



WORKING PAPER

ITLS-WP-14-03

**Capacity constrained stochastic
static traffic assignment with
residual point queues
incorporating a proper node
model**

By

**Michiel C.J. Bliemer, Mark P.H. Raadsen,
Erik-Sander Smits¹, Bojian Zhou² and
Michael G.H. Bell**

¹ Department of Transport and Planning, Delft University
of Technology, The Netherlands

² School of Transportation, Southeast University, Nanjing,
China

March 2014

**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-14-03

TITLE: **Capacity constrained stochastic static traffic assignment with residual point queues incorporating a proper node model**

ABSTRACT: Static traffic assignment models are still widely applied for strategic transport planning purposes in spite of the fact that such models produce implausible traffic flows that exceed link capacities and predict incorrect congestion locations. There have been numerous attempts in the literature to add capacity constraints to obtain more realistic traffic flows and bottleneck locations, but so far there has not been a satisfactory model formulation. After reviewing the literature, we come to the conclusion that an important piece of the puzzle has been missing so far, namely the inclusion of a proper node model. In this paper we propose a novel path-based static traffic assignment model for finding a stochastic user equilibrium in which we include a first order node model that yields realistic turn capacities, which are then used to determine consistent traffic flows and residual point queues. The route choice part of the model is specified as a variational inequality problem, while the network loading part is formulated as a fixed point problem. Both problems are solved using existing techniques. We illustrate the model using hypothetical examples, and also demonstrate feasibility on large-scale networks.

KEY WORDS: *Static traffic assignment, stochastic user equilibrium, capacity constrained, residual queues, node model*

AUTHORS: **Bliemer, Raadsen, Smits, Zhou and Bell**

Acknowledgements: We would like to thank Luuk Brederode and Luc Wismans for their comments on the draft version of the paper, and Omnitrans International for providing the Streamline framework. Furthermore, we would like to thank Wenlong Jin for the useful discussions on the node model. The paper solely expresses the opinion of the authors. This research is partly sponsored by funding from the Australian Research Council (LP130101048).

CONTACT: INSTITUTE OF TRANSPORT AND LOGISTICS STUDIES (C13)
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 2014

1. Introduction

Traffic assignment models are widely used all over the world in transport planning to assist in making important decisions with respect to infrastructure investments and project appraisal. Even though the (academic) research in traffic assignment is shifting towards dynamic traffic assignment, static traffic assignment is still by far the most widely used tool in strategic transport planning. While we recognize that dynamic traffic assignment models are able to produce much more realistic traffic flows and travel times than static counterparts, they have a much higher computational complexity, require more sophisticated calibration techniques, and are less scalable. Therefore, we expect that transport planners and policy makers will rely on static traffic assignment models for some time because of their ease of use, comprehensibility, computational convenience, and mathematical rigor.

As is well-known, traditional static models make several strong assumptions, resulting in traffic flow predictions that are infeasible (flows exceeding the capacity of the road), congestion in the wrong locations, and travel times that are implausible. There have been several attempts to extend static models to include capacity constraints, however none of these proposed models produce realistic traffic flows, queues, and travel times.

In this paper, we propose a novel capacity constrained static traffic assignment model that aims to describe traffic in a much more realistic fashion while maintaining a rigorous analytical model and computational tractability. We demonstrate that our model predicts queues in the correct locations, which enables more accurate travel time calculations. We achieve this by not relying on link performance functions, but rather adopt elements of first order (steady state) dynamic models. In particular, we adopt a proper node model, which is an important element that is missing in all previous studies. Inclusion of such a node model is key in establishing both realistic and consistent turn capacities and flows.

The main contributions of this paper can be stated as follows. First, we provide an extensive literature review of earlier proposed static traffic assignment models that aim to include capacity constraints and queues, and show that none of them yield realistic traffic flows and queue locations. Secondly, we formulate a novel stochastic capacity constrained static traffic assignment model with residual point queues that is able to generate more realistic flows, queue locations, and travel times. Thirdly, we propose a relatively simple solution procedure without the need to rely on complex algorithms, in contrast to most previous studies. Finally, we illustrate the model with several hypothetical examples and demonstrate the feasibility on large scale networks using real life case studies. We believe that our novel model formulation is a major step towards more realistic steady-state network modeling, consistent with important traffic flow principles.

The outline of the paper is as follows. In Section 2 we provide an overview of the traditional static traffic assignment model, and we classify several extensions proposed in the literature to include capacity constraints. In Section 3 we formulate our new model, in which we focus on a capacity constrained network loading model that is consistent with a proper first order node model. Section 4 presents an algorithm for solving our new static traffic assignment model, which includes an algorithm for route choice, and an algorithm for network loading. In Section 5, the model and algorithms are illustrated using hypothetical network examples, but we also demonstrate feasibility on large real life networks. In Section 6 we summarize our main findings and provide a brief discussion on related topics.

2. Static traffic assignment models in the literature

2.1 Traditional unconstrained models

In this section, we briefly describe the traditional unconstrained version of the static traffic assignment, and introduce notation that we will use in the remainder of the paper.

Consider a transport network $G = (N, A)$ with a set of nodes N and a set of directed links, A . Let $R \subseteq N$ denote the set of origins, and $S \subseteq N$ the set of destinations. The average travel demand over time period $[0, T]$ between origin-destination (OD) pair (r, s) , with $r \in R$ and $s \in S$, is assumed to be given by D_{rs} . We will assume in the remainder of the paper that all demand, flows, and capacities are rates, i.e. vehicles per hour. Let P_{rs} be the (non-empty) set of available paths from origin r to destination s , and let $P = \bigcup_{(r,s)} P_{rs}$ be the set of all routes between all OD pairs.

