dependencies between the link and node model. The shown variables will be introduced and explained later in this section. Secondly, we explicitly assume simplified fanning is modelled via *D*-step inner linearization. This allows us to rewrite the flow propagation formulation of Part I such that, on the one hand it becomes more concise, while on the other hand aids the conversion to the final event-based formulation discussed in Section 0.

We first discuss the link model, followed by the node model

3.1 Link model

The link model is responsible for propagating traffic states upstream and downstream along a link, as well as propagating vehicle mixtures downstream. At the link boundaries, this information will be passed on to the node model via sending and receiving flows. Individual components of the link model will be described in the following subsections. Again in all variables we omit link index *a* for notational convenience.

3.1.1 Flow propagation

Consider upstream link boundary inflow rate $u(t)$ (veh/h) and downstream link boundary outflow rate $v(t)$ (veh/h) at time $t \in [0,T],$ where T denotes the simulation end time. Cumulative inflow $U(t)$ (veh) and cumulative outflow $V(t)$ (veh) are piecewise linear under the adopted simplified fanning scheme such that

$$\begin{aligned} &U(t) = U(t_i) + u_i(t - t_i), \quad \text{if } t_i \le t \le t_{i+1}, \quad i = 1, \dots I, \\ &V(t) = V(t_j) + v_j(t - t_j), \quad \text{if } t_j \le t \le t_{j+1}, \quad j = 1, \dots J, \end{aligned} \tag{3}$$

with u_i denoting the i^{th} piecewise stationary inflow rate that started at time t_i and v_j denoting the j^{th} piecewise stationary outflow rate started at time t_j, resulting in $u(t) = u_i,\ t_i \le t < t_{i+1},$ and $v(t) = v_j,\ t_j \le t < t_{j+1}.$

Recall the final lower envelope problem formulation established in Part I. This formulation embeds *D*-step simplified fanning within the model, but it is not restricted to an inner linearization method. By imposing this restriction, we use this section to simplify this formulation. We first consider the situation for finding the downstream potential cumulative outflow $\vec{U}(t)$ in case it is dictated by an earlier (hypocritical) upstream link boundary state. Following Part I, the “arrow” notation is adopted whenever a type of projection (of a time period, a flow rate, a cumulative, etc.) on the opposite link boundary is denoted. Given cumulative inflows $U(t)$; $\vec{U}(t)$ is the lower envelope of projected cumulative segments of earlier upstream traffic flow states given by

$$\vec{U}(t) = \min\left\{ \min_{i \in \mathcal{I}(t)} \vec{U}_i(t), \min_{i \in \mathcal{I}_F(t)} \vec{F}_i(t \mid \cdot) \right\}, \tag{4}$$

where $\vec{U}_i(t)$ and $\vec{F}_i(t \mid \cdot)$ are downstream projections of upstream cumulative inflow segments. While $\vec{U}_i(t)$ is a “regular” projection (in fact, it is a simple translation) of $U(t)$, $\vec{F}_i(t \mid \cdot)$ represents the additional projected simplified fanning segments that occur due to a flow increase at time instant t_i. Determining which flow rates u_i (and their projected segments) need to be considered to find the lower envelope at time instant t depends on each segment’s time period of influence. The time period of influence is defined as the time period that inflow rate u_i can influence the outflow rate on the downstream link boundary. Let us denote this time period of influence for each upstream segment by time interval $[\vec{t}_i^{\min}, \vec{t}_i^{\max})$. To capture all segments that potentially represent the downstream lower envelope at time t given their periods of influence, we define “regular” segment set $\mathcal{I}(t)$ and fanning segment set $\mathcal{I}_F(t)$ via (5) and (6).

$$\mathcal{I}(t) = \left\{ i = 1, \dots I \mid \vec{t}_i^{\min} \le t < \vec{t}_i^{\max} \right\}, \tag{5}$$

$$\mathcal{I}_F(t) = \left\{ i = 1, \ldots I \mid \vec{t}_{i-1}^{\max} \le t < \vec{t}_i^{\min} \right\}. \tag{6}$$

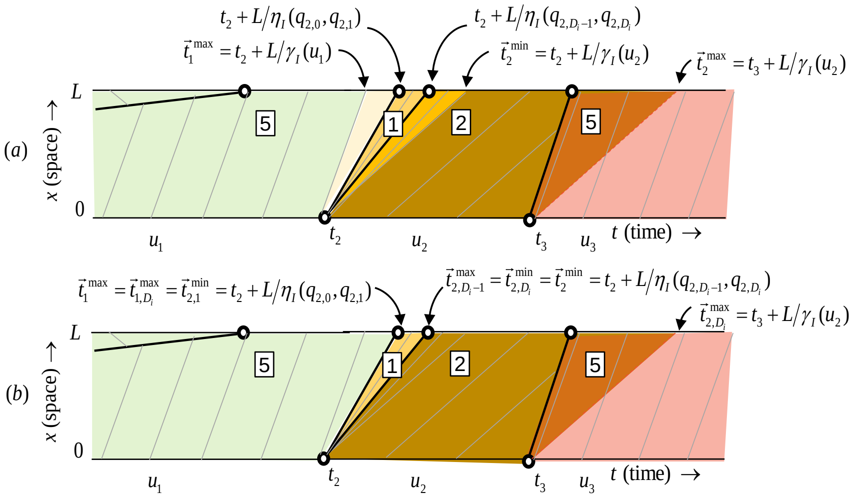
From Part I we know that for regular segments $\vec{t}_i^{\min} = t_i + L/\gamma_I(u_i)$ and $\vec{t}_i^{\max} = t_{i+1} + L/\gamma_I(u_i)$, yielding the projection of regular segments $\vec{U}_i(t)$ via

$$\vec{U}_i(t) = \begin{cases} U(t_i) + u_i\left((t - t_i) - \dfrac{L}{\gamma_I(u_i)} \right) + \xi_I(u_i), & \text{if } \vec{t}_i^{\min} \le t \le \vec{t}_i^{\max}, \\ \infty, & \text{otherwise.} \end{cases} \tag{7}$$

For fanning segments, period $[\vec{t}_{i-1}^{\max}, \vec{t}_i^{\min})$ fills the gap between two consecutive regular segments in case of a flow increase, covering the time period of the expansion fan. The projected piecewise linear fanning segments $\vec{F}_i(t \mid \cdot)$ on the downstream link boundary are given in (8), as per Part I. Note that an inner linearization scheme is adopted, where fanning is split in multiple steps d, with $d = 0, \ldots, D_i$. Each step has its own (internal) time period of influence. Further, it holds by definition that $q_{i,0} = u_{i-1}$ and $q_{i,D_i} = u_i$.

$$\vec{F}_i(t \mid q_{i,0}, \ldots, q_{i,D_i}) = \begin{cases} U(t_i) + q_{i,0}\left((t - t_i) - \dfrac{L}{\gamma_I(q_{i,0})} \right) + \xi_I(q_{i,0}), & \text{if } \vec{t}_{i-1}^{\max} \le t \le t_i + \dfrac{L}{\eta_I(q_{i,0}, q_{i,1})}, \\ \vdots & \\ U(t_i) + q_{id}\left((t - t_i) - \dfrac{L}{\gamma_I(q_{i,d})} \right) + \xi_I(q_{i,d}), & \text{if } t_i + \dfrac{L}{\eta_I(q_{i,d-1}, q_{id})} \le t \le t_i + \dfrac{L}{\eta_I(q_{i,d}, q_{i,d+1})}, \\ \vdots & \\ U(t_i) + q_{i,D_i}\left((t - t_i) - \dfrac{L}{\gamma_I(q_{i,D_i})} \right) + \xi_I(q_{i,D_i}), & \text{if } t_i + \dfrac{L}{\eta_I(q_{i,D_i-1}, q_{i,D_i})} \le t \le \vec{t}_i^{\min}, \\ \infty, & \text{otherwise.} \end{cases} \tag{8}$$

A graphical interpretation of some example periods of influence, based on (7) and (8), are depicted in Figure 4(a), noting that these periods can overlap.



$t_2 + L/\eta_I(q_{2,0}, q_{2,1})$
$t_2 + L/\eta_I(q_{2,D_i-1}, q_{2,D_i})$
$\vec{t}_1^{\max} = t_2 + L/\gamma_I(u_1)$
$\vec{t}_2^{\min} = t_2 + L/\gamma_I(u_2)$
$\vec{t}_2^{\max} = t_3 + L/\gamma_I(u_2)$
L
5
1
2
5
(a)
x (space) →
0
u_1
t_2
u_2
t_3
u_3
t (time) →
$\vec{t}_1^{\max} = \vec{t}_{1,D_i}^{\max} = \vec{t}_{2,1}^{\min} = t_2 + L/\eta_I(q_{2,0}, q_{2,1})$
$\vec{t}_{2,D_i-1}^{\max} = \vec{t}_{2,D_i}^{\min} = \vec{t}_2^{\min} = t_2 + L/\eta_I(q_{2,D_i-1}, q_{2,D_i})$
$\vec{t}_{2,D_i}^{\max} = t_3 + L/\gamma_I(u_2)$
L
5
1
2
5
(b)
x (space) →
0
u_1
t_2
u_2
t_3
u_3
t (time) →

Figure 4: (a) Space-time diagram of (colour coded) periods of influence for regular and fanning segments, (b) simplified and generalised (colour coded) periods of influence for segments of $\vec{U}_{id}(t)$.

By restricting ourselves to inner linearization, we can observe from (8) that the flow rates during the first and last fanning segments, namely $q_{i,0} = u_{i-1}$ and $q_{i,D_i} = u_i$, are the same as the flow rates as in the regular projected segments as formulated in (7). This provides the opportunity to reformulate (4) such that merge fanning segments $\vec{F}_i(t)$ are merged with regular projected segments $\vec{U}_i(t)$. The only difference between (7) and (8) lies in the definitions of each segment's period of influence. Therefore, we redefine the start and end point of the period of influence from $\vec{t}_i^{\min}, \vec{t}_i^{\max}$ to the more general $\vec{t}_{i,d}^{\min}$ and $\vec{t}_{i,d}^{\max}$ defined in (9) and (10), respectively.

$$\vec{t}_{i,d}^{\min} = \begin{cases} t_i + \dfrac{L}{\eta_I(u_{i,d-1}, u_{i,d})}, & \text{if } u_i > u_{i-1} \text{ and } d > 1, \\ t_i + \dfrac{L}{\eta_I(u_{i-1,D_{i-1}-1}, u_{i,d})}, & \text{if } u_i > u_{i-1} \text{ and } d = 1, \\ t_i + \dfrac{L}{\gamma_I(u_{i,d})}, & \text{otherwise.} \end{cases} \tag{9}$$

$$\vec{t}_{i,d}^{\max} = \begin{cases} t_i + \dfrac{L}{\eta_I(u_{i,d}, u_{i,d+1})}, & \text{if } u_{i+1} > u_i \text{ and } d < D_i - 1, \\ t_{i+1} + \dfrac{L}{\eta_I(u_i, u_{i+1,1})}, & \text{if } u_{i+1} > u_i \text{ and } d = D_i - 1, \\ t_{i+1} + \dfrac{L}{\gamma_I(u_{i,d})}, & \text{otherwise,} \end{cases} \tag{10}$$

One interprets (9) and (10) as follows: First, the last case of both (9)-(10) is activated only in the absence of fanning, replacing original definitions $\vec{t}_i^{\min}, \vec{t}_i^{\max}$ for "regular" segments. Second, we look at the middle case of (9)-(10). Observe from (8) that the first projected fanning segment extends the previous flow rate's regular period of influence $\vec{t}_{i-1}^{\max}$ while the last projected fanning segment brings forward the current flow rate's regular period of influence $\vec{t}_i^{\min}$. The middle case in (9-10) identifies these two outer segments of $\vec{F}_i(t\,|\,\cdot)$ and extends the related regular periods of influence by adopting the shockwave rather than the characteristic wave. This removes the need to model these fanning steps as separate segments because they are "glued" to the preceding and current "regular" segments. Thirdly, we examine the first case of (9)-(10). This covers the "internal" fanning segments of $\vec{F}_i(t)$, "internal" refers to the fact that these segments reside in between other fanning segments. The only difference with the middle case is that the preceding flow rate is the previous fanning step of the same inflow rate change, rather than an earlier inflow rate change.

Utilising (9)-(10), all segments originally contained in either $\vec{F}_i(t)$ or $\vec{U}_i(t)$ are replaced by the more concise $\vec{U}_{i,d}(t)$:

$$\vec{U}_{i,d}(t) = \begin{cases} U(t_i) + u_{i,d}\left((t - t_i) - \dfrac{L}{\gamma_I(u_{i,d})}\right) + \xi_I(u_{i,d}), & \text{if } \vec{t}_{i,d}^{\min} \le t \le \vec{t}_{i,d}^{\max}, \\ \infty, & \text{otherwise.} \end{cases} \tag{11}$$

Inflow rate u_i in (7) is replaced with $u_{i,d}$, which is the d^{th} fanning step if $u_i > u_{i-1}$, and otherwise is defined to be u_i in case of a flow decrease (in which no fanning segments are generated). Observe that (11) results in less separate periods of influence than the original (7)-(8) due to the merging of adjacent segments with equal flow rates. This can also be seen in Figure 4(b) which shows the same situation as Figure 4(a), but now based on (11). The original lower envelope problem formulation of (4) simplifies to

$$\vec{U}(t) = \min_{(i,d)\in\mathcal{I}(t)} \left\{\vec{U}_{i,d}(t)\right\}, \tag{12}$$

where the original definition of $\mathcal{I}(t)$ is replaced with (13) denoting the set of flow rate, fanning step pairings (i,d),

$$\mathcal{J}(t)=\left\{(i,d)\,|\,\vec{t}_{i,d}^{\,\min}\le t<\vec{t}_{i,d}^{\,\max};\ \ i\in\{1,\ldots I\};\ \ d\in\{1,\ldots,D_i-1\}\right\}. \tag{13}$$

We call a segment (i,d) and the corresponding flow rate dominant at a certain time instant *t* if it is part of the lower envelope.

We refer to Appendix A for the hypercritical formulations, which are obtained in a similar fashion. The lower envelope problem formulation of Equation (12) serves as the foundation to derive the implicit event-based solution scheme. The event-based formulation will no longer be formulated in terms of segments, but only considers the points (and flow rates) at which a segment first coincides with the lower envelope (i.e., at which the flow rate changes). We postpone this discussion to Section 0 in order to first complete the link and node model formulations. We continue with the introduction of the mixture propagation model.

3.1.2 Mixture propagation

Like flow rates, mixtures are tracked at the link boundaries. The content of a mixture $m\in\mathcal{M}$ depends on the chosen modelling strategy (path based, destination based, departure time period based, etc.), but this choice has no impact on how this information propagates on the link level. Hence, mixture propagation is agnostic to the type of information it carries. The portion of flow belonging to an upstream, downstream mixture m on a link at time t is denoted by $\underline{\rho}_m(t)$ and $\bar{\rho}_m(t)$, respectively, for which holds that

$$\sum_{m\in\mathcal{M}}\bar{\rho}_m(t)=1,\quad \text{and}\quad \sum_{m\in\mathcal{M}}\underline{\rho}_m(t)=1. \tag{14}$$

The conservation of mixture information within a link is preserved by equating mixture propagation to (cumulative) vehicle propagation. For example, consider a vehicle following a path which makes a right turn at the end of a link. When the vehicle exits the link, it should still make the intended right turn, i.e. this information is attached to this vehicle and therefore mixture information is too. According to kinematic wave theory, vehicles propagate differently compared to flow rates and therefore a separate formulation for propagating mixture information is required. This is exemplified by Figure 5; it shows some flow mixture trajectories based on the FD introduced in Figure 2(a). Consider for example $\underline{\rho}_m(t_0)=\bar{\rho}_m(t_6)$, and $\underline{\rho}_m(t_2)=\bar{\rho}_m(t_7)$, because $V(t_6)=U(t_0)$ and $V(t_7)=U(t_2)$, which relate to travel time of the vehicles carrying the mixture information (without explicitly tracking the trajectories).