Wardrop (1952) defined a user equilibrium in traffic assignment as the situation in which no traveler can be better off (i.e., obtain lower travel costs, taking possible congestion delays into account) by unilaterally switching routes. Such traffic equilibria can be found by solving variational inequality problems (Dafermos, 1980). In this paper, we will adopt the more general case of a stochastic user equilibrium (SUE), in which travellers are assumed to take the route with the lowest *perceived* costs (Daganzo and Sheffi, 1977). Fisk (1980) has considered a special case of this stochastic user equilibrium, in which the route probabilities are given by a conditional logit model (McFadden, 1974), such that flow on route p is given by:

$$f_p = \frac{\exp(-\theta c_p)}{\sum_{p' \in P_{rs}} \exp(-\theta c_{p'})} D_{rs}, \quad \forall p \in P_{rs}, \forall (r, s), \quad (1)$$

where c_p is the cost of route p , and θ is a scale parameter (i.e., if $\theta = 0$ then route choice is completely random, while if $\theta \rightarrow \infty$ then route choice is completely deterministic). Let $\mathbf{f} \equiv [f_p]_{p \in P}$ denote the vector of all route flows. In case of a logit based assignment, \mathbf{f}^* is the set of stochastic user equilibrium flows if it satisfies the following variational inequality (VI) problem (see e.g., Guo et al., 2010; Luo et al., 2012):

$$\sum_{(r,s)} \sum_{p \in P_{rs}} (\theta c_p^* + \ln(f_p^*)) (f_p - f_p^*) \geq 0, \quad \forall \mathbf{f} \in \Omega, \quad (2)$$

where c_p^* is the equilibrium route cost (which depends on equilibrium route flows \mathbf{f}^*), and Ω is the set of feasible route flows defined by the following two constraints:

$$\sum_{p \in P_{rs}} f_p = D_{rs}, \quad \forall (r, s), \quad (3)$$

$$f_p \geq 0, \quad \forall p \in P_{rs}, \forall (r, s). \quad (4)$$

Constraint (3) ensures that all travel demand is assigned to a route, and constraint (4) ensures nonnegative flows. It is well-known that the original conditional logit model in Equation (1) does not properly deal with route overlap, hence adaptations to the model by adding a commonality factor (C-logit, see Cascetta et al., 1996) or a path-size factor (Ben-Akiva and Bierlaire, 1999) have been proposed. The VI problem formulation in Equation (2) can easily be

Capacity constrained stochastic static traffic assignment with residual point queues incorporating a proper node model

Bliemer, Raadsen, Smits, Zhou and Bell

adjusted accordingly, as shown in Zhou et al. (2012). In this paper we will consider the simple logit model in Equation (1), as the focus of this paper is on the network loading part, as described below.

In order to compute the route travel costs c_p , in general the following relationships are used:

$$c_p = \sum_{a \in A} \delta_{ap} c_a, \quad \forall p \in P_{rs}, \forall (r, s), \quad (5)$$

$$c_a = \tau_a(\mathbf{q}), \quad \forall a \in A, \quad (6)$$

In Equation (5), additive costs are assumed, such that the route travel costs c_p can be expressed as a summation of link costs c_a using link-route incidence indicator δ_{ap} (which equals one if link a is on route p , and zero otherwise). However, note that the route based problem formulation in inequality (2) can handle non-additive costs. In Equation (6), the link costs are determined by an explicit or implicit function $\tau_a(\cdot)$ of the link (in)flows $\mathbf{q} = [q_a]_{a \in A}$.

Finally, the following common relationship can be used to provides the link flows necessary to calculate the link costs,

$$q_a = \sum_{(r,s)} \sum_{p \in P_{rs}} \delta_{ap} f_p, \quad \forall a \in A. \quad (7)$$

It is important to note that this equation essentially describes a static network loading procedure. In this paper we will mainly focus on replacing this classic network loading model with a much more realistic one, due to its increased level of consistency with respect to dynamic network loading models.

Under the conditions (i) $\theta \rightarrow \infty$ and (ii) separable link travel cost functions, i.e. $c_a = \tau_a(q_a)$ in which the link cost only depends on the flow in the link itself, VI problem (2) can be written as the well-known mathematical programming problem (Beckmann et al., 1956):

$$\min_{\mathbf{q}} \sum_a \int_0^{q_a} \tau_a(w) dw, \quad (8)$$

subject to constraints (3), (4) and (7).

In traditional static traffic assignment models, explicit link performance functions $\tau_a(\cdot)$ are formulated, such as the widely used BPR (Bureau of Public Roads, 1964) function,

$$\tau_a(q_a) = \frac{L_a}{v_a^{\max}} \left(1 + \gamma_a \left(\frac{q_a}{C_a} \right)^{\kappa_a} \right), \quad \forall a \in A, \quad (9)$$

where for each link a , L_a is the length (km), v_a^{\max} is the maximum speed (km/h), C_a is the capacity (veh/h), and γ_a and κ_a are parameters that depend on the road type.

It is important to note that link flows can exceed the link capacities in this unconstrained model formulation. Since this is physically not possible, Daganzo (1977a,b) proposed to use a link performance function $\tau_a(q_a)$ with an asymptote near capacity, which aims to prevent the link flow from exceeding the capacity. The Frank-Wolfe algorithm (Frank and Wolfe, 1956) was adopted to solve the problem. Application of asymptotical link performance functions will

produce very high travel times in saturated links (Boyce et al., 1981), and therefore does not improve the realism of the solution. Alternative models were proposed to include hard capacity constraints, as described in the next section.

2.2 *Capacity constrained models*

The traditional formulation of the static traffic assignment problem does not take any explicit link capacity constraints into account. All path flows are assumed to be able to pass through each link, such that the link flows can be determined by a simple mapping from the path flows, given by Equation (7). In order to take into account that each link a has a limited capacity, the following straightforward constraints have to be added,

$$q_a \leq C_a, \quad \forall a. \quad (10)$$

This results in what is called a *capacity constrained* formulation. Although adding the capacity constraints seems natural, it is not consistent with the link travel time functions $\tau_a(q_a)$, such that ‘tricks’ with Lagrange multipliers or penalty functions are needed. Although adding these constraints to the problem formulation is easy, conventional solution methods cannot be used anymore, making this problem much harder to solve. Furthermore, a feasible solution may no longer exist.

In this type of models, the Lagrange multipliers of the capacity constraints are interpreted as the optimal toll charge (Jorgensen, 1963) or steady-state link queues (Payne and Thompson, 1975, Smith, 1987). Solution methods of this type of model algorithms can be divided into exterior penalty function methods (Hearn, 1980), inner penalty function methods (Inouye, 1987; Nie et al., 2004; Prashker and Toledo, 2004; Shahpar et al., 2008), and Lagrange multiplier methods (Hearn and Ribera, 1980; Larsson and Patriksson, 1995; Yang and Yagar, 1994, 1995). Bell (1995) built a linearly constrained convex minimization model for the capacity constrained logit SUE problem, proved that the Lagrange multipliers of his model give the equilibrium delays in the network, and proposed an iterative balancing method to solve the problem. Meng et al. (2008) studied the general SUE problem with capacity constraints, and developed a Lagrangian dual method for the problem.

Essentially all studies mentioned above have approached the problem from an operations research perspective in which the capacity constraints were seen as merely extra side constraints to the problem. This results in assignment outcomes in which no queues are formed, but rather flows are re-routed to other parts of the network that have sufficient capacity. If insufficient capacity is available in the network, a solution does not exist. Such outcomes are again unrealistic; as queues do build up, traffic flows can and will not always be rerouted and situations can occur where demand actually exceeds capacity. A somewhat different approach has been followed by Marcotte et al. (2004), who extend the hyperpath assignment concept to model traffic assignment in networks with rigid finite capacities. In their model, users are assigned to strategies which provide at every node, a set of sub-paths and the order of preference. The rigid finite capacity assumption is assumed to be a proxy for travel delay and the travel cost function is not given in closed form. The model was formulated as a variational inequality and various theoretical properties of the model were analyzed. A partial linearization method and a projection method were proposed to solve the model.

Instead of an operations research perspective, others have looked at the problem using traffic engineering principles, and adopt the idea of residual queues, as described in the next section.

2.3 *Residual queuing models*

A next step forward was made by models that consider what the literature has called *residual queues*, which is blocked flow that is unable to proceed. In the traditional static models and capacity constrained static models is assumed that link inflow equals link outflow (hence, only

'link flow' was defined). In residual queuing models, more vehicles can flow into the link than may exit (during a certain period of time), such that a residual queue appears on the link.

As before, each link is assumed to have a certain capacity (typically depending on the number of lanes, the maximum speed, and the road type), but in residual queuing models often so-called link-exit capacities are used, while there are no specific link entrance capacities (and therefore usually all flow is assumed to be able to enter a link). Different models determine these exit capacities differently. Mostly it is assumed to be equal to the capacity of the link itself, but sometimes it is a function of the next link(s). None of the proposed models apply a proper node model to determine outflow capacity.

Bifulco and Crisalli (1998) were among the first to move away from interpreting Lagrange multipliers as queuing delays, which they note are not necessarily consistent with what could be observed at oversaturated links. They describe a stochastic user equilibrium problem in which the network loading is formulated as a fixed point problem. No proper node model is used in their model formulation. Lam and Zhang (2000) also investigated the traffic assignment problem with constraints on the link-exit capacity by taking residual queues into account. In their model, the Lagrange multipliers of the capacity constraints are interpreted as the residual queue volumes, and the queuing delay is calculated using the equation given by Akçelik and Roupail (1993). The model is formulated as a mathematical programming problem. The Frank-Wolfe algorithm that includes a cutoff path operator is developed to solve this model. Nesterov and De Palma (1998, 2000, 2003) assume lower bounds on the travel time (free-flow travel time) and upper bounds on the flow (link capacity) instead of using link travel time functions. They search for stable traffic equilibria in which the fundamental relationship between flow, speed, and density holds. Queues longer than the link length are assumed not to occur; hence spillback is again not taken into account. A detailed comparison of the model of Nesterov and De Palma (1998) and Beckmann et al. (1956) model is given by Chudak et al. (2007). Smith (2013) proposed a link-based elastic demand equilibrium model with explicit link-exit capacities and explicit queuing delays. They state that their model is between steady state and dynamic equilibrium transport models. In this model, link flow is constrained by the maximum exit flow. Smith proved that, under very weak conditions, a solution exists, and suggested using the alternating direction method of Lv et al. (2007) to solve the model. Blocking back or spillback is not considered in this model. Smith et al. (2013) describes an alternative approach with explicit link-exit capacities with a spatial queuing model considering no blocking back as well as blocking back. They show that if capacity restrictions are handled through prices and the queue-storage capacities are large enough, then an equilibrium solution exists.

Although basically all previously described models have been formulated in a rigorous mathematical optimisation problem, there also exist models that are merely described by an operational procedure. This offers more flexibility in making the outcomes more realistic, however, prevents analysing properties of the model and the solution. Bundschuh et al. (2006) developed an operational steady state model that they term quasi-dynamic, as it takes capacity constraints and spillback into account. This model has been implemented in VISUM. They use a kind of incremental assignment (temporarily keeping route choice fixed), in which iteratively a fraction of the travel demand is put on the network. The flow is propagated over the consecutive links of a path until the exit capacity of a link is reached. The extra flow on that link will be stored in a residual queue in the bottleneck link, and blocked back to upstream links if the queue exceeds the storage capacity of the link. Bakker et al. (1994) and 4Cast (2009) developed an operational model called QBLOK, which they also termed quasi-dynamic. This operational model is used in the Dutch national and regional models. It describes a heuristic procedure that ensures that link capacities are not exceeded. Residual queues start, similar to the model of Bundschuh et al. (2006), in the bottleneck link, and not upstream this link. Queues longer than the link length can occur, such that spillback can be taken into account in this model.

While models with residual queues generate more realistic traffic flow than models that simply add capacity constraints as described in the previous section, the bottleneck locations are still predicted in the wrong locations. This is mainly due to the fact that link-exit capacities are considered, which places residual queues inside the bottleneck instead of upstream the bottleneck. Further, none of the models consider a proper node model that determines turn capacities based on link capacities and travel demand. These turn capacities determine realistic link-exit capacities, possibly lower than the link capacity.

2.4 Comparison of different model types

In this section we will numerically show the differences in traffic flows and queues of the described static model types. Furthermore, we illustrate the major improvement in realism of the modeling approach proposed in this paper.