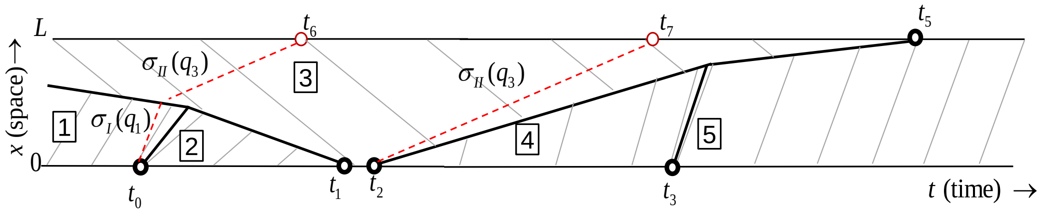


Figure 5: Space time diagram with some example mixture/vehicle trajectories, characteristic waves and shockwaves for a given cumulative flow pattern.

Knowing that the flow propagation model satisfies perfect first-in-first-out (FIFO), this property is used to relate downstream mixture $\bar{\rho}_m(t)$ to its upstream counterpart $\underline{\rho}_m(t)$ as follows:

$$\bar{\rho}_m(t+\tau(t))=\underline{\rho}_m(t),\quad \text{where}\quad \tau(t)\ \text{ solves }\ V(t+\tau(t))=U(t), \tag{15}$$

with $\tau(t)$ representing the link travel time of a vehicle with number $U(t)$, indicating the vehicle entered the upstream link boundary link at time t.

This completes the link model formulation. The event based formulation of (15) is derived in Section 4.2. We proceed by discussing the interface between link model and node model, making the model suitable for general networks. Sending and receiving flows are introduced, which transfer information between the link boundaries and the node model.

3.1.3 Connecting link and node models: sending and receiving flows

During simulation, each link boundary is in one of two possible flow states: (i) Hypocritical, or (ii) hypercritical. The combination of the two link boundary states defines the link state, see Table 1.

Table 1: Relation between link boundary state and link state

	Upstream boundary	***Downstream boundary***	***Link state***	***Excess sending flow***	***Excess receiving flow***
I	Hypocritical	Hypocritical	Free-flow	$S^{+}=0$	$R^{+}>0$
II	Hypocritical	Hypercritical	Congestion	$S^{+}>0$	$R^{+}>0$
III	Hypercritical	Hypercritical	Spillback	$S^{+}>0$	$R^{+}=0$
IV	Hypercritical	Hypocritical	Not possible	-	-

Sending flow $s(t)$ (veh/h) is the flow rate at which vehicles attempt to leave the link at the downstream link boundary. Whenever there is no excess demand, there is no excess sending flow, i.e. $S^{+}(t)=0.$ In this case, outflow rate $v(t)$ is dictated by some earlier upstream flow state $u_{i',d'}$, with $(i',d')\in\mathcal{I}(t).$ This means the link is in a free-flow state which follows from subtracting the cumulative outflow from the potential cumulative outflow, as defined in (16). We point out that free-flow does not necessarily mean driving at maximum speed (when adopting a non-linear hypocritical branch in the FD). On the other hand, if $S^{+}(t)>0,$ congestion is present, with $S^{+}(t)$ (veh) being the number of vehicles unsuccessful in leaving the link at time t (i.e., the number of vehicles in the queue),

$$S^{+}(t)=\min_{(i,d)\in\mathcal{I}(t)}\left\{\vec{U}_{id}(t)\right\}-V\left(t\right), \tag{16}$$

with sending flow rate $s(t)$ conditional on $S^{+}(t)$:

$$s(t)=\begin{cases} u_{i',d'}, & \text{if } S^{+}(t)=0,\\ q^{\max}, & \text{otherwise,}\end{cases} \quad \text{such that } (i',d')=\operatorname*{argmin}_{(i,d)\in\mathcal{I}(t)}\left\{\vec{U}_{i,d}(t)\right\}. \tag{17}$$

Observe that whenever there is a queue, the sending flow rate is set to capacity. Analogous to Equations (4)-(13), excess receiving capacity (supply), denoted by $R^{+}(t)$ (veh), is given by

$$R^{+}(t)=\min_{(j,d)\in\mathcal{J}(t)}\left\{\overleftarrow{V}_{jd}(t)\right\}-U\left(t\right), \tag{18}$$

with possible receiving flow

$$r(t)=\begin{cases} v_{j',d'}, & \text{if } R^{+}(t)=0,\\ q^{\max}, & \text{otherwise,}\end{cases} \quad \text{such that } (j',d')=\operatorname*{argmin}_{(j,d)\in\mathcal{J}(t)}\left\{\overleftarrow{V}_{j,d}(t)\right\}. \tag{19}$$

Here, possible receiving flow $r(t)$ (veh/h) is set to capacity if there is no spillback. In case of spillback the receiving flow is dictated by earlier downstream flow rate $v_{j',d'}$, with $(j',d')\in\mathcal{J}(t).$

3.2 Node model

The node model holds two components. A flow distribution model and a mixture distribution model. The latter disentangles flows based on mixture information to yield a continuous-time splitting rate (recall Figure 3). The flow distribution model is tasked with distributing link flows across the nodes based on a set of rules, desired sending flows, possible receiving flows and splitting rates obtained from the mixture distribution model. We first discuss flow distribution, followed by mixture distribution. Since the node model considers multiple incoming and outgoing links, we re-instate link index *a* in this section.

3.2.1 Flow distribution

Tampère et al. (2011) introduced a set of requirements to which a proper first order macroscopic node model should adhere. Since they only consider single-commodity flow, the term node model is used interchangeably with flow distribution model. Smits et al. (2015) identified a family of node models that satisfy these requirements, which include models proposed by Gibb (2011) and Flötteröd and Rhode (2011). Current LTM implementations and derivatives like Himpe et al. (2016) and Raadsen et al. (2016) adopt the node model of Tampère et al. (2011), but could adopt any other node model that satisfies the requirements.

Given node $n \in \mathcal{N},$ the flow distribution model takes directional sending flow rates $s_{a,b}(t)$ (veh/h), with $a \in \mathcal{A}_n^-, b \in \mathcal{A}_n^+$, and distributes them across its outgoing links, where this distribution is conditional on possible receiving flow rates $r_b(t),\ b \in \mathcal{A}_n^+$. Converting sending flow rates $s_a(t)$ to directional sending flow rates $s_{ab}(t)$ is achieved via

$$s_{a,b}(t) = s_a(t)\varphi_{a,b}(t), \quad a \in \mathcal{A}_n^-, b \in \mathcal{A}_n^+, \tag{20}$$

where $\varphi_{a,b}(t)$ denotes the splitting rate, which holds the flow proportion of link a designated to exit the node via link *b* at time instant *t*. An alternative to splitting rates would be to embed multi-commodity formulations in the flow rate definition itself, combining the flow distribution and mixture distribution model. Observe that this requires an explicit definition of the adopted flow composition method (path based, destination based, departure time period based, etc.). To prevent this, we adopt splitting rates, allowing the flow distribution model to remain agnostic to the underlying mixture distribution method and instead rely on $\varphi_{ab}(t)$ to be provided as input to obtain desired sending turn flows via (20). This then yields:

$$\left[v_{a,b}(t)\right]_{a\in\mathcal{A}_n^-, b\in\mathcal{A}_n^+} = \Gamma_n\left(s_{a,b}(t), r_b(t),\ a \in \mathcal{A}_n^-, b \in \mathcal{A}_n^+\right), \quad n \in \mathcal{N}, \tag{21}$$

where $\Gamma_n(\cdot)$ denotes any first order node model compliant with the requirements described in Tampère et al. (2011). Realised directional outflows $v_{a,b}(t)$ (veh/h) are output of the model. Note that by definition $u_b(t) = \sum_{a\in\mathcal{A}_n^-} v_{a,b}(t),$ and $v_a(t) = \sum_{b\in\mathcal{A}_n^+} v_{a,b}(t).$

3.2.2 Mixture distribution

In this section we incorporate path information within the DNL. In Raadsen et al. (2016) the rather restrictive assumption was made that all path flows remained proportional to each other over different departure time periods throughout the simulation. This effectively only allows a uniformly applied departure profile, but lacks support for (within-iteration) route choice. Here, we relax this assumption by considering periodically fixed splitting rates per route choice period (aligned with each departure time period). This results in as many splitting rates as route choice periods and no longer requires path flow proportionality. Although not fully consistent with the path flows obtained from route choice, it does capture the effects of route choice in a way that, arguably, is most in line with the granularity of the targeted application, namely, large scale DTA. Clearly, there are more ways to embed path information in the model than the departure time based approach proposed here, for example a destination based or fully path based formulation. To illustrate the framework is capable of supporting such alternative formulations as well, we provide a path based mixture distribution formulation in Appendix B (note that this formulation serves as a demonstration, it will not be pursued further for the sake of both brevity and maintaining a clear focus).

The set of available paths p is given by $\mathcal{P},$ where the flow rate on each path at time instant *t* is given by $f_p(t)$ (veh/h). However, since path flow rates are assumed to be stationary per departure time period, $f_p(t)$ simplifies to

$$f_p(t) = f_{p,h}, \quad \text{if } t \in [T_h, T_{h+1}), \text{ where } T_h = \frac{(h-1)T}{H}, \quad \text{for } h = 1, \ldots, H; p \in \mathcal{P}, \tag{22}$$

with simulation end time $T = T_{H+1}$ and $f_{p,h}$ (veh/h) the average path flow rate of path p during period h. Let us now discuss obtaining splitting rates, followed by the propagation of mixtures across the node to yield the upstream mixture portions.

3.2.2.1 *Departure time period based splitting rates*

By modelling departure time period based splitting rates, link level path information is absent and simplifying assumptions are made to approximate this data. Typically, one adopts traditional static assignment to estimate the splitting rates per period, by assuming instantaneous flow propagation and not considering capacity constraints:

$$\varphi_{a,b,h} = \frac{\sum_{p\in P} \delta_{a,p}\delta_{b,p} f_{p,h}}{\sum_{p\in P} \delta_{a,p} f_{p,h}}, \quad h = 1,2,\ldots,H;\, a \in \mathcal{A}_n^-,\, b \in \mathcal{A}_n^+,\, n \in \mathcal{N}, \tag{23}$$

with path incidence indicator $\delta_{a,p}$ taking on a value of 1 whenever the link a is part of path p, and zero otherwise. This then yields continuous time splitting rate $\varphi_{a,b}(t)$ for each node via

$$\varphi_{a,b}(t) = \sum_{h=1}^{H} \bar{\rho}_{a,h}(t)\varphi_{a,b,h}, \quad a \in \mathcal{A}_n^-, b \in \mathcal{A}_n^+, n \in \mathcal{N}, \tag{24}$$

with downstream mixture $\bar{\rho}_{a,h}(t)$ denoting the portion of flow on the downstream link boundary of link a belonging to departure time period h for time instant t. Observe that we replaced $\bar{\rho}_{a,m}(t)$ with $\bar{\rho}_{a,h}(t)$ here. This is a syntactic difference only because each mixture component $m \in \mathcal{M}$ is a departure time period h in this particular mixture distribution formulation. Other model components adopt the non-explicit notation, i.e. $\bar{\rho}_{a,m}(t)$, $m \in \mathcal{M}$ to illustrate they are agnostic as to what mixture method is used. Only within the mixture distribution model this is necessarily made explicit such that it is clear we are dealing with departure time periods (or paths, or destinations, etc.).

3.2.2.2 *Upstream mixture portions*

The splitting rates in (24) serve as input to the flow distribution model discussed in Section 3.2.1, which in turn yield accepted outflow rates. The mixture distribution then takes the accepted flow rates and utilises them to update the upstream mixtures on outgoing links via Equation (26).

$$v_{a,b,h}(t) = \bar{\rho}_{a,h}(t)\varphi_{a,b,h}v_a(t), \tag{25}$$

$$\underline{\rho}_{b,h}(t) = \frac{\sum_{a\in\mathcal{A}_n^-} v_{a,b,h}(t)}{\sum_{a\in\mathcal{A}_n^-} v_{a,b}(t)}, \quad \text{if } b \in \mathcal{A}_n^+, n \in \mathcal{N} \setminus \mathcal{O}. \tag{26}$$

Note that in case of an origin node $n \in \mathcal{O}$, (26) is replaced with a binary choice, where $\underline{\rho}_{bh}(t) = 1$ when $t \in [T_h, T_{h+1})$, and $\underline{\rho}_{b,h}(t) = 0$ otherwise. This is due to the fact that, at the origin, departing flows can only belong to the current period. This section concludes with providing the general definition for link inflow rates: The earlier definition of $u_b(t) = \sum_{a\in\mathcal{A}_n^-} v_{a,b}(t)$ in the flow distribution model did not include origin nodes. We therefore replace this definition with (27) which also ensures the conservation of mixture information across nodes.

$$u_{b,h}(t) = \begin{cases} \underline{\rho}_{b,h}(t)\sum_p f_{p,h}, & \text{if } b \in \mathcal{A}_n^+, n \in \mathcal{O}, \\ \underline{\rho}_{b,h}(t)\sum_{a\in\mathcal{A}_n^-} v_{a,b}(t), & \text{otherwise,} \end{cases} \tag{27}$$

to obtain the more general link inflow rate formulation,

$$u_b(t) = \sum_{h=1}^{H} u_{b,h}(t). \qquad (28)$$

This concludes the continuous time DNL formulation for general networks. Based on the proposed flow and mixture propagation formulation we now introduce the implicit event-based form which underpins the solution scheme presented in Section 0.

4 Implicit event based propagation

We first discuss the implicit event based flow propagation followed by the mixture propagation. Observe that only link model components are rewritten. This is because the node model is time-invariant. As a result, the equations in Section 3.2 can directly be applied in an event-based context.

4.1 Flow events

This event based formulation is grid-free, because there is no fixed time and/or space interval. Of interest to us are only the time instants at which flow rates change on a link boundary. Each such instance is termed an event, written as a combination of a time instant and inflow rate (t_i, u_i) for the i^{th} upstream event, and (t_j, v_j) for the j^{th} downstream event. We will again omit link index a in this section. Events are chronologically ordered such that $i \in \{1,\dots,I_t\}$, $j \in \{1,\dots,J_t\}$, with I_t and J_t denote the last realised event up till time instant t on the upstream or downstream link boundary, respectively. All events result from earlier flow rate changes on the link itself or on adjacent links. Before flow and mixture events are actuated, they first need to pass through the node model. All events actuated by the node model are imposed immediately and are referred to as *"trigger"* events, because they trigger flow rate changes on the link level. Each trigger event may result in a flow rate change some time later on the other end of the link, releasing itself. This second type of event is termed *"release"* event. Trigger and release events can occur on either link boundary since (shock)waves travel both upstream and downstream, depending on the traffic state.

4.1.1 Processing a trigger event: Release event prediction

Let us first consider upstream flow rate change $u_{i,d}$, with $i = I_t$, $d \in \{0,\dots D_i\}$, actuated by the node model, at current time instant t, where $t = t_i$. We utilise Equation (16) to derive the expected time instant at which this inflow rate change affects the downstream outflow rate. This predicted downstream arrival time is denoted by $\vec{t}_{i,d}$. To do this we assume the following: (i) $S^+(\vec{t}_{i,d}) = 0$, (ii) perpetual inflow rate $u_{i,d}$, and (iii) perpetual outflow rate $v_{j,d}$, $j = J_t$. Assumption (i) considers the situation the trigger event's flow rate becomes the sending flow rate in the future (although it may turn out later it does not). Assumptions (ii) and (iii) are simply a result of the fact that we do not yet have information beyond time instant *t* and as such simply assume that current conditions continue indefinitely. This simplifies Equation (16) to

$$S^+(\vec{t}_{i,d}) = \vec{U}_{i,d}(t_i) - V(\vec{t}_{i,d}), \quad \vec{t}_{i,d}^{\,\min} \le \vec{t}_{i,d} \le \vec{t}_{i,d}^{\,\max}. \qquad (29)$$

Writing out (29) based on (11) assuming $S^+(\vec{t}_{i,d}) = 0$ yields

$$U(t_i) + u_{i,d}\left((\vec{t}_{i,d} - t) - \frac{L}{\gamma_I(u_{i,d})} \right) + \xi_I(u_{i,d}) = V(\vec{t}_{i,d}), \quad \vec{t}_{i,d}^{\,\min} \le \vec{t}_{i,d} \le \vec{t}_{i,d}^{\,\max}. \qquad (30)$$

Under the assumptions (ii) and (iii), i.e. constant inflow and outflow rates we utilise (3) to rewrite (30) from its cumulative segment based form to its event based form, with $\vec{t}_{i,d}$ the only unknown, yielding the following prediction equation:

$$\vec{t}_{i,d} = \frac{U(t_i) - V(t_j) - u_{i,d}\left(t_i + \dfrac{L}{\gamma_I(u_{i,d})}\right) + v_j t_j + \xi_I(u_{i,d})}{v_j - u_{i,d}}, \quad \vec{t}^{\min}_{i,d} \le \vec{t}_{i,d} \le \vec{t}^{\max}_{i,d}, \tag{31}$$

with $j = J_{t_i}$ and $i = I_t, d \in \{0,\ldots D_i\}$. At the time of prediction we can immediately verify whether it holds that $\vec{t}^{\min}_{i,d} \le \vec{t}_{i,d}$. In case it this does not hold, assumption (i) is violated, indicating the link is currently in congestion, but does not go to free flow. Hence, the prediction must be discarded and no release event is to be created, see t_0 in Figure 2(b) for an example. We cannot yet verify $\vec{t}_{i,d} \le \vec{t}^{\max}_{i,d}$, because $\vec{t}^{\max}_{id}$ depends on the next, currently unknown, upstream flow rate.