Consider the simple corridor network in Figure 1, in which link capacities and free-flow travel times are given. Further, a travel demand of 4000 veh/h is assumed, and we assume a time horizon of one hour. Table 1 shows the static traffic assignment outcomes for the different model types, namely the link (in)flows q_a and the number of vehicles in the queue after one hour, Q_a . For simplicity we assume that the link lengths are sufficiently long, such that we can ignore spillback effects.

For the traditional unconstrained model, all link flows are 4000 veh/h and the travel times are highest in links 3 and 5 (according to the BPR function, as they have the highest volume/capacity ratio). No residual queues are predicted.

Capacity constrained models would typically predict a route flow of 2000 veh/h (according to the most restricted capacities) through this corridor. However, since there is no alternative route available to direct the remaining 2000 veh/h, the capacity constrained models are unable to find a solution. These models are restricted in the sense that they assume that all flow goes through the network and do not assume any residual queues.

Models with residual queues consider link-exit capacities and place the remaining flow on the link in a residual queue. For example, 4000 vehicles entered link 2 but only 3000 vehicles exit this link after one hour, such that a residual queue of 1000 vehicles remains in link 2. Similarly, 3000 vehicles enter link 3 but only 2000 vehicle exit, yielding a residual queue of 1000 vehicles in link 3. After one hour, only 2000 vehicles flow out of links 4 and 5. High travel times are therefore predicted on links 2 and 3.

What actually happens in real life (i.e., traffic flows that would be predicted if loop detectors would measure the inflow of each link) is that 4000 vehicles enter link 1 but only 3000 vehicles would enter link 2, forming a residual queue of 1000 vehicles in link 1. From the 3000 vehicles flowing into link 2, only 2000 vehicles exit this link, resulting in a residual queue of 1000 vehicles in link 2. The other links will not have any queues. Therefore, queues and high travel times should only occur on links 1 and 2.

Note that the flows and queues predicted by the different models all differ significantly from what actually would happen. This has significant consequences for infrastructure decisions based on these forecasts. For example, based on outcomes of the traditional unconstrained model, one could draw the conclusion that increasing the capacity of link 5 would significantly reduce the travel time on this corridor, while in real life such an extension would have no effect on the traffic flows or the travel times.

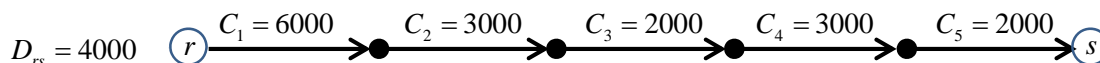


Figure 1: Simple corridor network

Table 1: Outcomes of different static traffic assignment models for simple corridor network

	Link 1		Link 2		Link 3		Link 4		Link 5	
	q_1	Q_1	q_2	Q_2	q_3	Q_3	q_4	Q_4	q_5	Q_5
Trad. unconstrained	4000	0	4000	0	4000	0	4000	0	4000	0
Capacity constrained	--	--	--	--	--	--	--	--	--	--
Residual queuing	4000	0	4000	1000	3000	1000	2000	0	2000	0
Actual	4000	1000	3000	1000	2000	0	2000	0	2000	0

Now consider a second network shown in Figure 2, which has been analyzed in Lam and Zhang (2000), in which they aim to illustrate that their model with residual queues provides accurate flows and queues. Link capacities and free-flow travel times are listed in the figure. A travel demand of 1200 veh/h is assumed and we again consider a time horizon of one hour. Table 2 presents the predicted link (in)flows and queues, in which we assume a deterministic user equilibrium. The delays in residual queues equal the number of vehicles in the queue divided by the service rate (i.e., the capacity). For the traditional unconstrained model, a BPR function with $\gamma_a = 0.15$ and $\kappa_a = 4$ is assumed for each link.

In the unconstrained model, the route flow is distributed between the routes on the top and bottom, where the flow on link 3 exceeds the capacity. The capacity constrained model would only allow 600 vehicles through the bottom route, and direct the remaining flow to the top route. In both cases, all 1200 vehicles would leave the network after one hour. The model with residual queues (i.e., the model proposed by Lam and Zhang, 2000) puts the queue inside the bottleneck (link 3) and forecast 1140 vehicles to leave the network after one hour. Note that the actual flows would again be completely different than predicted by any of these models. Namely, a queue would form at the end of link 1, and all travellers (independent on which route they take) have to wait in the queue.¹ When having reached the end of the queue, there is no reason for travellers to choose the top route, as it is longer than the bottom route. Hence, all travellers would choose the bottom route. This means that only 600 vehicles exit link 1 and exit the network after one hour.

We have adopted the deterministic user equilibrium example for illustration purposes, but clearly the example can be easily extended to a stochastic user equilibrium. In that case, there will be some flow on the top route, although the queue in our model will still be predicted on link 1 (in contrast to existing models).

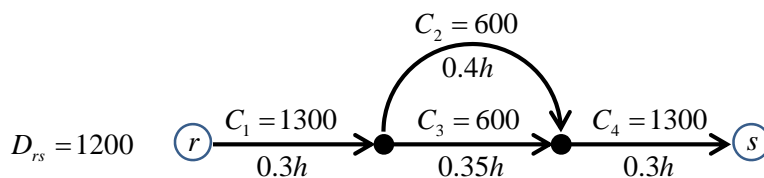


Figure 2: Simple two-route network (source: Lam and Zhang, 2000)

¹ Here we have assumed that there is a first-in-first-out (FIFO) queue, hence no separate queues for each turn. This so-called FIFO diverging rule is further discussed in Section 3.3. In case there are (long) separate turning lanes, one could adjust the network such that each turning lane is represented by a separate link.

Table 2: Outcomes of different static traffic assignment models for simple two-route network

	Link 1		Link 2		Link 3		Link 4	
	q_1	Q_1	q_2	Q_2	q_3	Q_3	q_4	Q_4
Traditional unconstrained	1200	0	524	0	676	0	1200	0
Capacity constrained	1200	0	600	0	600	0	1200	0
Residual queuing	1200	0	540	0	600	60	1140	0
Actual	1200	600	0	0	600	0	600	0

With these two simple examples we show that none of the current static traffic assignment models give satisfactory results. The model proposed in this paper predicts the actual flows and queues given in Tables 1 and 2 and would therefore constitute a major improvement. The model can be defined by adding constraints to the traditional static traffic assignment model, leading to a rigorous, elegant, and generic problem formulation, which can be solved efficiently on large networks.

3. Newly proposed model

The model we propose in this paper is based on the same SUE problem formulation as described by VI problem (2), but applies a different network loading procedure. To be more precise, we will use relationships (5) and (6), with a specific formulation of the link performance function $\tau_a(\mathbf{q})$, but replace Equation (7) with a capacity constrained mapping from route flows to link flows consistent with a first order node model. More precisely, we will replace Equation (7) with a fixed point problem that ensures consistency between link flows and link capacities. It is important to note that we adopt a route based formulation.

In our approach in which we adopt a novel static network loading model, we assume all residual queues are point queues, which do not have a physical length. This is a common assumption made in static models with residual queues. Although realistically representing spillback in static models remains a challenge, we believe our proposed model is a major step towards more realistic outcomes in static traffic assignment. Bliemer et al. (2012) proposed a post processing stage, based on an event-based dynamic link transmission model, that converts the vertical queues into physical queues, which can take spillback into account. It is beyond the scope of this paper to further discuss such a post processing stage.

In the next section we describe our new network loading procedure formulated as a fixed point problem, which can in principle take any node model into account. Then in Section 3.2 we discuss how to calculate the link costs, and in Section 3.3 we will explain our choice of node model adopted in this paper.

3.1 Capacity constrained network loading with residual point queues

The network loading procedure takes the route flows (demand) as input, and determines the link flows (and travel costs) as output. Our capacity constrained network loading procedure consists of two main building blocks, namely (i) a node model that computes reduction factors for each turn, and (ii) an instantaneous network loading procedure that applies these reduction factors to the route flows.

Let A_n^{in} denote the set of links flowing into node n , and let A_n^{out} be the set of links flowing out of node n . For each node $n \in N$, define reduction factors per turn direction, $\alpha_{ab} \in [0,1]$, where

$a \in A_n^{\text{in}}$ and $b \in A_n^{\text{out}}$, as the ratio of the actual turn outflow (limited by capacity constraints) and the desired turn outflow (i.e., local demand). If $\alpha_{ab} = 1$, then the flow is not capacity constrained and all desired flow can flow out. Let q_{ap} be the flow into link a following route p , and let η_{ap} be the set of turns on path p from the origin to link a . As stated in Bifulco and Crisalli (1998), the path-specific link inflow is given by the route flow multiplied by previous reduction factors on the route:

$$q_{ap} = \delta_{ap} f_p \prod_{(a',b') \in \eta_{ap}} \alpha_{a'b'}, \quad \forall a \in A, \forall p \in P_{rs}, \forall (r,s), \quad (11)$$

where δ_{ap} is the same link-route incidence indicator as defined earlier. For example, consider the corridor network in Figure 1. The flow into link 1 is 4000, however, only 3000 can flow out. A realistic node model will therefore compute a reduction factor of $\alpha_{12} = 0.75$. According to Equation (11), the flow into link 2 is then 3000, while the outflow is constrained by the capacity of link 3 to 2000, hence $\alpha_{23} = 0.67$. All remaining reduction factors are equal to 1. This yields an inflow into link 5 of $4000 \cdot 0.75 \cdot 0.67 \cdot 1 \cdot 1 = 2000$.

In general, the reduction factors are obtained from a node model that takes the desired turn flows (demand) and the given link capacities (supply) into account. The turn flows from link a to link b can be determined from the path-specific link flows by the following definition:

$$t_{ab} = \sum_{(r,s)} \sum_{p \in P_{rs}} \delta_{bp} q_{ap}, \quad \forall a \in A_n^{\text{in}}, \forall b \in A_n^{\text{out}}, \forall n \in N. \quad (12)$$

Further, by definition also holds that the link flow is the sum over all relevant path-specific link flows,

$$q_a = \sum_{(r,s)} \sum_{p \in P_{rs}} q_{ap}, \quad \forall a \in A. \quad (13)$$

In order to keep the model general, we consider a general function $\Gamma^n(\cdot)$ for each node n that computes the reduction factors per turn based on a given node model. In Section 3.3 we will discuss the chosen node model. For each node n , the set of reduction factors for all turn movements is given by the following implicit function:

$$[\alpha_{ab}]_{a \in A_n^{\text{in}}, b \in A_n^{\text{out}}} = \Gamma^n(t_{a'b'}, C_{a'}, C_{b'}, \forall a' \in A_n^{\text{in}}, \forall b' \in A_n^{\text{out}}), \quad \forall n \in N. \quad (14)$$

In other words, the reduction factors per turn are a function of the desired link turn flows (demand) and the capacities of all incoming and outgoing links (supply).

It should be pointed out that the traditional unconstrained model and the residual queuing models are special cases of this model formulation by adopting alternative formulations of the function $\Gamma^n(\cdot)$. In case of the traditional model, which does not have capacity constraints, this function always returns the value one, i.e. $\alpha_{ab} = 1$ for all turns. In case of previously proposed residual queuing models, this function is typically a simple function of the link inflow and capacities of the incoming links (i.e., $\alpha_{ab} = C_a / q_a$) for all turn directions b . In general, these reduction factors are not separable, meaning that they depend not only on the flow on the current link, but also on the flows of other links competing for the same capacity. This non-separability of the reduction factors makes the problem more challenging to solve.