Similarly, given a downstream flow rate change $v_{j,d}$ with $j = J_t, d \in 0,\ldots D_j$, at current time instant $t = t_j$, we obtain the predicted upstream arrival time $\overleftarrow{t}_{j,d}$ of this hypercritical flow state via (32).

$$\overleftarrow{t}_{j,d} = \frac{U(t_i) - V(t_j) - u_i t_j + v_{j,d}\left(t + \dfrac{L}{\gamma_{II}(v_{j,d})}\right) - \xi_{II}(v_{j,d})}{v_{j,d} - u_i}, \quad \overleftarrow{t}^{\min}_{j,d} \le \overleftarrow{t}_{j,d} \le \overleftarrow{t}^{\max}_{j,d}, \tag{32}$$

with $i = I_t$.

4.1.2 *Release event updates*

The previous section discussed the prediction and scheduling of a new release event upon the arrival of a trigger event actuated by the node model. However, it is possible this initial prediction is incorrect due to the arrival of other flow rates on the release event's link boundary. In such a case assumption (ii) is violated and the prediction needs to be updated incorporating this additional link boundary information.

Consider Figure 6(a), an outflow rate change at time t' causes inflow rate $u_{i,d}$ to arrive downstream earlier than initially predicted. To not miss this flow rate change, updates are executed the moment this information becomes available on the opposite link boundary, i.e. at time instant t'. Observe that the difference with the initial prediction lies in the downstream reference time, it has changed from J_{t_i} to $J_{t'}$. The number of events that require updating is typically very limited. Only trigger events (and fanning steps) for which holds that $(i,d) \in \mathcal{J}(t')$ need to be updated, where we recall from Section 3.1.1 that $\mathcal{J}(t')$ denotes events whose period of influence includes t' and therefore might be affected. When the trigger event's updated arrival time exceeds $\vec{t}^{\min}_{i,d}$, it is potentially valid and a release event is scheduled at the updated time instance (if it differs from an already predicted arrival time).

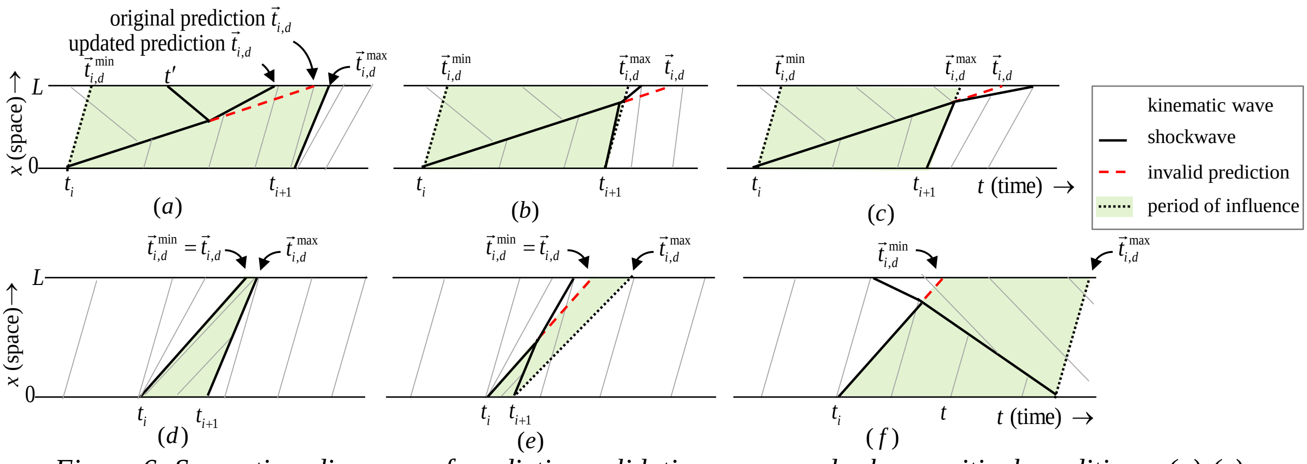


Figure 6: Space-time diagrams of prediction validation cases, under hypercritical conditions; (a)-(c) invalid predictions. Under hypocritical conditions; (a) valid prediction, (b)-(c) invalid prediction.

To keep the algorithm simple and concise, the updating of events does not affect the original prediction. So, multiple release events could be generated for the same trigger event. To determine which release event is the correct one (if any), a separate validation mechanism is used. This validation method is discussed next.

4.1.3 Release event (in)validation

The validation of an earlier predicted release event does one of two things: It either accepts the event (validate), or discards it (invalidate). Let us consider the validation of predicted downstream release event $(\vec{t}_{i,d}, u_{i,d})$. This event is validated if and only if: (i) the predicted arrival time $\vec{t}_{i,d}$ has not changed, i.e. if we would calculate (31) again with the most recent information we would obtain the same prediction, and (ii) it holds that $\vec{t}_{i,d} \leq \vec{t}_{i,d}^{\max}$, i.e. the period of influence reaches to at least current time t. This is simply a matter of checking Equation (10). Observe that in case the upstream flow rate has not changed since the original prediction, $\vec{t}_{i,d} \leq \vec{t}_{i,d}^{\max}$ holds by definition. An example of a valid prediction is shown in Figure 6(d). Whenever a release event is validated, its flow rate is converted to a sending flow rate as in (17), which after the node model update yields accepted outflow rates. Depending on the link states these flow rates may or may not differ from the current flow rates. If they do differ, new trigger events are generated and processed.

Let us now explore the situations a release event is invalidated and discarded. This happens, for instance, in case assumption (iii) no longer holds, i.e. the outflow rate changed, which causes a shift in the cumulative outflow curve compared to original prediction $\vec{t}_{i,d}$. We recall that a potential outflow rate at time instant t is dominant if its projected cumulative segment is minimum, see (12). If this outflow rate change led to an updated arrival time, as discussed in the previous Section, an updated release event has already been scheduled and this (original) prediction should now be discarded (Figure 6(a)). It is also possible the update failed and the flow rate no longer becomes dominant. Schematic examples of this situation are depicted in Figure 6(b), (e) and (f). Either way, the originally predicted arrival time has changed and the release event associated with that arrival time is invalidated and discarded.

Another possibility that can lead to invalidating a release event is the violation of prediction assumption (ii), i.e. one or more inflow rate changes have happened. If one of the inflow rate changes in time period dominates the current release event's projected cumulative curve (that is, if the lower envelope changes), this other inflow rate must already have been actuated on the downstream link boundary. This situation has just been discussed, and means that the event is invalidated and discarded, see Figure 6(b) and (e) for specific examples. This leaves the situation depicted in Figure 6(c), namely an inflow rate change ending the period of influence $\vec{t}_{i,d}^{\max}$ of $u_{i,d}$ prematurely. Observe that here, (31) with $j = J_t$ still equals original prediction $\vec{t}_{i,d}$, but constraint $\vec{t}_{i,d} < \vec{t}_{i,d}^{\max}$ is violated. In other words, the flow rate‘s period of influence does not "reach" the point it can dominate the lower envelope and therefore never yields a solution to (12). Although space-time diagrams do not show lower envelopes, Figure 6(f) clearly shows $\vec{t}_{i,d} > \vec{t}_{i,d}^{\max}$, which reflects the inability of this flow rate to dominate the lower envelope. This again means that the event is invalidated and discarded.

Prediction, updates, and validation for a downstream flow rate change have not been discussed. However, the exact same mechanism and reasoning applies compared to their upstream counterpart. For the sake of brevity, we choose to not repeat ourselves by including this discussion. This completes the flow propagation model. We now proceed to introduce the event-based mixture propagation.

4.2 Mixture events

From (15) we see that mixture propagation depends solely on cumulative segment information. Since cumulative segments are piecewise linear (Equation (3)), our mixtures, like flow rates, become piecewise stationary. Again, like flow rates, we term a node model imposed mixture change a trigger event and the arrival of such an earlier predicted mixture on the opposite link boundary a release event. We introduce $\underline{t}_y, \overline{t}_y$ for the time instance the y^{th} mixture event is imposed on the upstream, downstream link boundary respectively. A single mixture component $m \in \mathcal{M}$ within an upstream, downstream mixture event is denoted via $\rho_{m,y}$ respectively. Note that each mixture event carries all mixture components $m \in \mathcal{M}$ such that an upstream mixture event is represented by the pair $(\underline{t}_y, \boldsymbol{\rho}_y)$ where $\underline{\boldsymbol{\rho}}_y = (\underline{\rho}_{1,y}, \underline{\rho}_{2,y}, \ldots, \underline{\rho}_{\mathcal{M},y})$. Its downstream counterpart is denoted similarly via $(\overline{t}_y, \boldsymbol{\rho}_y)$. Events are ordered such that $y \in \{1, \ldots, \underline{Y}_t\}$, with $\underline{Y}_t$ denoting the last realised mixture event up till time instant t on the upstream link boundary respectively. Note that on the downstream link boundary not all mixture might have arrived yet, therefore $\overline{Y}_t$ denotes the last realised mixture event up till time instant t on the downstream link boundary with $1 \leq \overline{Y}_t \leq \underline{Y}_t$. this then yields

$$\begin{aligned} \underline{\rho}_m(t) &= \rho_{m,y}, \quad t_y \leq t < t_{y+1},\ y \in \{1, \ldots, \underline{Y}_t\}, \\ \overline{\rho}_m(t) &= \rho_{m,y}, \quad t_y \leq t < t_{y+1},\ y \in \{1, \ldots, \overline{Y}_t\}. \end{aligned} \tag{33}$$

4.2.1 Processing a trigger event: Release event prediction

Like flow event triggers, mixture event triggers are imposed by the node model. Contrary to flow rates, mixture information only propagates forward and therefore triggers are only generated on the upstream link boundary and released downstream. Let us consider a change in upstream mixture $\underline{\rho}_{m,y}$, at time $\underline{t}_y = t$, where $y = \underline{Y}_t$. We want to predict when this mixture event is expected to arrive downstream knowing $\vec{t}_y = \underline{t}_y + \tau(\underline{t}_y)$. Utilising (3) we rewrite condition $V(\vec{t}_y) = U(\underline{t}_y)$ in (15) to

$$V(t_j) + v_j(\vec{t}_y - t_j) = U(t_i) + u_{i,d}(\underline{t}_y - t_i), \quad t_i \leq \underline{t}_y < t_{i+1};\ t_j \leq \vec{t}_y < t_{j+1}. \tag{34}$$

Note that $d = D_i$ if $\underline{t}_y > t_i$. Only when the mixture and flow rate change coincide and an expansion fan exist, i.e. $u_i > u_{i-1}$, then $\underline{t}_y = t_i$ and $d = 1$, because the lead vehicle within the fan already carries the new mixture information. In addition it holds that $j = J_t$. To rewrite (34) to its event-based form predicting $\vec{t}_y$, we again assume constant inflow and outflow rates beyond time t, yielding prediction equation

$$\vec{t}_y = \frac{U(t_i) - V(t_j) + u_{i,d}(\underline{t}_y - t_i) + v_j t_i}{v_j}, \quad \text{with } t_i \leq \underline{t}_y < t_{i+1}, \text{and } t_j \leq \vec{t}_y < t_{j+1}. \tag{35}$$

Observe that (35) does nothing more than extrapolate the current outflow rate, until the difference between up and downstream cumulatives vanishes. Notice that a valid prediction can always be made unless the outflow rate is zero, if so we set $\vec{t}_y = \infty$.

4.2.2 Release event validation and corrected predictions

As discussed earlier for flow rates, changes in outflow rate in period $[\underline{t}_y, \vec{t}_y)$ can cause an initial mixture prediction $\vec{t}_y$ to be discarded, even though the mixture still arrives downstream at some other time. Making sure this true arrival time is captured requires additional predictions that update the arrival time, similar to the updates of flow event predictions as discussed in Section 4.1.3. Whenever the outflow rate changes at some time instant t', all upstream mixture events that have not yet arrived downstream require updating. Or more formally, all mixture trigger events $y' \in \{\overline{Y}_{t'}, \ldots, \underline{Y}_{t'}\}$ require updating via (35). The newly predicted arrival time is then attached to a new release event.

Like flow events, this can lead to multiple predicted mixture release events for the same upstream mixture trigger event. It is then up to the mixture validation mechanism to evaluate each release event upon arrival, discard the invalid events and validate only the one true release event.

Validation of mixture events works in a similar way to the validation of flow events. From (35) we know that $\vec{t}_y < t_{j+1}$ cannot yet be verified at the time of prediction, because $j = J_t$, and the time of the next event $J_t + 1$ is yet unknown. Therefore, we wait until the simulation reaches time $\vec{t}_y$ to validate the release event. If at that time it holds that $J_{\underline{t}_y} = J_{\vec{t}_y}$, no outflow rate changes occurred since it was predicted and so the mixture is validated, otherwise the event is discarded. When the mixture is validated a new downstream mixture release event is created which in turn triggers the node model for an update. This completes the event-based formulations.

5 Solution scheme

The presented algorithm is split into three parts: First, an initialization step to setup variables. Second, the route choice to DNL interface component, responsible for triggering inflow and mixture updates at the origins as a result of altered path flows. Third, the main algorithm itself, see Figure 7 for an overview.

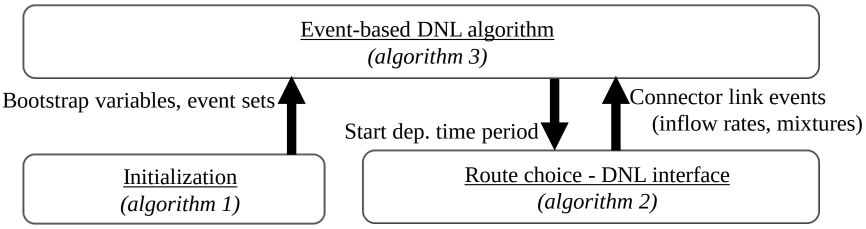


Figure 7: Algorithm components and schematic interaction.

5.1 Initialization

In this step all links are initialised: Inflow rate, outflow rate, upstream cumulatives and downstream cumulatives are all set to zero. Further, future scheduled flow events are tracked in newly introduced sets $\vec{\mathcal{E}}_a^f = \{(t,q)\}$ and $\overleftarrow{\mathcal{E}}_a^f = \{(t,q)\}$, which are sorted by time in ascending order. Scheduled mixture release events are tracked in $\vec{\mathcal{E}}_a^m = \{(t,\boldsymbol{\rho})\}$, also sorted by time in ascending order. Algorithm 1 shows the initialisation step as executed on each link (subscript *a* is omitted). Note that initialising the start time to $-\infty$ and initial flow rates and cumulative to zero ensures that we start with an empty network. Mixtures do not need to be initialised.

Algorithm1: Initialization.

```
$I = J = Y := 0.$
$t_0 := -\infty,$
$U_0 = V_0 := 0,$
$u_0 = v_0 := 0;$
$\vec{\mathcal{E}}^f = \overleftarrow{\mathcal{E}}^f = \vec{\mathcal{E}}^m := \varnothing,$
                                        Initialization
```

5.2 Route choice DNL interface

Path flows are dependent on the departure time periods. Even when they are assumed given, they need to be imposed onto the network during simulation. The route choice to DNL interface is responsible for this task. It updates the connector link flows and/or mixtures whenever a new departure time period starts as shown in Algorithm 2.