We can observe from Equations (11) and (12) that the turn flows depend on the reduction factors, while the reduction factors according to Equation (14) depend on the turn flows. Let $\mathbf{t}=[t_{ab}]$ and $\boldsymbol{\alpha}=[\alpha_{ab}]$ denote the vectors of turn flows and reduction factors, respectively. We can write the flow propagation, as specified by Equations (11)-(13), in the compact form of $\mathbf{t} = \Upsilon(\boldsymbol{\alpha} | \mathbf{f})$, where \mathbf{f} is the vector of given route demand flows. Further, we can write the node model, as specified in Equation (14), in the compact form of $\boldsymbol{\alpha} = \Gamma(\mathbf{t} | \mathbf{C})$, where \mathbf{C} is the vector of given link capacities. This yields the following fixed point problem:

$$\mathbf{t} = \Upsilon(\Gamma(\mathbf{t} | \mathbf{C}) | \mathbf{f}). \quad (15)$$

The vector of turn flows \mathbf{t}^* that satisfies $\mathbf{t}^* = g(\mathbf{t}^* | \mathbf{f}, \mathbf{C})$, where $g = \Upsilon \circ \Gamma$ is the composite function, is a solution to the fixed point problem. This also yields reduction factors $\boldsymbol{\alpha}^*$, such that the link flows can be computed by Equations (11) and (13). This problem formulation can be viewed as a generalization of the model presented in Bifulco and Crisalli (1998). In Section 4 we will propose an algorithm for solving this fixed point problem.

3.2 Link travel times

Now let us derive the travel time function $\tau_a(\mathbf{q})$, which we assume consists of the free-flow travel time and the average queuing delay over time period $[0, T]$. Although our model is general enough to specify different queues per turn, most node models adopt the so-called first-in-first-out (FIFO) diverging rule (Daganzo, 1995). This results in equal reduction factors for a given incoming link to all directions, i.e., $\alpha_{ab} = \alpha_a$ for all turn directions b . In this paper, we will adopt the same rule, such that a single queue will form for at each link-exit (see also footnote 1 and Section 3.3). In the remainder of the paper we refer to link reduction factors α_a , noting that they apply to all turn directions. The (vertical) queues at time instant 0 equal zero, while the queues for each link at time instant T can be computed as

$$Q_a = (1 - \alpha_a)q_a T. \quad (16)$$

Hence, the average queue length at the end of the link is $\frac{1}{2}Q_a$. The outflow (service) rate (veh/h) for each link is given by $\alpha_a q_a$. Hence, the average queuing delay is equal to the average queue length divided by the outflow rate, $\frac{1}{2}Q_a / (\alpha_a q_a) = \frac{1}{2}(1 - \alpha_a)T / \alpha_a$. Since we have assumed a point queue that does not have a physical length, we can compute the travel time by summing the free-flow travel time, $\tau_a^{\text{ff}}(q_a)$, and the average queuing delay, i.e.,

$$\tau_a(\mathbf{q}) = \tau_a^{\text{ff}}(q_a) + \frac{1 - \alpha_a}{2\alpha_a} T. \quad (17)$$

Clearly, if the flow is not capacity constrained, $\alpha_a = 1$, which results in $\tau_a = \tau_a^{\text{ff}}(q_a)$ (free-flow travel time). It is further important to note that the travel time depends on the length of the considered time period, T , in contrast to existing static models. This makes sense, as there will be twice as many vehicles in the queue if the time period is twice as long, because queues are not stationary. This is a subtle but important difference with existing models.

The free-flow travel time $\tau_a^{\text{ff}}(q_a)$ is a function of the link (in)flow q_a . One could use the BPR function as specified in Equation (9), noting that in our model the link flow will never exceed capacity and therefore the link travel time will be between L_a / v_a^{max} and L_a / v_a^{crit} , where $v_a^{\text{crit}} = v_a^{\text{max}}(1 + \gamma_a)$ is the critical speed at capacity. In this paper, we propose to use the free-

flowing part of what we call the Quadratic-Linear (QL) fundamental diagram in which the flow in the free-flowing part is quadratic with respect to density, k_a , and linear in the congested part (see Figure 3(a)):

$$q_a(k_a) = \begin{cases} v_a^{\max} \left(1 - \frac{1}{k_a^{\text{crit}}} \left(1 - \frac{v_a^{\text{crit}}}{v_a^{\max}} \right) k_a \right) k_a, & \text{if } 0 \leq k_a \leq k_a^{\text{crit}}, \\ C_a \left(1 - \frac{k_a - k_a^{\text{crit}}}{k_a^{\max} - k_a^{\text{crit}}} \right), & \text{if } k_a^{\text{crit}} < k_a \leq k_a^{\max}, \end{cases} \quad (18)$$

where the jam density is defined as k_a^{\max} , while the critical density is defined by $k_a^{\text{crit}} = C_a / v_a^{\text{crit}}$. If $v_a^{\text{crit}} = v_a^{\max}$, the QL fundamental diagram simplifies to the well-known triangular fundamental diagram (Newell, 1993). In that case, the free-flow travel time would be flow independent, namely $\tau_a^{\text{ff}} = L_a / v_a^{\max}$. We would argue that assuming that the critical speed is the same as the maximum speed is not realistic (e.g., on a 120km/h motorway, the critical speed at capacity is around 80km/h), and even though the QL diagram looks similar to the triangular diagram, it would generate very different and significantly more realistic travel times in the free-flowing part as shown in Figure 3(b). The corresponding travel time function derived from the QL fundamental diagram is

$$\tau_a^{\text{ff}}(q_a) = \frac{2L_a}{v_a^{\max} + \sqrt{v_a^{\max} \left(v_a^{\max} - \frac{4q_a}{k_a^{\max}} \left(1 - \frac{v_a^{\text{crit}}}{v_a^{\max}} \right) \right)}}. \quad (19)$$

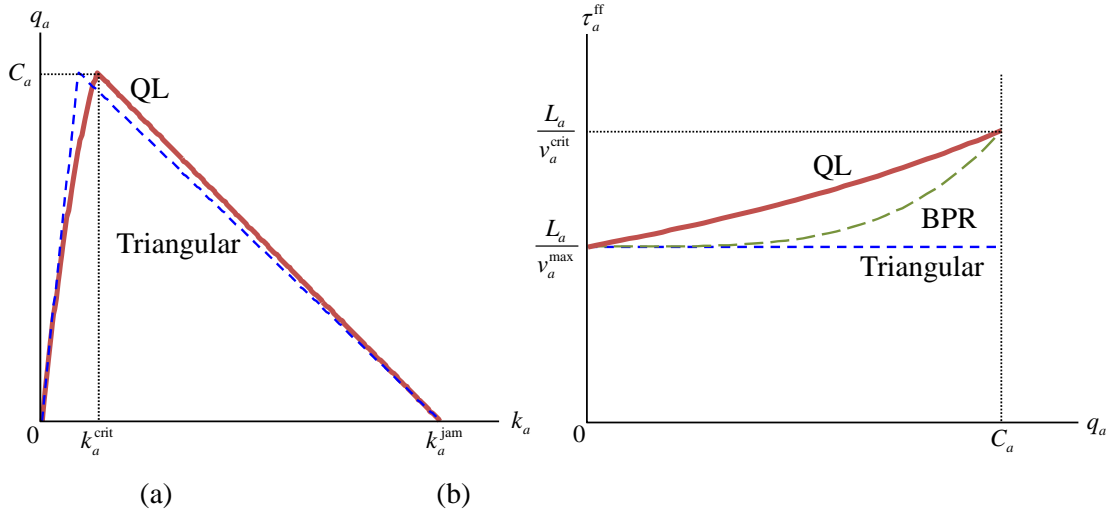


Figure 3: The QL fundamental diagram (a) and related travel time function (b)

3.3 Adopted generic first order node model

Several macroscopic stationary (first order) node models have been proposed in the literature. Tampère et al. (2011) provided a list of requirements that are necessary for node models to produce consistent and realistic results, namely (i) general applicability (i.e., can handle cross-nodes with any number of incoming and outgoing links), (ii) maximizing flows, (iii) non-negativity, (iv) conservation of vehicles, (v) satisfying demand and supply constraints, (vi) obeying conservation of turning fractions (CTF), and (vii) satisfaction of the invariance principle (Lebacque and Khoshyaran, 2005).

Daganzo (1995) proposed models for merges and diverges, but not general cross-nodes. Diverge nodes follow simple first-in-first-out (FIFO) diverging rules (Daganzo, 1995), while merge nodes are somewhat more complex and require a parameter for the merging priority. Jin and Zhang (2003) introduced the fair merging rule, which can be defined as a capacity proportional distribution for congested upstream links. Jin and Zhang (2004) proposed a model for general cross-nodes, however, the model does not satisfy the CTF condition. Bliemer (2007) provided a direct extension of the merge and diverge node models of Daganzo (1995) to general cross-nodes, however, the model does not satisfy the invariance principle. Jin (2012a) investigated stationary states for diverge and merge networks, but no cross-node formulation is given. Jin (2012b) proposes a closed form invariant model for general cross-nodes, although it does not satisfy all requirements outlined above.

Tampère et al. (2011) and Flötteröd and Rohde (2011) propose models for general cross-nodes that satisfy all requirements. Gibb (2011) proposes an alternative formulation that also satisfies all requirements but is more computationally demanding to solve.

In this paper we adopt the node model proposed in Tampère et al. (2011), which satisfies all requirements, although we would like to point out that our formulation can handle essentially any node model, as our formulation in Section 3.1 is generic. Appendix A outlines a condensed version of the iterative procedure that Tampère et al. (2011) describe for generating reduction factors consistent with their node model requirements.

4. Solution algorithm

In this section we present algorithms for finding a stochastic user equilibrium solution. It is a bi-level simulation based algorithm, in which at the outer level we solve the route choice problem, defined by VI problem (2), while at the inner level we solve our capacity constrained static network loading model with residual point queues, defined by fixed point problem (15).

4.1 Algorithm for solving the route choice problem

The route choice problem defined in inequality (2) is different from most formulations that focus on finding a deterministic user equilibrium (DUE). Several efficient algorithms have been proposed in the literature to determine a DUE, such origin-based assignment (OBA, Bar-Gera, 2002), linear user cost equilibrium (LUCS, Gentile and Noekel, 2009), and traffic assignment by paired alternative segments (TAPAS, Bar-Gera, 2010). Bar-Gera et al. (2012) notes the importance of path proportionality, which becomes even more relevant when including a node model.

We propose the more general notion of a stochastic user equilibrium (SUE) and adopt a logit-based traffic assignment model, which ensures path proportionality. For the SUE problem, only few efficient path-based solution algorithms have been proposed (see Bekhor and Toledo, 2005) which mostly rely on gradients of the objective function. Since the travel times in our model formulation are given by an implicit function, deriving gradients is not trivial. Our main contribution is on the network loading part, hence in this paper we adopt the well-known but simple method of successive averages (MSA, Sheffi and Powell, 1982) for iteratively finding a stochastic user equilibrium solution. Liu et al. (2009) proposed some variations of MSA that may be more efficient.

An initial route set is input into the algorithm. This may contain only a single fastest route per OD pair, or can include multiple routes per OD pair. We would like to point out that using a large initial route set that contains multiple (relevant) routes per OD pair will speed up convergence, particularly in heavily congested networks. We adopt a stochastic route choice generator as proposed in Fiorenzo-Catalano et al. (2004), which will determine multiple routes per OD pair. This route set generation procedure is able to produce route sets without irrelevant alternatives, which is an important feature in logit based route choice models (see Bliemer and

Bovy, 2008). The initial route set may not include *all* relevant routes, hence updating the route set with newly found fastest routes after each iteration in the assignment may be necessary.

In order to check whether convergence has been reached, we adopt a gap function. Because we search for an SUE, we cannot apply existing gap functions that have been developed for DUEs as they are unable to identify convergence towards an SUE. Instead, we propose a new gap function geared towards the properties of an SUE.

Define for each OD pair, $\pi_{rs} = \min_{p \in P_{rs}^{(i)}} \{ \theta c_p + \ln(f_p) \}$, where $P_{rs}^{(i)}$ is the route set at the end of iteration i , including the actual fastest route. By taking the logarithm of Equation (1), it follows that for each OD pair (r, s) , all routes $p \in P_{rs}^{(i)}$ have the same value $\theta c_p + \ln(f_p) = \pi_{rs}$. This leads to the following relative gap function for each iteration i that will reach zero upon convergence:

$$G^{(i)} = \frac{\sum_{(r,s)} \sum_{p \in P_{rs}^{(i)}} f_p (\theta c_p + \ln(f_p) - \pi_{rs})}{\sum_{(r,s)} D_{rs} \pi_{rs}}. \quad (20)$$

Algorithm (outer level)

Input: Initial route sets $P_{rs}^{(0)}$ and travel demand D_{rs} for each OD pair (r, s) , assignment maps δ_{ap} , and link characteristics.