Algorithm2: Route choice - DNL interface component.

```
if new departure time period starts, i.e. $t = T_h, \quad h \in 1, \ldots, H$
  for each origin connector link $a \in \mathcal{A}_n^+, n \in \mathcal{O}$
    if the inflow changes, i.e. $\sum_{m \in \mathcal{M}} u_{a,m}(t) = u_{a,I_t}$                    (Equation 27)
      new flow trigger via $I_t := I_t + 1$, such that $u_{a,I_t} := \sum_{m \in \mathcal{M}} u_{a,m}(t)$.
      schedule release(s) $\vec{\mathcal{E}}_a^{\,f} := \vec{\mathcal{E}}_a^{\,f} \bigcup (\vec{t}_{I_t,d}, u_{I_t,d}), \quad d \in 0, \ldots, D_{I_t}$, if $\vec{t}_{I_t,d} \geq \vec{t}_{I_t,d}^{\min}$.    (Equation 31)
    end if
    if the mixture changes, i.e. $\exists \underline{\rho}_{a,m}(t) \neq \underline{\rho}_{a,m,\underline{Y}_t}, \quad m \in \mathcal{M}$.                    (Equation 26)
      new mixture trigger via $\underline{Y}_t := \underline{Y}_t + 1$, such that $\underline{\rho}_{a,m,\underline{Y}_t} := \underline{\rho}_{a,m}(t),\ m \in \mathcal{M}$.
      schedule release $\vec{\mathcal{E}}_a^{\,m} := \vec{\mathcal{E}}_a^{\,m} \bigcup (\vec{t}_{\underline{Y}_t}, \boldsymbol{\rho}_{\underline{Y}_t})$.                    (Equation 35)
    end if
  end for
end if
```

Route choice - DNL interface

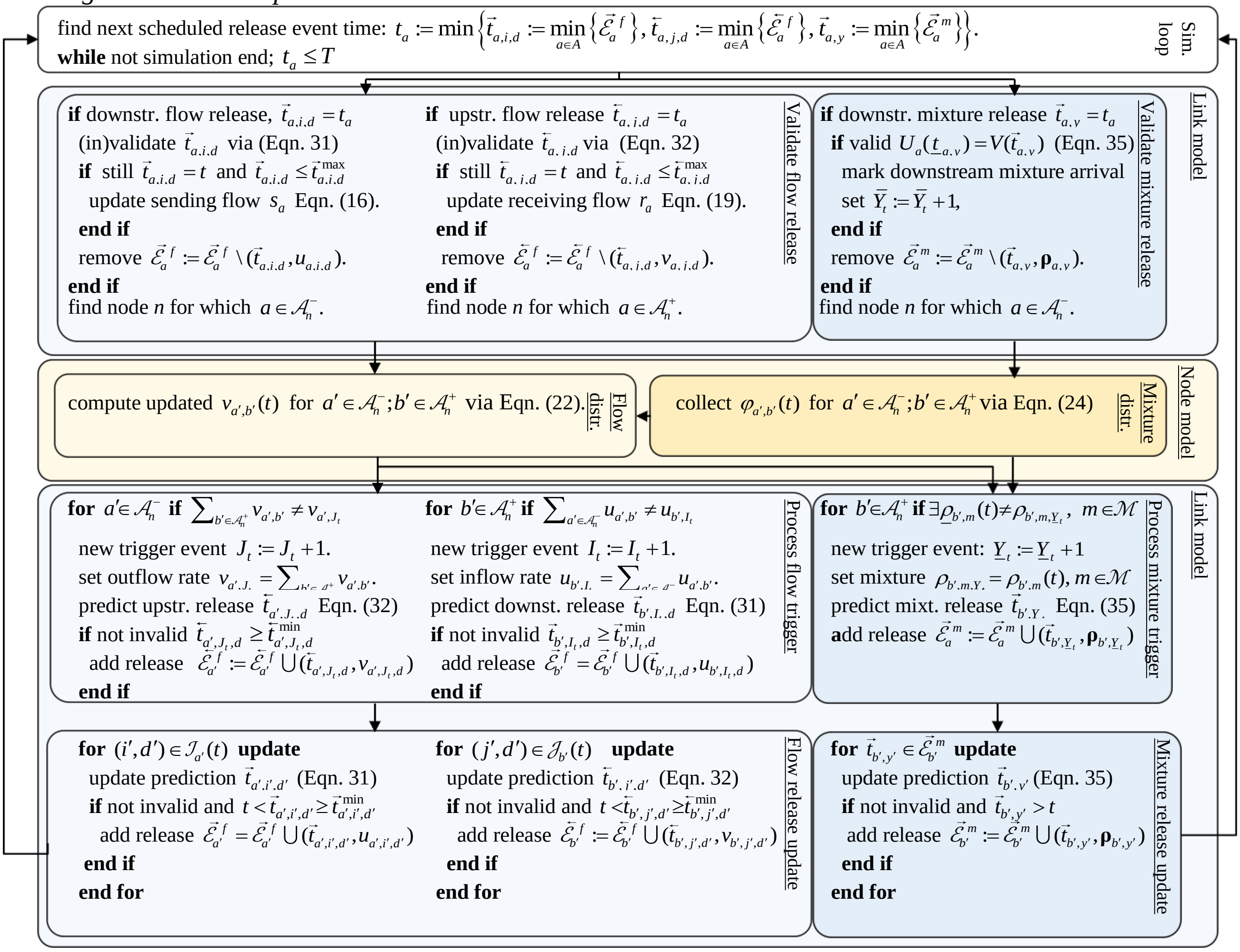
5.3 DNL Simulation

The main procedure (Algorithm 3) entails a recursive process that takes the earliest future scheduled release event in the entire network and attempts to validate it. This event might be a flow rate change or a mixture change. If the event is validated, it triggers the node model that based on sending flows, receiving flows, and splitting rates, computes newly accepted turn flow rates $v_{a,b}(t)$. Based on these turn flow rates; accepted link inflow rates, link outflow rates, as well as updated upstream mixtures are computed. Whenever a link inflow rate, link outflow rate or mixture percentage differs from its last known value, a trigger event carrying the updated flow/mixture information is generated. For each new trigger event, the related link predicts when the flow rate or mixture is expected to be released on the opposite link boundary. All potentially valid predictions are then scheduled as release events for future processing. Whenever a flow rate changes on a link boundary (either via a trigger or a release event) it provides additional information on the link boundary and potentially impacts the arrival time of other flow or mixture events. Consequently, events are updated if needed in such cases (recall Sections 4.1.3. and 4.2.2). This procedure repeats itself until the simulation end time is reached.

The main reason why this algorithm is efficient is twofold. First, it hardly requires any lookups or searches which are known to be costly. Secondly, the equations used in the algorithm only contain basic operations. So even when the simulation, on a large network, might generate millions of events, the total simulation time is comparatively low compared to other DNL algorithms, see also Section 6.

We can see from Algorithm 3 that whenever the sending or receiving flow rate changes (validate flow release in the link model), an update of the node model is triggered. Similarly, whenever the splitting rate, i.e. flow mixture, changes an update of the node model is also required. Consequently, the fewer changes in either flow rates or splitting rates are produced, the fewer the number of events and hence the lower the resulting computational burden. This observation leads us to the discussion of the approximate solution scheme.

Algorithm 3: Main procedure.



```
find next scheduled release event time: $t_a := \min\left\{\vec{t}_{a,i,d} := \min_{a\in A}\left\{\vec{\mathcal{E}}_a^{\,f}\right\}, \overleftarrow{t}_{a,j,d} := \min_{a\in A}\left\{\overleftarrow{\mathcal{E}}_a^{\,f}\right\}, \vec{t}_{a,y} := \min_{a\in A}\left\{\vec{\mathcal{E}}_a^{\,m}\right\}\right\}$.
while not simulation end; $t_a \le T$
```

```
if downstr. flow release, $\vec{t}_{a,i,d} = t_a$
  (in)validate $\vec{t}_{a,i,d}$ via (Eqn. 31)
  if still $\vec{t}_{a,i,d} = t$ and $\vec{t}_{a,i,d} \le \vec{t}_{a,i,d}^{\max}$
    update sending flow $s_a$ Eqn. (16).
  end if
  remove $\vec{\mathcal{E}}_a^{\,f} := \vec{\mathcal{E}}_a^{\,f} \setminus (\vec{t}_{a,i,d}, u_{a,i,d})$.
end if
find node $n$ for which $a \in \mathcal{A}_n^-$.
```

```
if upstr. flow release $\overleftarrow{t}_{a,j,d} = t_a$
  (in)validate $\overleftarrow{t}_{a,j,d}$ via (Eqn. 32)
  if still $\overleftarrow{t}_{a,j,d} = t$ and $\overleftarrow{t}_{a,j,d} \le \overleftarrow{t}_{a,j,d}^{\max}$
    update receiving flow $r_a$ Eqn. (19).
  end if
  remove $\overleftarrow{\mathcal{E}}_a^{\,f} := \overleftarrow{\mathcal{E}}_a^{\,f} \setminus (\overleftarrow{t}_{a,j,d}, v_{a,j,d})$.
end if
find node $n$ for which $a \in \mathcal{A}_n^+$.
```

```
if downstr. mixture release $\vec{t}_{a,y} = t_a$
  if valid $U_a(\underline{t}_{a,y}) = V(\vec{t}_{a,y})$ (Eqn. 35)
    mark downstream mixture arrival
    set $\bar{Y}_t := \bar{Y}_t + 1$,
  end if
  remove $\vec{\mathcal{E}}_a^{\,m} := \vec{\mathcal{E}}_a^{\,m} \setminus (\vec{t}_{a,y}, \boldsymbol{\rho}_{a,y})$.
end if
find node $n$ for which $a \in \mathcal{A}_n^-$.
```

```
compute updated $v_{a',b'}(t)$ for $a' \in \mathcal{A}_n^-; b' \in \mathcal{A}_n^+$ via Eqn. (22).
```

```
collect $\varphi_{a',b'}(t)$ for $a' \in \mathcal{A}_n^-; b' \in \mathcal{A}_n^+$ via Eqn. (24)
```

```
for $a' \in \mathcal{A}_n^-$ if $\sum_{b'\in\mathcal{A}_n^+} v_{a',b'} \ne v_{a',J_t}$
  new trigger event $J_t := J_t + 1$.
  set outflow rate $v_{a',J_t} = \sum_{b'\in\mathcal{A}_n^+} v_{a',b'}$.
  predict upstr. release $\overleftarrow{t}_{a',J_t,d}$ Eqn. (32)
  if not invalid $\overleftarrow{t}_{a',J_t,d} \ge \overleftarrow{t}_{a',J_t,d}^{\min}$
    add release $\overleftarrow{\mathcal{E}}_{a'}^{\,f} := \overleftarrow{\mathcal{E}}_{a'}^{\,f} \cup (\overleftarrow{t}_{a',J_t,d}, v_{a',J_t,d})$
  end if
```

```
for $b' \in \mathcal{A}_n^+$ if $\sum_{a'\in\mathcal{A}_n^-} u_{a',b'} \ne u_{b',I_t}$
  new trigger event $I_t := I_t + 1$.
  set inflow rate $u_{b',I_t} = \sum_{a'\in\mathcal{A}_n^-} u_{a',b'}$.
  predict downst. release $\vec{t}_{b',I_t,d}$ Eqn. (31)
  if not invalid $\vec{t}_{b',I_t,d} \ge \vec{t}_{b',I_t,d}^{\min}$
    add release $\vec{\mathcal{E}}_{b'}^{\,f} = \vec{\mathcal{E}}_{b'}^{\,f} \cup (\vec{t}_{b',I_t,d}, u_{b',I_t,d})$
  end if
```

```
for $b' \in \mathcal{A}_n^+$ if $\exists \underline{\rho}_{b',m}(t) \ne \rho_{b',m,\underline{Y}_t}$, $m \in \mathcal{M}$
  new trigger event: $\underline{Y}_t := \underline{Y}_t + 1$
  set mixture $\rho_{b',m,Y_t} = \rho_{b',m}(t), m \in \mathcal{M}$
  predict mixt. release $\vec{t}_{b',Y_t}$ Eqn. (35)
  add release $\vec{\mathcal{E}}_a^{\,m} := \vec{\mathcal{E}}_a^{\,m} \cup (\vec{t}_{b',\underline{Y}_t}, \boldsymbol{\rho}_{b',\underline{Y}_t})$
```

```
for $(i',d') \in \mathcal{I}_{a'}(t)$ update
  update prediction $\vec{t}_{a',i',d'}$ (Eqn. 31)
  if not invalid and $t < \vec{t}_{a',i',d'} \ge \vec{t}_{a',i',d'}^{\min}$
    add release $\vec{\mathcal{E}}_{a'}^{\,f} = \vec{\mathcal{E}}_{a'}^{\,f} \cup (\vec{t}_{a',i',d'}, u_{a',i',d'})$
  end if
end for
```

```
for $(j',d') \in \mathcal{J}_{b'}(t)$ update
  update prediction $\overleftarrow{t}_{b',j',d'}$ (Eqn. 32)
  if not invalid and $t < \overleftarrow{t}_{b',j',d'} \ge \overleftarrow{t}_{b',j',d'}^{\min}$
    add release $\overleftarrow{\mathcal{E}}_{b'}^{\,f} := \overleftarrow{\mathcal{E}}_{b'}^{\,f} \cup (\overleftarrow{t}_{b',j',d'}, v_{b',j',d'})$
  end if
end for
```

```
for $\vec{t}_{b',y'} \in \vec{\mathcal{E}}_{b'}^{\,m}$ update
  update prediction $\vec{t}_{b',y'}$ (Eqn. 35)
  if not invalid and $\vec{t}_{b',y'} > t$
    add release $\vec{\mathcal{E}}_{b'}^{\,m} := \vec{\mathcal{E}}_{b'}^{\,m} \cup (\vec{t}_{b',y'}, \boldsymbol{\rho}_{b',y'})$
  end if
end for
```

5.4 Approximate solution scheme

The algorithm presented in the previous section yields exact solutions in continuous-time. In general networks the number of events typically grows exponentially because a downstream release event could generates multiple upstream trigger events in case of free-flow conditions, and an upstream release event could generate multiple downstream trigger events in case of spillback. Cyclic dependencies[3] that exist in networks further magnify this problem, in particular when using splitting rates. We refer to Raadsen et al. (2016) for a detailed analysis on this issue. Therefore, for medium and large sized transport networks, the algorithm as presented thus far is not practically feasible. We therefore propose a variation in the algorithm that significantly reduces the number of events generated on the network, although this will in general mean that solutions are no longer exact but rather approximate. In order to reduce the number of events, we need to cut-off the proliferation of events at the link level or node level. Following Raadsen et al. (2016), we apply thresholds at the node level that determine whether events are triggered on adjacent links. Cutting off events at the node level means that all essential properties such as FIFO and conservation of flow are still satisfied within each link, such that the link model is still solved exactly. However, conservation of flow through the node (and hence across links) may be violated, at least locally and/or temporarily, such that the network flows become approximations. Although we lose exactness, as shown in Raadsen et al. (2016) the approximation error is expected to be small with appropriate thresholds. In this paper we assume that each node considers the same threshold value in determining whether to pass the flow rate change on or not. However, we

[3] With cyclic dependency is meant a situation where the flow rate of a link on the upstream link boundary to some degree depends on its flow rate on the downstream link boundary.

note that these thresholds can be set differently across the network. This would allow to maintain flow conservation on parts of the network where the analyst deems this is important, while in other parts of the network this constraint is relaxed to increase computational efficiency. We postpone the analysis of the impact of applying these thresholds in general networks to Section 6.2.4. First, instead we first discuss their concepts.

5.4.1 Flow rate propagation threshold

Flow rate changes are diluted the further one moves away from its source, similar to how water 'ripples' when one throws a stone in a pond. The few links in the direct neighbourhood are affected most, while the relative impact on (a large number of) links far away is often very small. To limit the proliferation of (increasingly small) flow changes across the network, we adopt a flow rate difference threshold q^{Δ} (veh/h) as proposed in Raadsen et al. (2016). After each node model update, accepted turn flow rates are converted to inflow rates on outgoing links (Equation 27) and outflow rates on incoming links $v_a(t) = \sum_{b \in \mathcal{A}_n^+} v_{a,b}(t)$. These updated link flow rates are only effectuated when at least one flow rate's change exceeds flow rate difference threshold q^{Δ}. This measure eventually stops the 'ripple' from progressing and also breaks any remaining cyclic dependency. It only does so when the flow rate change is deemed too insignificant to affect the result (much). Using this measure, they found that, in absence of any routing or fanning effects, for $q^{\Delta} \leq 5$, the results still closely resembled the (near) exact solution. This was demonstrated both in terms of lost vehicle hours (<5% difference) and final cumulative vehicle numbers on the link level. We point out that whenever, due to this threshold, a flow rate change is not propagated over the node, it is still imposed on the link boundary it arrived on in order to satisfy FIFO and flow conservation on the link.

5.4.2 Mixture rate propagation threshold

By introducing multi-commodity flows via mixture events, an additional event stream is introduced alongside the propagation of flow events compared to the algorithm described in Raadsen et al. (2016). Like flow events, mixture events also suffer from the same exponential growth in general networks. To cope with this phenomenon, a second threshold is put into place. Instead of a flow rate difference threshold, a mixture difference threshold ρ^{Δ} is proposed. After each node model update, new upstream mixtures $\underline{\rho}_{a,m}(t)$ are computed for the outgoing links compliant with (26). Per outgoing link, new mixture rates will only be effectuated when at least one mixture component m has changed more than the adopted mixture difference threshold ρ^{Δ} (which is expressed as a relative change in mixture).

6 Numerical results

In this section we analyse two hypothetical networks and one real-life network. All case studies adopt the Quadratic-Linear FD (see Part I) unless specified otherwise. The strictly concave hypocritical branch allows more realism than a linear branch (especially around the saturation point), while as argued in Part I there is little evidence that there is a real need for a strictly concave hypercritical branch.. Further, in all case studies we use the following intermediate hypocritical flow rates $\{q_{i,1}, \ldots, q_{i,D_i-1}\}$ in Equation (8) to simplify fanning in case of a flow rate increase event *i*:

$$q_{i,d} = u_{i-1} + \frac{d}{D_i}(u_i - u_{i-1}), \;\; 1 \leq d \leq D_i, \quad \text{with } D_i = \left\lceil \frac{u_i - u_{i-1}}{\lambda^{\Delta}} \right\rceil, \tag{36}$$

where $\lambda^{\Delta} > 0$ (veh/h) is a user defined maximum allowed step size in terms of flow rate increase in each fanning segment *d*, and operator $\lceil \cdot \rceil$ rounds the value up to the nearest integer. In other words, Equation (36) states that we assume $D_i + 1$ equidistant flow rates in the D_i-step inner linearization of the FD. As noted in Part I, many other rules could be considered for determining the number of steps and the location of the points on the FD. Our proposed rule is simple and straightforward, but we acknowledge that other rules may exist that provide a better linear approximation of the FD. If we set $\lambda^{\Delta} = \infty$, then $D_i = 1$ and fanning is greatly simplified by not considering any intermediate flow rates. If λ^{Δ} is set to a small value, then many points on the FD will be used as intermediate flow rates in the linear interpolation, which means that fanning is simplified only marginally.

The first hypothetical network is a simple corridor network in which there exists no route choice. This network is used to demonstrate how flow rate events propagate forwards and backwards in the (exact) algorithm, and to illustrate how fanning is simplified. The second hypothetical network is a 'Manhattan' (grid) network in which also mixture rates are propagated. This grid network contains many interactions and is deemed sufficiently complex to investigate properties of the algorithm with respect to different settings for thresholds q^{Δ} and ρ^{Δ} (to obtain approximate solutions) as well as maximum step size λ^{Δ}, while still small enough to allow running many simulations and obtain a (near) exact solution. Finally, we apply the algorithm to a real world transport network of the Gold Coast (Queensland, Australia), which is available for download via Bar-Gera (2016). We will use the Gold Coast network to demonstrate feasibility of the algorithm on large scale networks. Further, this network is also used to compare eGLTM to its simplified sibling eLTM in order to assess the influence of a nonlinear hypocritical branch on computational efficiency. It is also used to provide some insight in the contribution of the various DTA components (route choice, route cost, DNL) to the overall computational cost in a practical context.

6.1 Corridor network

The corridor network consists of six links of equal length $(L=1)$ containing single path p. All but the last link have the following characteristics: $\sigma^{\max}=120$, $\sigma^{\text{crit}}=80$, $q^{\max}=4000$, $k^{\max}=180$. The last link is the same as the other links, only its capacity is reduced to $q^{\max}=500$. Three departure time periods with varying demand are imposed. The first period $(t\in[0,100)$ in seconds) carries a small initial flow rate $f_{p,1}=400$ (veh/h). The second period $(t\in[100,150))$ carries a flow rate of $f_{p,2}=4000$ while the third period $(t\in[150,800))$ has a slightly lower flow rate of $f_{p,3}=3000$. Simplified fanning is applied with $\lambda^{\Delta}=400$ (veh/h). The results are shown in Figure 8, where each line represents the cumulative inflow of each link while the last line is the cumulative outflow of the last (bottleneck) link.

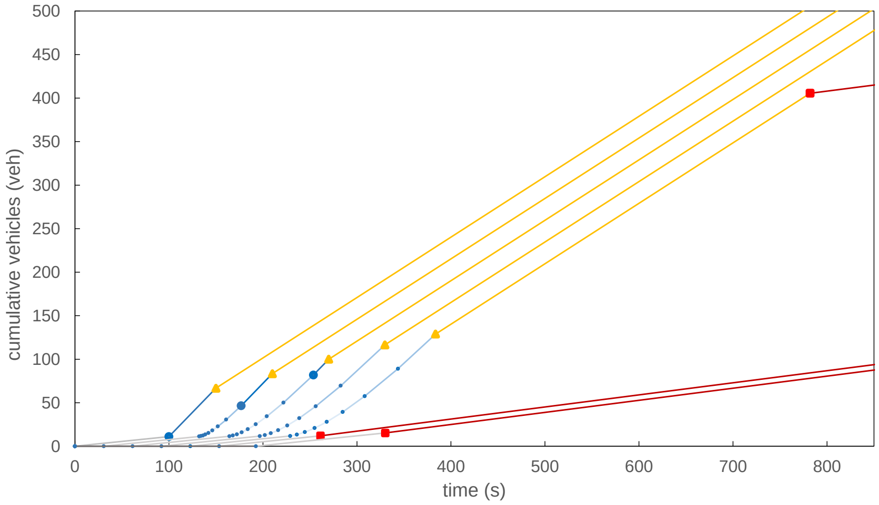


Figure 8: Corridor network showing cumulative inflow curves of each link as well as the cumulative outflow curve of the last link.

First we observe that the initial flow increase from 0 to 400 (veh/h) depicted in the grey lines, does not cause multiple fanning steps since this increase is within the maximum allow step size $\lambda^{\Delta}=400$. Secondly, the flow rate increase from 400 to 4000 (veh/h) does create additional fanning steps. It produces 9 steps based on Equation (36), depicted by the blue dot markers. We point out that, on the first link, this flow rate increase therefore generates 9 release events. On subsequent links however, each flow rate increase that arrives is again smaller than or equal to $\lambda^{\Delta}=400$, such that a 1-step linearization suffices. So even though the "ripple" of the fan disperses, it does not continue to increase the number of fanning events. This observation is important, because it implies that the performance penalty for adding fanning might be relatively small (see also the next two sections). At time $t=150$

the flow rate drops from 4000 to 3000 (veh/h), and no fanning events need to be generated. The propagation of this flow rate (yellow line segments and triangle markers) is faster than the preceding final fanning wave (carrying the flow rate change from 3600 to 4000 veh/h). This results in the final fanning step flow rate (large blue dot markers) to be absent on link 4 and beyond since it no longer dominates the lower envelope within its period of influence. Link 6 is the bottleneck link, the high flow rate generates a backward shockwave on link 5 which reaches the upstream link boundary at the point of the red squared marker where it enforces a receiving flow equal to the capacity of link 6 (500 veh/h), such that link 5 becomes in a spillback state

This example demonstrates the basic workings of the proposed implicit-event based algorithm: It computes events only at times the flow rate changes on any of its link boundaries, is capable of yielding exact results and can be applied to a (simple) network. We now proceed to a more complex network to explore the best settings for generating approximate results.

6.2 Grid network

The network used holds 25 zones in a 5-by-5 grid. There are 96 nodes and 220 identical directional links with the following characteristics: $\sigma^{\max}=50$, $q^{\max}=1800$, $\sigma^{\text{crit}}=35$, $k^{\max}=180$, $L=1$. Figure 9 shows the grid structure. A predefined route set is used containing 1084 routes. This route choice set is generated based on Fiorenzo-Catalano et al. (2004). We refer the reader to Appendix C.1 for the parameters used to construct this route set.

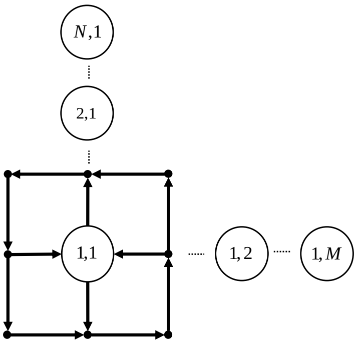


Figure 9: Grid structure for any N-by-M grid, holding NM zones.

6.2.1 Calibration of on-the-fly-linearization of the quadratic linear fundamental diagram

We now investigate different settings for the maximum allowed fanning step size λ^{Δ}, which we assume takes a value between 1 and 600 veh/h, and analyse the impact on network travel time, ceteris paribus. The flow threshold is set to $q^{\Delta}=0.3$ veh/h to obtain a near-exact solution, removing as much potential interference as possible caused by other approximation errors. Route choice is fixed since it is not important when investigating simplified fanning steps. The grid network is configured as follows; demand between each inter-zonal origin-destination (OD) pair is randomly generated by drawing a number from a uniform distribution between 0 and 80 veh/h. This creates congestion or spillback on $13/220\approx 6\%$ of all links during simulation[4]. We refer to Appendix C.2 for the OD-matrix used, simulation settings are presented in Table 2. Also, a uniformly applied departure profile is adopted, see also Figure 10.

[4] This number was obtained using $\lambda^{\Delta}=5$ veh/h.

Table 2: Grid network, simplified fanning analysis simulation parameters.

Parameter	Value
Simulation period	14,400s (4h)
Initial Route choice method	MNL (Multinomial logit)
Logit scale parameter	-7
Output granularity[5]	300s

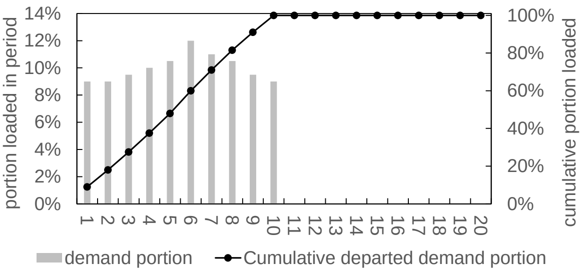


Figure 10: Departure profile, 20 periods of 12 minutes.

6.2.2 Measuring information loss

To identify suitable parameter settings resulting in “good” approximate solutions, we adopt a metric that captures network travel time. This metric utilises the notion of lost vehicle hours (which equals link travel time minus the minimum free-flow travel time) on each link over a certain time interval $[t,t')$, and is denoted by $\tau_a^+(t,t')$. It can be calculated as:

$$\tau_a^+(t,t') = \left(U_a(t) - U_a(t'-t)\right)\left(\frac{L_a}{\bar{\sigma}_a(t,t')} - \frac{L_a}{\sigma_a^{\max}}\right), \tag{37}$$

with $\bar{\sigma}_a(t,t')$ denoting the space mean speed on link a over time interval $[t,t')$. Summing over all links and time periods yields the network’s lost vehicle hours. We then obtain the information loss by comparing lost vehicle hours of an approximate scenario to the (near) exact solution.

6.2.3 Impact of simplified fanning on lost vehicle hours

By incorporating simplified fanning, eGLTM, unlike eLTM, is capable of capturing hypocritical delay. This means that even in free flow conditions vehicles typically no longer have a speed of $\sigma^{\max}$. this contributes to the lost vehicle hours of Equation (37), which are absent when adopting a triangular FD. To be able to differentiate between the hypercritical and hypocritical contribution to lost vehicle hours, hypercritical delay was first measured in isolation by adopting a triangular FD. This amounted to a total of 296.7 hours of delay caused by congestion only. Next, the hypocritical delays for representative fanning step sizes were obtained as shown in Figure 11(b). For this network, under mild congestion, hypocritical delay accounts for roughly 68% of total lost vehicle hours. We then further isolate the impact of the multi-step expansion fans within the hypocritical delay by comparing it to the 1-step linearization. These results are depicted in Figure 11(c). Notice how multi-step fanning contributes negatively (if at all); this is because a single-step linearization overestimates the lost vehicle hours as discussed in the previous section and in part I. From these results we can tentatively conclude the following:

- The ability to model hypocritical delay is important in capturing travel time information. In this particular case it contributes up to 68% of the total lost vehicle hours measured.
- The ability to model multi-step expansion fans is unlikely to significantly reduce travel time times compared to the 1-step linearization. The maximum effect measured is <0.1% for this general network case study.

[5] Each interval holds the cumulative inflow rate and space mean speed used to compute lost vehicle hours for each link within the interval compliant with Equation (37).

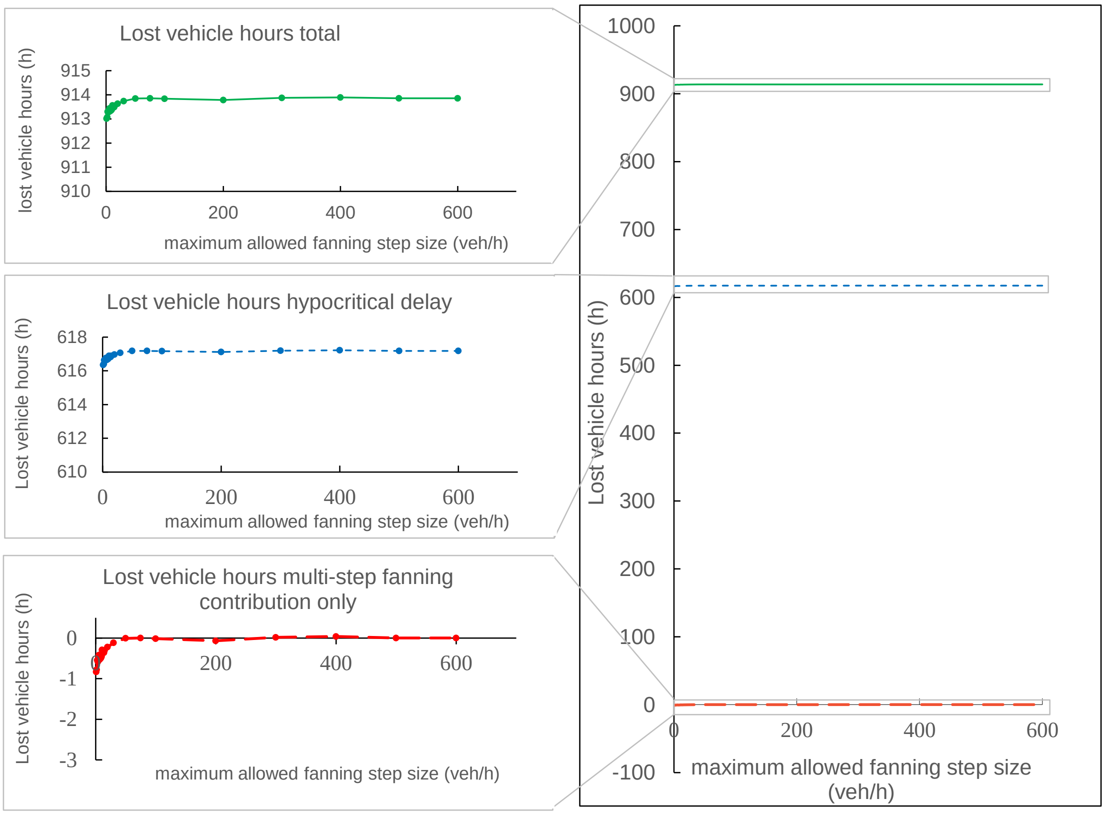


Figure 11: Lost vehicle hours per fanning step size. The insets on the left are close-ups of the respective delay components: Total lost vehicle hours, hypocritical delay and multi-step fanning contribution within hypocritical delay.

Multi-step expansion fans only contribute measurably to lost vehicle hours (compared to s1-step) when modelled at very small maximum allowed fanning step sizes ($\lambda^{\Delta} \leq 5$). Effects of expansion fans dilute quickly as flow propagates on the network, hence it is only measurable at very small fanning steps. That said, it is clear that even the most detailed multi-step expansion fans do not significantly reduce information loss (<0.1%). Figure 12 shows the number of generated events (logarithmic scale) as a function of λ^{Δ} for this case study. Here we see that adopting a small λ^{Δ} results in an exponential growth of events, and thus computation time (we refer to Raadsen et al. (2016), for a more detailed analysis of the relationship between the number of generated events and simulation time). We therefore conclude that in most cases and especially for strategic planning purposes, it does not seem worthwhile to model simplified fanning in a very detailed fashion.

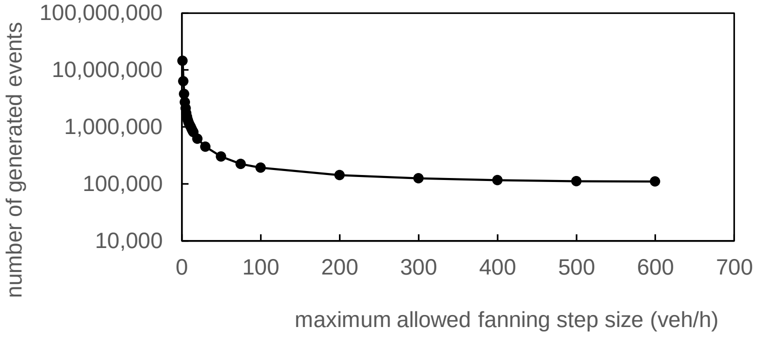


Figure 12: Number of generated events as a function of chosen fanning step size.

6.2.4 Joint Calibration of flow mixture and flow rate propagation threshold

When the mixture of a flow changes, it impacts on downstream flow rates whenever these flows are disentangled, for example at a diverge node. So, choosing a threshold to stop propagating changes in the flow mixture impacts the flow rate propagation. Conversely, a flow rate change also impacts the flow mixture. This occurs when flow rates merge, for example at an on-ramp or intersection. So, choosing a threshold to cut-off flow rate changes impacts on the flow mixture propagation. Due to this interaction, investigating the possible values for the flow and mixture thresholds is analysed in unison.

This analysis provides insight into the trade-off between the chosen threshold settings and computational cost. We adopt the mixture distribution method as presented in Section 3.2.2 to model multi-commodity flows. We isolate the effect of the chosen flow and mixture thresholds by looking at the first iteration only, this prevents the route choice from interfering. For the impact of route choice over iterations we refer to the next section. Eight 30 minute departure time periods are modelled where each period coincides with the route choice periods. Furthermore, 3 different OD matrices (see Appendix C.3) are applied, each of which spans one or more departure time periods. This guarantees a significant amount of change in the flow mixtures and flow rates over time. Simulation parameters are provided in Table 3. Given this input, $29/220 \approx 13\%$ of the links experience congestion or spillback at least once during the simulation[6].

Table 3: Simulation parameters.

Parameter	Value
Simulation period	14,400s (4h)
Route choice interval	1800s (0.5h)
OD matrix durations	5400s, 7200s, 1800s
Route choice method	MNL (Multinomial logit)
Logit scale parameter	-7
Output granularity	600s
Simplified fanning	1-step

Seven different values of $q^{\Delta} \in \{0.1, 1, 2, 3, 4, 5, 6\}$ are considered which encompass the range deemed acceptable according to Raadsen et al. (2016). In addition, 19 different values of $\rho^{\Delta} \in \{0.1, 0.2, \ldots, 0.9, 1, 2\ldots, 10\}$ are investigated in conjunction. The rationale behind the explored values of ρ^{Δ} is based on the minimum value being so small it can be expected to result in flow changes around or below 1 veh/h[7], while the maximum allows for flow changes well above the acceptable range in Raadsen et al. (2016). This leads to a total of $7 \cdot 19 = 133$ simulation runs of which the results are presented in Figure 13.

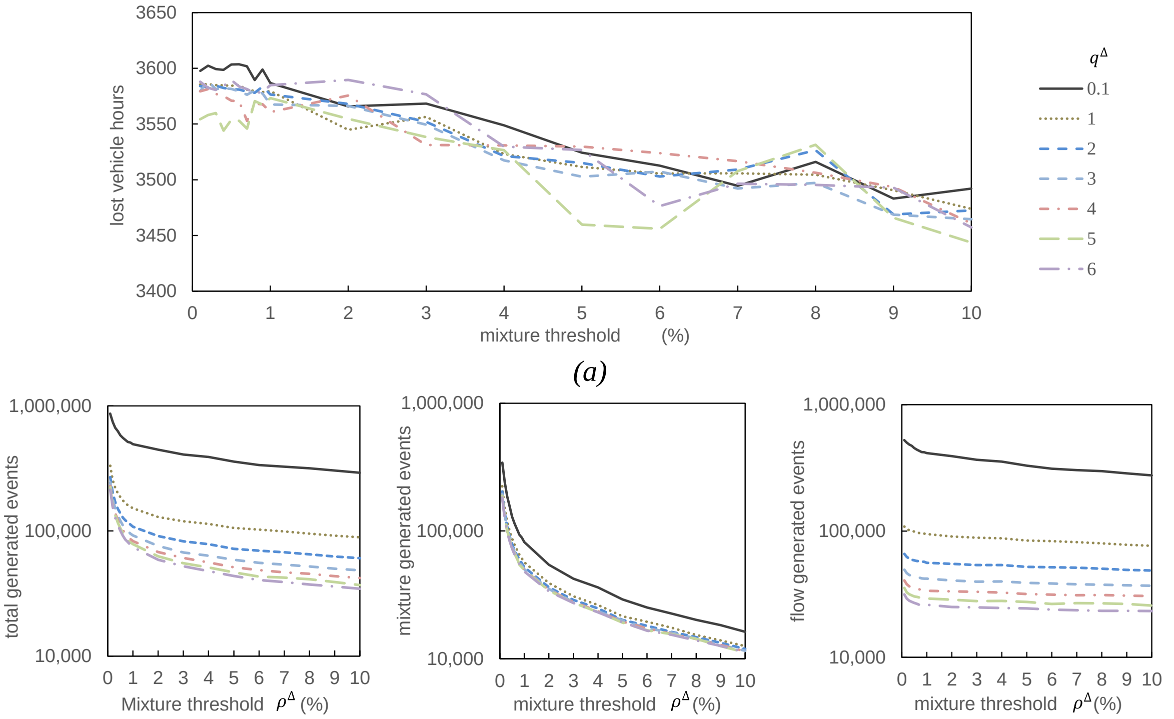
Figure 13(a) shows that in a multi-commodity context, the range of q^{Δ} that limits information loss to within 5% of the near exact solution ($q^{\Delta} = 0.1$ (veh/h), $\rho^{\Delta} = 0.1\%$) is in line with the findings regarding this threshold in Raadsen et al. (2016). We also observe that in general choosing a cruder q^{Δ} and/or cruder ρ^{Δ} both cause more information loss as is to be expected. Observe that for levels of ρ^{Δ} up to 4%, differences in lost vehicle hours remain below the 2% mark for all q^{Δ}, while the largest difference found is still smaller than 4%. This might be explained by the fact that each mixture event carries all components of flow $\boldsymbol{\rho}_y$ and therefore mixture information of components that change less than ρ^{Δ} often still get updated (unlike flow rates). It only takes one mixture component to exceed the threshold for the other components to profit from it, as a result of this information being bundled within each event.

Looking at the number of generated events in Figures 13(b)-(d), where we point out the logarithmic vertical axis, we observe the following: (i) The total number of events is sensitive to both q^{Δ} and ρ^{Δ} (Figure 13b). (ii) The number of generated mixture events exponentially decreases by with a larger ρ^{Δ}

[6] Value obtained from simulation runs where $q^{\Delta} = 5, \rho^{\Delta} = 2.$

[7] As a reference point we take a high capacity link (running at capacity), with $q^{\max} = 8000.$ Flow is assumed to be uniformly distributed between the departure time periods. In such a case a 0.1% mixture change can result in at most $0.001 \cdot 8000/8 = 1\text{veh/h}.$

(Figure 13c). (iii) The number of mixture events is relatively insensitive to choosing the level of q^{Δ} (again, Figure 13c). (iv) The number of flow events is somewhat sensitive to the level of ρ^{Δ}, but this effect quickly diminishes for values of $\rho^{\Delta} \geq 1\%$ (Figure 13(d)). (v) The number of mixture events is typically less than the number of flow events as long as one chooses relatively a mixture thresholds of $\rho^{\Delta} \geq 3\%$ (Figure 13(d-e)).



(a)

(b) *(c)* *(d)*

Figure 13: (a) Lost vehicle hours vs mixture threshold for different q^{Δ} with truncated y-axis, (b) total generated events, (c) generated mixture events only, (f) generated flow events only.

From these findings we conclude the following: The mixture threshold is best chosen between 2% and 4% which leads only to small approximation errors while the number of generated events is reduced significantly compared to the near exact solution. Further, the impact of q^{Δ} is demonstrated to be in line with earlier findings suggesting $q^{\Delta} \leq 5\%$ is acceptable. Lastly, we find a seemingly linear relation between cruder chosen thresholds and an increase in information loss in terms of lost vehicle hours not being captured.

6.2.5 Equilibrium and convergence testing

While the focus of this paper is on DNL and path proportions are assumed to be given, we want to demonstrate this DNL implementation is suitable for DTA by searching for a conditional SUE equilibrium[8] in our grid network. To do so, we compute a (stochastic) duality gap based on Bliemer et al. (2014) for each route choice period for each iteration simulated. The cost for each path is solely determined by travel time which in turn is collected from a probe vehicle departing within the period (at 10% offset from period start time). Simulation settings are identical to those provided in Table 3 and in addition we choose $\rho^{\Delta} = 3\%, q^{\Delta} = 5$ (veh/hr), based on findings in the previous sections. The results are shown in Figure 14, where each line denotes the stochastic duality gap per route choice period. As can be seen all route choice periods converge quite quickly to an acceptable gap of < 0.00001, and as expected, later periods have more trouble converging than earlier periods due to their increased interdependency. The small spikes are the result of an earlier period making a small convergence “mistake” trickling down to successive periods making a temporarily larger “mistake”. This brings us

[8] It is conditional because its path set is fixed and generated a-priori.

to discuss the final and largest case study: The Gold Coast case study, which investigates the computational efficiency of e(G)LTM under different simplified fanning schemes on large scale networks as well as investigating the cost per DTA component, i.e. route choice versus DNL.

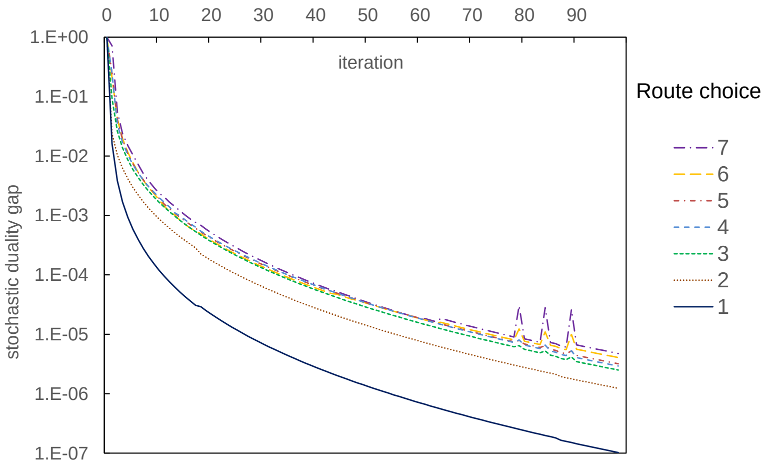
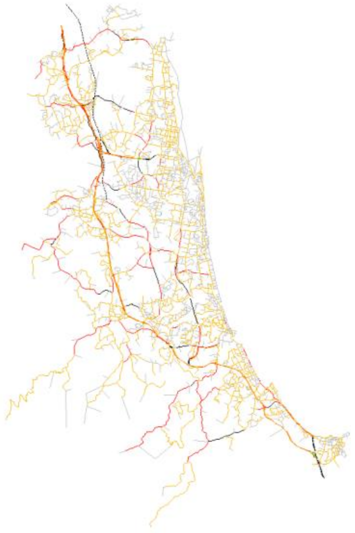


Figure 14: Convergence per route choice period.

6.3 Gold Coast network

The Gold Coast network is a large scale, real world, strategic planning network containing suburban and urban areas with a detailed zoning system. The network and its simulation settings are provided in Figure 15. As stated earlier, we are interested in the relative performance, both in terms of simulation time and information loss, of the newly proposed eGLTM versus the original eLTM model. In addition we verify whether the findings correspond with the results found on the smaller grid network. To do this we analyse the results for three different FD's: Triangular (eLTM), Quadratic-Linear with 1-step simplified fanning and Quadratic-Linear with multi-step simplified fanning with $\lambda^{\Delta}=50$ (veh/h). We also consider three different flow difference threshold values: $q^{\Delta}=1,2,5$ (veh/h) as well as three different mixture threshold values $\rho^{\Delta}=1,2,5\%$.



Simulation:

- $T=10{,}800s$ (3h).

Network:

- 1,068 zones
- 4,675 nodes
- 11,140 links
- Total length: 2,818km

OD Matrix:

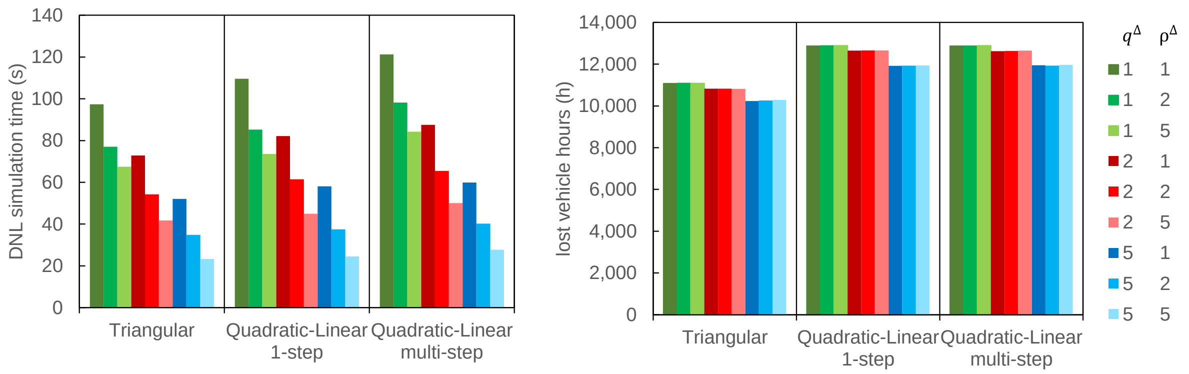
- Synthetic morning peak period with demand profile: $[0-3600:24\%, 3600-7200:57\%, 7200-T:19\%]$

Routes, route choice, route cost

- Fixed route set with 1,104,374 pre-generated routes
- MNL with logit scale parameter set to -7.
- Route choice periods coincide with dep. time periods
- Trajectory travel time with probe vehicle dep. time at 60% of time period.

Figure 15: Gold Coast network and simulation settings

We examine the results of a single iteration because for this case study we are interested in (relative) computational cost only. The results are provided in Figure 16.



(a) *(b)*

Figure 16: (a) DNL propagation times for Gold Coast[9]*, (b) absolute number of lost vehicle hours.*

From Figure 16(a) we find that using a quadratic-linear FD is consistently more costly than a triangular FD. This is to be expected when we apply multi-step fanning, while it is less obvious for 1-step fanning. A triangular FD and or a quadratic-linear FD with 1-step fanning only generates one wave carrying a flow rate change. Theoretically a quadratic-linear FD with 1-step fanning can even result in less accepted events due to "overtaking" flow rate changes within the hypocritical branch (see also Part I). The reason it is still slightly more costly is because more invalid predictions and updates are required due to increased (shock)wave interaction. Further, since changes in either variable (flow threshold, mixture threshold, FD choice) result in different simulation times, we conclude that each of the three considered variables impact the computational cost. Overall the DNL simulation time for such a large network is found to be between 20 and 100s for eLTM and moving to a concave hypocritical branch still yields relatively low computation times. In relative terms, the additional cost of switching from the triangular FD to a quadratic-linear (1- step) is 5.5-13%, whereas multi-step with $\lambda^{\Delta} = 50$ increases the cost 15-27.5% compared to the base case. From Figure 16(b) we find that the lost vehicle hours, i.e. information loss, is virtually identical for the two types of fanning approaches, which confirms the earlier findings, that there is little point in modelling multi-step fanning in strategic planning applications. Most importantly, we see that we do find a significant difference in lost vehicle hours between the triangular and quadratic-linear results. This difference amounts to roughly 15% of the total (Figure 16(b)). Finally, we observed that the lost vehicle hours seem to be invariant (in terms of relative differences regarding q^{Δ} and ρ^{Δ}) to the chosen FD. While we only examined three different FD approaches this might indicate that these findings possibly extend to beyond the FDs considered in this paper.

7 Conclusions and future research

In this paper we presented a novel implicit event-based solution scheme for solving continuous-time general link transmission models on large networks. We rewrote the formulation presented in Part I to an event-based formulation that can be used to find exact or approximate grid free solutions with simplified fanning. The proposed DNL framework supports different models within a more general context, where any non-linear concave two regime fundamental diagram is supported. Raadsen et al. (2016) is in fact a special (limited) case of this more general formulation. In addition the model was extended from single commodity to multi-commodity. Path based information is supported by proposing an additional stream of mixture events alongside the flow propagation. Mixtures are agnostic to the information they carry, allowing a range of different modelling approaches from fully path based to more aggregated approaches with periodically fixed splitting rates.

One of the main findings is that in the proposed implicit event-based solution scheme, the computational penalty for moving to a non-linear concave hypocritical branch is quite modest . At the same time,

[9] Results obtained on PC with Intel Xeon 3.1 GHz, Windows7 SP1. Implementation written in C++ (compiler: 32bit C++ MS Visual Studio 2008 SP1).

numerical results on a grid network and the Gold coast network demonstrated that adopting this type of FD has significant (positive) impact on the ability to capture network wide hypocritical delay compared to the popular but more limited triangular FD. The increase in computational cost of supporting this type of FD was found to be in the range of 5.5-27.5% (depending on the settings for simplified fanning) compared to adopting a triangular FD (following proposed approximation settings). We also demonstrated that adopting a 1-step inner linearization to simplify expansion fans seems to suffice for strategic planning purposes. While increasing the number of steps is possible and will more accurately describe expansion fans, it does little to reduce information loss.

Further, investigating the balance between information loss and reduced computational efficiency by choosing different parameter settings resulted in the following: An arguably "good" compromise between exactness and computational cost is found by choosing a flow rate difference threshold $1 \leq q^{\Delta} \leq 5,$ and a mixture percentage difference threshold $2 \leq \rho^{\Delta} \leq 4$. The large scale Gold Coast case study reveals that with $q^{\Delta} = 5,$ and $\rho^{\Delta} = 2$ it takes about 40 seconds to solve the multi-commodity DNL (with three route choice/departure time periods) for a single route choice iteration.

Future research directions are twofold: The adopted thresholds and adopted FD have now been set rigidly to the same value across the network. More intelligent and adaptive thresholds could further reduce information loss as well as reduce the computation burden. Of interest is also to develop destination-based, path-based and/or other related mixture distribution methods within our mixture framework. Having implementations available alongside each other would allow for consistent comparisons between these different types of approaches.

References

Bliemer, M.C.J., 2001 Analytical dynamic traffic assignment with interacting user-classes: Theoretical advances and applications using a variational inequality approach. Ph.D. Thesis, Delft University of Technology, The Netherlands.

Bar-Gera, H. 2016. Transportation network test problem, https://github.com/bstabler/TransportationNetworks.

Bliemer, M.C.J., Raadsen, M.P.H., (under review). Continuous-time general link transmission model with simplified fanning (Part I Theory and link mode formulation).

Bliemer, M.C.J., Raadsen, M.P.H., Smits, E.-S., Zhou, B., Bell, M.G.H., 2014. Quasi-dynamic traffic assignment with residual point queues incorporating a first order node model. Transp. Res. Part B Methodol. 68, 363–384. doi:10.1016/j.trb.2014.07.001

Carey, M., 1987. Optimal time varying flows on congested networks. Oper. Res. 35, 58–69.

Cascetta, E., 2009. Transportation Systems analysis. Second edition. Springer

Coclite G.M., Garavello M., P.B., 2005. Traffic flow on a road network. Soc. Ind. Appl. Math. 36, 1862–2886. doi:10.1137/S0036141004402683

Chen, W., 2009. Front tracking algorithm for the lighthill-whitham-richards traffic flow model with a piecewise quadratic, continuous, non-smooth, and non-concave fundamental diagram. Int. J. Numer. Anal. Model. 6, 562–585.

Claudel, C., Bayen, A.M., 2010. Lax–Hopf Based Incorporation of Internal Boundary Conditions Into Hamilton-Jacobi Equation. Part II: Computational Methods. IEEE Trans. Automat. Contr. 55, 1158–1174. doi:10.1109/TAC.2010.2045439

Daganzo, C.F., 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transp. Res. Part B 28, 269–287. doi:10.1016/0191-2615(94)90002-7

Daganzo, C.F., 1995. The cell transmission model, part II: Network traffic. Transp. Res. Part B Methodol. 29, 79–93. doi:10.1016/0191-2615(94)00022-R

Daganzo, C.F., 2005. A variational formulation of kinematic waves: basic theory and complex boundary conditions. Transp. Res. Part B Methodol. 39, 187–196. doi:10.1016/j.trb.2004.04.003

Fiorenzo-Catalano, S., van Nes, R., Bovy, P.H.L., 2004. Choice set generation for multi-modal travel analysis. European Journal of Transport and Infrastructure Research 4 (2), 195–209.

Flötteröd, G., Rohde, J., 2011. Operational macroscopic modeling of complex urban road intersections. Transp. Res. Part B Methodol. 45, 903–922. doi:10.1016/j.trb.2011.04.001

Friesz, T.L., Luque, J., Tobin, R.L., Wie, B., 1989. Dynamic Network Traffic Assignment Considered As a Continuous Time Optimal Control Problem. Oper. Res.

Gentile, G., 2010. The General Link Transmission Model for Dynamic Network Loading and a comparison with the DUE algorithm, in: New Developments in Transport Planning: Advances in Dynamic Traffic Assignment (Chapter 8). Edward Elgar, pp. 153-178.

Gibb, J., 2011. Model of Traffic Flow Capacity Constraint Through Nodes for Dynamic Network Loading with Queue Spillback. Transp. Res. Rec. J. Transp. Res. Board 2263, 113–122. doi:10.3141/2263-13

Henn, 2003. A wave-based resolution scheme for the hydrodynamic LWR traffic flow model, in: Traffic and Granular Flow. pp. 105–124.

Han, K., Piccoli, B., Friesz, T.L., Yao, T., 2012. A Continuous-time Link-based Kinematic Wave Model for Dynamic Traffic Networks 23.

Himpe, W., Corthout, R., Tampère, M.J.C., 2016. An efficient iterative link transmission model. Transp. Res. Part B Methodol. 92, 170–190. doi:10.1016/j.trb.2015.12.013

Hoogendoorn, S., Knoop, V., 2012. Traffic flow theory and modelling. Transp. Syst. Transp. Policy An Introd. 125–159.

Janson, B.N., 1991. Dynamic traffic assignment for urban road networks. Transp. Res. Part B Methodol. 25, 143–161. doi:10.1016/0191-2615(91)90020-J

Jin, W., Zhang, H., 2004. Multicommodity kinematic wave simulation model for network traffic flow. Transp. Res. Rec. J. 1883, 59–67. doi:10.3141/1883-07

Jin, W., 2014. Continuous formulations and analytical properties of the link transmission model. Transp. Res. Part B 74, 27. doi:10.1016/j.trb.2014.12.006

Lighthill, M.J., Whitham, G.B., 1955. On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads. Proc. R. Soc. A Math. Phys. Eng. Sci. 229, 317–345. doi:10.1098/rspa.1955.0089

Lebacque, J., Khoshyaran, M., 2005. First order macroscopic traffic flow models: intersections modeling, network modeling. In: Proceedings of the 16th ISTTT.

Mazaré, P.-E., Dehwah, A.H., Claudel, C.G., Bayen, A.M., 2011. Analytical and grid-free solutions to the Lighthill–Whitham–Richards traffic flow model. Transp. Res. Part B Methodol. 45, 1727–1748. doi:10.1016/j.trb.2011.07.004

Messmer A., Papageorgiou M., 1990. METANET: a macroscopic simulation program for motorway networks, Traffic Engineering and Control, 31, p466–470.

Newell, G.F., 1993. A simplified theory of kinematic waves in highway traffic, part II: Queueing at freeway bottlenecks. Transp. Res. Part B Methodol. 27, 289–303. doi:10.1016/0191-2615(93)90039-D

Peeta, S., Ziliaskopoulos, a. K., 2001. Foundations of dynamic traffic assignment: The past, the present and the future. Networks Spat. Econ. 233–265. doi:10.1023/A:1012827724856

Payne H.J., 1971. Models of freeway traffic and control. Mathematical models of public systems. Simulation council proceedings, volume 1, p51-61.

Raadsen, M.P.H., Bliemer, M.C.J., Bell, M.G.H., 2016. An efficient and exact event-based algorithm for solving simplified first order dynamic network loading problems in continuous time. Transp. Res. Part B Methodol. doi:10.1016/j.trb.2015.08.004

Richards, P.I., 1956. Shock Waves on the Highway. Oper. Res. 4, 42–51. doi:10.1287/opre.4.1.42

Smits, E.-S., Bliemer, M.C.J., Pel, A.J., van Arem, B., 2015. A family of macroscopic node models. Transp. Res. Part B Methodol. 74, 20–39. doi:10.1016/j.trb.2015.01.002

Tampère, C.M.J., Corthout, R., Cattrysse, D., Immers, L.H., 2011. A generic class of first order node models for dynamic macroscopic simulation of traffic flows. Transp. Res. Part B Methodol. 45, 289–309. doi:10.1016/j.trb.2010.06.004

van Wageningen-Kessels, F., van Lint, H., Vuik, K., Hoogendoorn, S., 2014. Genealogy of traffic flow models. EURO J. Transp. Logist. 4, 445–473. doi:10.1007/s13676-014-0045-5

Wu, X., Liu, H.X., 2011. A shockwave profile model for traffic flow on congested urban arterials. Transp. Res. Part B Methodol. 45, 1768–1786. doi:10.1016/j.trb.2011.07.013

Yperman, I., Logghe, S., Immers, B., 2005. The Link Transmission Model : an Efficient Implementation of the Kinematic Wave Theory in Traffic Networks. Adv. OR AI Methods Transp.

Yperman, I., 2007. The Link Transmission Model for Dynamic Network Loading. Katholieke Universiteit Leuven.

Appendix A: Hypercritical event-based flow propagation formulations

The lower envelope problem formulation under spillback conditions:

$$\overleftarrow{V}(t) = \min\left\{\min_{j\in\mathcal{J}(t)} \overleftarrow{V}_j(t),\ \min_{j\in\mathcal{J}_F(t)} \overleftarrow{F}_j(t\,|\,\cdot)\right\}, \tag{A.1}$$

with upstream regular periods of influence time interval $[\overleftarrow{t}_j^{\min}, \overleftarrow{t}_j^{\max}]$ where $\overleftarrow{t}_j^{\min} = t_j + L/\gamma_{II}(v_j)$, and $\overleftarrow{t}_j^{\max} = t_{j+1} + L/\gamma_{II}(v_j)$, such that

$$\mathcal{J}(t) = \left\{ j = 1,\ldots,J \,|\, \overleftarrow{t}_j^{\min} \le t < \overleftarrow{t}_j^{\max} \right\}, \tag{A.2}$$

$$\mathcal{J}_F(t) = \left\{ j = 1,\ldots,J \,|\, \overleftarrow{t}_{j-1}^{\max} \le t < \overleftarrow{t}_j^{\min} \right\}, \tag{A.3}$$

The cumulative segment projections are given by

$$\overleftarrow{V}_j(t) = \begin{cases} V(t_j) + v_j\left((t - t_j) - \dfrac{L}{\gamma_{II}(v_j)}\right) + \xi_I(v_j), & \text{if } \overleftarrow{t}_j^{\min} \le t \le \overleftarrow{t}_j^{\max}, \\ \infty, & \text{otherwise,} \end{cases} \tag{A.4}$$

$$\overleftarrow{F}_j(t\,|\,q_{j,0},\ldots,q_{j,D_j}) = \begin{cases} V(t_j) + q_{j,0}\left((t - t_j) - \dfrac{L}{\gamma_{II}(q_{j,0})}\right) + \xi_{II}(q_{j,0}), & \text{if } \overleftarrow{t}_{j-1}^{\max} \le t \le t_j + \dfrac{L}{\eta_{II}(q_{j,0}, q_{j,1})}, \\ \vdots & \\ V(t_j) + q_{j,d}\left((t - t_j) - \dfrac{L}{\gamma_{II}(q_{j,d})}\right) + \xi_{II}(q_{j,d}), & \text{if } t_j + \dfrac{L}{\eta_{II}(q_{j,d-1}, q_{j,d})} \le t \le t_j + \dfrac{L}{\eta_{II}(q_{j,d}, q_{j,d+1})}, \\ \vdots & \\ V(t_j) + q_{j,D_j}\left((t - t_j) - \dfrac{L}{\gamma_{II}(q_{j,D_j})}\right) + \xi_{II}(q_{j,D_j}), & \text{if } t_j + \dfrac{L}{\eta_{II}(q_{j,D_j-1}, q_{j,D_j})} \le t \le \overleftarrow{t}_j^{\min}, \\ \infty, & \text{otherwise.} \end{cases} \tag{A.5}$$

We generalise the periods of influence such that we can merge (A.4) and (A.5):

$$\overleftarrow{t}^{\min}_{j,d} = \begin{cases} t_j + \dfrac{L}{\eta_{II}(v_{j,d-1}, v_{j,d})}, & \text{if } v_j > v_{j-1} \text{ and } d > 1, \\ t_j + \eta_{II}(v_{j-1,D_{j-1}-1}, v_{j,d}), & \text{if } v_j > v_{j-1} \text{ and } d = 1, \\ t_j + \dfrac{L}{\gamma_{II}(v_{jd})}, & \text{otherwise.} \end{cases} \tag{A.6}$$

$$\overleftarrow{t}^{\max}_{j,d} = \begin{cases} t_j + \dfrac{L}{\eta_{II}(v_{j,d}, v_{j,d+1})}, & \text{if } v_{j+1} > v_j \text{ and } d < D_j - 1, \\ t_{j+1} + \dfrac{L}{\eta_{II}(v_j, v_{j+1,1})}, & \text{if } v_{j+1} > v_j \text{ and } d = D_j - 1, \\ t_{j+1} + \dfrac{L}{\gamma_{II}(v_{j,d})}, & \text{otherwise,} \end{cases} \tag{A.7}$$

Now, segments originally contained in $\overleftarrow{F}_j(t\,|\,\cdot)$ and $\overleftarrow{V}_j(t)$ are replaced by the more concise $\overleftarrow{V}_{j,d}(t)$.

$$\overleftarrow{V}_{j,d}(t) = \begin{cases} V(t_j) + v_{j,d}\left((t - t_j) - \dfrac{L}{\gamma_{II}(v_{j,d})}\right) + \xi_{II}(v_{j,d}), & \text{if } \overleftarrow{t}^{\min}_{j,d} \le t < \overleftarrow{t}^{\max}_{j,d}, \\ \infty, & \text{otherwise.} \end{cases} \tag{A.9}$$

We point out that v_j in (A.4) is replaced with $v_{j,d}$, this denotes the d^{th} fanning step of flow rate v_j in case $v_j > v_{j-1}$ and otherwise $v_{j,d} = v_j$. We can now rewrite the original lower envelope problem of (A.1) to

$$\min_{(j,d)\in\mathcal{J}(t)} \left\{\overleftarrow{V}_{j,d}(t)\right\}, \tag{A.10}$$

With redefined $\mathcal{J}(t) = \left\{(j,d)\,|\,\overleftarrow{t}^{\min}_{j,d} \le t < \overleftarrow{t}^{max}_{j,d};\ j = 1,\ldots J;\ d = 1,\ldots,D_j - 1\right\}$.

Appendix B: Path based mixture distribution model formulation

A path based formulation for the mixture distribution model is provided. Note that a path $p \in P$ then becomes identical to a mixture component $m \in \mathcal{M}$. Within this section we utilise the path specific notation to highlight the path based nature of this formulation, so $\overline{\rho}_{a,p}$ instead of $\overline{\rho}_{a,m}$, for example. Let us start with the (binary) path specific splitting rate φ_{abp} via

$$\varphi_{a,b,p} = \delta_{a,p}\delta_{b,p}, \quad p \in \mathcal{P}, a \in \mathcal{A}_n^-, b \in \mathcal{A}_n^+. \tag{B.1}$$

This then yields continuous time splitting rate $\varphi_{a,b}(t)$ via

$$\varphi_{a,b}(t) = \sum_{p\in\mathcal{P}} \overline{\rho}_{a,p}(t)\varphi_{a,b,p}, \quad a \in \mathcal{A}_n^-, b \in \mathcal{A}_n^+, \tag{B.2}$$

with downstream mixture $\overline{\rho}_{a,p}(t)$ denoting the portion of flow belonging to path p. The splitting rate of (B.2) serves as input to the flow distribution model discussed in Section 3.2.1, which in turn yields accepted outflow rates. Note that Equations (B.3)-(B.6) remain identical in the path based formulation we only replace the departure time period h with path p, indicating this part of the mixture distribution model is in fact also agnostic to the adopted method (departure time, destination based, path based etc.) and could in fact just have been formulated in terms of mixture components $m \in \mathcal{M}$, if one so desires.

$$v_{a,b,p}(t) = \bar{\rho}_{a,p}(t)\varphi_{a,b,p}v_a(t), \tag{B.3}$$

$$\underline{\rho}_{b,p}(t) = \frac{\sum\limits_{a \in \mathcal{A}_n^-} v_{a,b,p}(t)}{\sum\limits_{a \in \mathcal{A}_n^-} v_{a,b}(t)}, \quad \text{if } b \in \mathcal{A}_n^+, n \in \mathcal{N} \setminus \mathcal{O}. \tag{B.4}$$

Note that in case node $n \in \mathcal{O}$, (26) is replaced with $\underline{\rho}_{b,p}(t) = \delta_{b,p}$. Finally resulting in

$$u_{b,p}(t) = \begin{cases} \underline{\rho}_{b,p}(t)\sum_p f_{p,h}, & \text{if } b \in \mathcal{A}_n^+, n \in \mathcal{O}, \\ \underline{\rho}_{b,p}(t)\sum_{a \in \mathcal{A}_n^-} v_{a,b}(t), & \text{otherwise,} \end{cases} \tag{B.5}$$

and

$$u_b(t) = \sum_{p \in \mathcal{P}} u_{b,p}(t). \tag{B.6}$$

Appendix C: Grid network data

C.1 Route set parameters

Implementation used is part of the OmniTRANS software package version 1.14 (which in turn is based on Fiorenzo-Catalano, 2004). Default parameters were used unless specified otherwise:

Table C.1: non default parameter settings

Non-default parameter	**Value**
iterations	100
variance grow value	0.03
max number of routes (per od)	5
max allowed overlap factor	0.5
max total detour factor	10
Max non common detour factor	10

C.2 OD-matrix for fanning step-size calibration

Table C.2: OD matrix used

Centroids	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	77.784	76.931	77.722	48.245	76.616	14.398	57.137	43.585	7.641	20.228	65.502	49.331	59.706	3.514	15.6	76.523	14.702	55.203	68.145	68.799	33.467	15.327	51.83	72.484
2	33.682	0	72.016	39.453	9.319	57.478	1.432	64.314	17.949	52.826	70.427	39.444	54.784	23.608	19.072	66.341	56.543	56.687	20.455	55.579	2.766	28.379	4.75	31.703	53.797
3	17.479	77.753	0	23.571	56.488	73.434	51.233	0.888	27.956	14.684	48.218	17.832	10.659	67.95	43.535	64.406	29.866	36.605	75.367	53.007	64.596	17.525	70.659	29.266	18.957
4	69.592	12.605	74.677	0	78.116	57.525	60.094	18.595	5.415	47.673	26.54	37.926	37.421	14.314	3.985	22.615	32.769	20.63	48.698	35.389	20.517	43.734	3.675	49.805	66.797
5	79.381	58.008	47.089	46.48	0	13.717	7.514	12.132	11.046	54.036	2.614	68.564	70.442	78.687	59.055	27.658	78.014	16.948	72.175	52.877	6.579	49.535	21.831	63.09	43.493
6	43.272	19.997	51.267	57.103	54.955	0	54.489	2.445	74.581	23.241	41.855	25.686	2.29	27.068	26.48	67.852	35.359	50.07	60.489	64.905	68.935	52.246	67.451	30.485	11.607
7	73.457	30.442	59.048	52.251	8.754	6.35	0	4.726	22.583	18.02	75.779	58.75	78.245	52	34.902	32.343	17.504	55.752	12.175	59.551	24.188	66.283	44.796	52.657	46.564
8	71.829	60.921	45.307	52.723	0.699	30.601	16.397	0	79.052	66.13	56.808	72.13	23.173	77.441	27.804	65.489	14.547	38.842	57.839	48.427	67.05	79.178	43.005	45.301	36.221
9	25.675	13.517	37.144	44.87	8.868	63.194	62.552	73.868	0	14.973	12.632	42.614	24.02	18.926	28.719	45.048	50.63	22.975	29.769	27.362	4.72	4.632	0.719	25.763	45.094
10	8.033	41.254	36.311	69.875	20.756	21.237	26.097	20.1	45.409	0	77.215	9.206	10.369	12.149	37.607	52.425	38.752	21.133	55.901	25.952	52.198	61.053	28.767	75.423	53.3
11	49.169	4.92	28.411	50.726	52.547	17.262	70.052	74.71	40.421	76.669	0	73.801	77.725	76.521	28.278	30.242	29.491	76.497	52.44	24.399	23.419	49.682	7.418	12.326	65.801
12	0.004	57.555	61.32	75.8	57.889	22.966	40.108	68.854	39.372	30.238	21.193	0	27.65	19.284	11.777	68.24	57.194	23.726	25.375	26.937	35.276	9.878	41.194	31.203	56.012
13	49.206	9.12	11.72	44.472	20.313	33.33	38.765	12.493	50.45	50.281	10.952	3.289	0	67.348	64.594	43.148	23.106	28.196	47.671	0.091	13.326	9.019	7.282	14.334	9.696
14	41.853	18.529	49.675	26.896	59.503	13.425	3.563	1.679	47.746	19.678	48.786	32.128	65.628	0	44.369	39.466	8.227	18.923	10.531	13.138	71.382	70.876	44.605	60.477	18.839
15	27.845	78.096	67.733	8.208	74.181	43.812	72.599	27.512	65.465	14.311	68.049	30.683	23.5	52.961	0	17.773	26.779	32.613	55.334	65.386	17.563	14.625	41.877	54.454	44.694
16	79.147	44.045	46.828	13.783	2.225	55.919	47.162	42.456	46.435	39.872	25.409	70.742	0.687	57.734	0.238	0	56.534	35.319	50.519	30.183	56.726	11.511	51.625	66.103	18.816
17	60.608	21.384	17.133	55.613	77.871	6.001	43.099	53.523	13.818	56.439	63.605	57.927	65.472	33.583	55.946	11.079	0	42.452	37.768	21.942	16.715	31.305	12.852	30.937	64.967
18	36.774	49.6	75.242	26.356	26.725	30.221	63.775	11.999	40.697	10.774	37.933	43.219	47.29	20.85	51.955	16.104	16.11	0	77.589	75.328	6.274	13.97	40.113	59.127	38.394
19	52.055	26.22	6.672	57.062	31.987	35.067	31.09	8.259	63.727	16.446	77.154	57.23	47.878	8.324	44.279	56.647	65.695	45.05	0	25.839	72.507	53.595	3.488	73.735	46.663
20	0.784	21.87	76.65	54.362	52.856	19.468	11.417	43.273	64.414	30.712	21.161	35.659	21.557	54.396	50.507	69.298	57.067	42.939	79.436	0	38.376	7.111	5.431	5.265	4.439
21	62.762	42.009	42.206	1.896	16.017	43.385	42.24	21.153	11.476	30.09	40.199	17.447	73.464	14.644	4.388	61.645	27.288	60.638	51.223	15.639	0	50.664	31.697	36.83	62.389
22	65.206	45.816	54.124	71.406	40.955	67.582	31.342	33.462	74.528	38.248	30.714	47.52	1.94	1.968	3.368	63.194	25.936	58.303	31.081	7.838	37.095	0	29.359	51.135	51.64
23	62.4	30.647	32.031	7.669	27.375	47.54	70.508	21.482	69.095	20.132	59.765	54.035	41.112	73.556	50.201	4.929	32.231	46.279	60.852	47.523	44.044	18.444	0	1.648	16.367
24	5.404	20.51	41.105	68.751	37.559	60.542	14.138	30.031	8.405	29.917	61.789	60.795	19.664	52.51	70.966	33.007	17.378	14.845	9.514	28.651	15.002	41.739	20.285	0	73.269
25	64.598	31.034	29.768	4.782	79.053	43.962	59.909	53.251	24.88	3.519	17.076	72.675	38.023	6.728	55.252	8.694	74.905	2.009	76.191	69.662	23.603	3.164	45.469	58.568	0

C.3 OD-matrices for flow rate and flow mixture cut-off calibration

The following departure time periods are used[0,5400),[5400,12600) and [12600,14400), all denoted in seconds. The first period adopts the OD-matrix presented in Appendix B.2, the last period is an all zeros matrix to allow the network to empty while the second departure time period adopts the OD-matrix shown in Table B.3. This matrix has halved all cells of the inner 9 centroids in the grid (7-9, 12-14, 17-19) and doubled all others.

Table C.3: OD matrix second departure time period

Centroids	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0	155.57	153.86	155.44	96.49	153.23	7.199	28.569	21.793	15.282	40.456	32.751	24.666	29.853	7.028	31.2	38.262	7.351	27.602	136.29	137.6	66.934	30.654	103.66	144.97
2	67.364	0	144.03	78.906	18.638	114.96	0.716	32.157	8.9745	105.65	140.85	19.722	27.392	11.804	38.144	132.68	28.272	28.344	10.228	111.16	5.532	56.758	9.5	63.406	107.59
3	34.958	155.51	0	47.142	112.98	146.87	25.617	0.444	13.978	29.368	96.436	8.916	5.3295	33.975	87.07	128.81	14.933	18.303	37.684	106.01	129.19	35.05	141.32	58.532	37.914
4	139.18	25.21	149.35	0	156.23	115.05	30.047	9.2975	2.7075	95.346	53.08	18.963	18.711	7.157	7.97	45.23	16.385	10.315	24.349	70.778	41.034	87.468	7.35	99.61	133.59
5	158.76	116.02	94.178	92.96	0	27.434	3.757	6.066	5.523	108.07	5.228	34.282	35.221	39.344	118.11	55.316	39.007	8.474	36.088	105.75	13.158	99.07	43.662	126.18	86.986
6	86.544	39.994	102.53	114.21	109.91	0	27.245	1.2225	37.291	46.482	83.71	12.843	1.145	13.534	52.96	135.7	17.68	25.035	30.245	129.81	137.87	104.49	134.9	60.97	23.214
7	36.729	15.221	29.524	26.126	4.377	3.175	0	2.363	11.292	9.01	37.89	29.375	39.123	26	17.451	16.172	8.752	27.876	6.0875	29.776	12.094	33.142	22.398	26.329	23.282
8	35.915	30.461	22.654	26.362	0.3495	15.301	8.1985	0	39.526	33.065	28.404	36.065	11.587	38.721	13.902	32.745	7.2735	19.421	28.92	24.214	33.525	39.589	21.503	22.651	18.111
9	12.838	6.7585	18.572	22.435	4.434	31.597	31.276	36.934	0	7.4865	6.316	21.307	12.01	9.463	14.36	22.524	25.315	11.488	14.885	13.681	2.36	2.316	0.3595	12.882	22.547
10	16.066	82.508	72.622	139.75	41.512	42.474	13.049	10.05	22.705	0	154.43	4.603	5.1845	6.0745	75.214	104.85	19.376	10.567	27.951	51.904	104.4	122.11	57.534	150.85	106.6
11	98.338	9.84	56.822	101.45	105.09	34.524	35.026	37.355	20.211	153.34	0	36.901	38.863	38.261	56.556	60.484	14.746	38.249	26.22	48.798	46.838	99.364	14.836	24.652	131.6
12	0.002	28.778	30.66	37.9	28.945	11.483	20.054	34.427	19.686	15.119	10.597	0	13.825	9.642	5.8885	34.12	28.597	11.863	12.688	13.469	17.638	4.939	20.597	15.602	28.006
13	24.603	4.56	5.86	22.236	10.157	16.665	19.383	6.2465	25.225	25.141	5.476	1.6445	0	33.674	32.297	21.574	11.553	14.098	23.836	0.0455	6.663	4.5095	3.641	7.167	4.848
14	20.927	9.2645	24.838	13.448	29.752	6.7125	1.7815	0.8395	23.873	9.839	24.393	16.064	32.814	0	22.185	19.733	4.1135	9.4615	5.2655	6.569	35.691	35.438	22.303	30.239	9.4195
15	55.69	156.19	135.47	16.416	148.36	87.624	36.3	13.756	32.733	28.622	136.1	15.342	11.75	26.481	0	35.546	13.39	16.307	27.667	130.77	35.126	29.25	83.754	108.91	89.388
16	158.29	88.09	93.656	27.566	4.45	111.84	23.581	21.228	23.218	79.744	50.818	35.371	0.3435	28.867	0.476	0	28.267	17.66	25.26	60.366	113.45	23.022	103.25	132.21	37.632
17	30.304	10.692	8.5665	27.807	38.936	3.0005	21.55	26.762	6.909	28.22	31.803	28.964	32.736	16.792	27.973	5.5395	0	21.226	18.884	10.971	8.3575	15.653	6.426	15.469	32.484
18	18.387	24.8	37.621	13.178	13.363	15.111	31.888	5.9995	20.349	5.387	18.967	21.61	23.645	10.425	25.978	8.052	8.055	0	38.795	37.664	3.137	6.985	20.057	29.564	19.197
19	26.028	13.11	3.336	28.531	15.994	17.534	15.545	4.1295	31.864	8.223	38.577	28.615	23.939	4.162	22.14	28.324	32.848	22.525	0	12.92	36.254	26.798	1.744	36.868	23.332
20	1.568	43.74	153.3	108.72	105.71	38.936	5.7085	21.637	32.207	61.424	42.322	17.83	10.779	27.198	101.01	138.6	28.534	21.47	39.718	0	76.752	14.222	10.862	10.53	8.878
21	125.52	84.018	84.412	3.792	32.034	86.77	21.12	10.577	5.738	60.18	80.398	8.7235	36.732	7.322	8.776	123.29	13.644	30.319	25.612	31.278	0	101.33	63.394	73.66	124.78
22	130.41	91.632	108.25	142.81	81.91	135.16	15.671	16.731	37.264	76.496	61.428	23.76	0.97	0.984	6.736	126.39	12.968	29.152	15.541	15.676	74.19	0	58.718	102.27	103.28
23	124.8	61.294	64.062	15.338	54.75	95.08	35.254	10.741	34.548	40.264	119.53	27.018	20.556	36.778	100.4	9.858	16.116	23.14	30.426	95.046	88.088	36.888	0	3.296	32.734
24	10.808	41.02	82.21	137.5	75.118	121.08	7.069	15.016	4.2025	59.834	123.58	30.398	9.832	26.255	141.93	66.014	8.689	7.4225	4.757	57.302	30.004	83.478	40.57	0	146.54
25	129.2	62.068	59.536	9.564	158.11	87.924	29.955	26.626	12.44	7.038	34.152	36.338	19.012	3.364	110.5	17.388	37.453	1.0045	38.096	139.32	47.206	6.328	90.938	117.14	0