Step 0: *Initialisation.* Assume an empty network, i.e., $q_a^{(0)} = 0$ for all $a \in A$, and $f_p^{(0)} = 0$ for all $p \in P_{rs}^{(0)}$. Further, assume that all reduction factors are equal to one (no constraints), i.e., $\alpha_a = 1$ for all $a \in A$, and set all link costs to free-flow costs, $c_a^{(0)} = \tau_a(\mathbf{0})$, yielding free-flow route costs $c_p^{(0)}$. Set $i := 1$.

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

Step 2: *Route flow averaging.* Compute the new MSA averaged route flows by $\mathbf{f}^{(i)} = \mathbf{f}^{(i-1)} + \frac{1}{i}(\tilde{\mathbf{f}}^{(i)} - \mathbf{f}^{(i-1)})$.

Step 3: *Network loading.* Solve fixed point problem (15), see Section 4.2, yielding link flows $\mathbf{q}^{(i)}$ and reduction factors $\boldsymbol{\alpha}^{(i)}$.

Step 4: *Travel cost calculation.* Compute the link travel times $\tau_a^{(i)}$ using Equation (17) and the route travel costs $c_p^{(i)}$ using Equations (5)-(6).

Step 5: *Route set updating.* For each OD pair (r, s) , determine the fastest path ρ_{rs} based on link travel times $\tau_a^{(i)}$ and update the route set $P_{rs}^{(i)} := P_{rs}^{(i-1)} \cup \rho_{rs}$. If $\rho_{rs} \notin P_{rs}^{(i-1)}$, then set $f_{\rho_{rs}}^{(i)} = 0$.

Step 6: *Convergence check.* Calculate the gap $G^{(i)}$ using Equation (20). If $G^{(i)} < \varepsilon_i$ for some pre-determined small $\varepsilon_i > 0$, then we stop. Otherwise, we set $i := i+1$ and return to Step 1.

4.2 *Algorithm for solving the network loading problem*

In this section we describe the algorithm for solving the network loading in Step 3 of the outer level algorithm. We note that since we may be dealing with very large networks, solving fixed point problem (15) requires a very efficient algorithm. An important part of the algorithm is determining which links and turns are (potentially) blocked.

In order to determine which turns may be blocked, we start by performing a normal static traffic assignment with $\alpha_a = 1$ for all links. Then we compute the volume/capacity ratios for each link. If a link has a volume/capacity ratio larger than one, it is a potential bottleneck. All turns into a potential bottleneck link are therefore potentially blocked. In addition to that, other turns sharing the same inlink of directly potentially blocked turns but that do not lead to a directly blocked outlink are also (indirectly) potentially blocked. This is because we adopt a node model that obeys the FIFO diverging rule and is vital for realistic results. We will illustrate this with an example in Section 5.2. We can discard all other turns and the associated reduction factors from the fixed point problem (i.e., they are fixed at one). Route flows that do not pass through any potentially blocked turns will be assigned as in a traditional unconstrained network loading, while for routes passing through one or more potentially blocked turns we determine the capacity constrained network loading by solving the fixed point problem. Note that for the fixed point problem, we can represent the potentially blocked routes in terms of consecutive potentially blocked turns (since setting a reduction factor equal to one is the same as removing the reduction factor, see Equation (11)), which greatly reduces the size of the problem.

Denote the reduced set of potentially blocked routes by \tilde{P} . We iteratively assign all flows on routes in this reduced route set with reduction factors resulting from the node model described in Tampère et al. (2011) until the reduction factors (and hence the flows) do not change (much) anymore between iterations. In that case, a fixed point has been found.

Algorithm (inner level)

Input: Route set $P^{(i)}$, route flows $\mathbf{f}^{(i)}$, and link capacities \mathbf{C} .

Step 0: *Initialization.* Initialize all turn reduction factors with $\boldsymbol{\alpha}^{(0)} = \mathbf{1}$.

Step 1: *Perform network loading of all routes.* For all paths $p \in P^{(i)}$, assign the route flows $\mathbf{f}^{(i)}$ to the network using Equation (11), calculate the turn flows $\mathbf{t}^{(0)}$ using Equation (12), and calculate link flows \mathbf{q} using Equation (13). Set $j := 1$.

Step 2: *Compute potentially blocked turns.* For each link $b \in A$, if $q_b > C_b$, then link b is a potential bottleneck and all turn flows t_{ab} that have a positive flow will be potentially blocking, as well as all other turns from a .

Step 3: *Determine potentially blocked route set.* Put all routes that pass through one or more potentially blocking turns into the reduced potentially blocking route set \tilde{P} .

Step 4: *Compute turn reduction factors.* Using turn flows $\mathbf{t}^{(j-1)}$ and link capacities \mathbf{C} , apply the node model in Equation (14) to obtain turn reduction factors $\boldsymbol{\alpha}^{(j)}$.

Step 5: *Compute desired turn flows.* For all potentially blocked turns along routes in \tilde{P} , compute the path-specific link flows using $\boldsymbol{\alpha}^{(j)}$ and $\mathbf{f}^{(i)}$ in Equation (11) and calculate the turn flows $\mathbf{t}^{(j)}$ using Equation (12).

Step 6: *Convergence check.* If $|\mathbf{t}^{(j)} - \mathbf{t}^{(j-1)}| < \varepsilon_2$ for some $\varepsilon_2 > 0$, then we have converged to a fixed point, go to Step 7. Otherwise, we set $j := j + 1$ and return to Step 4.

Step 7: *Perform network loading of potentially blocked routes.* We do a final network loading using route flows $\mathbf{f}^{(i)}$ and reduction factors $\alpha^{(j)}$ in order to calculate final link flows \mathbf{q} using Equations (11) and (13). Note that we only need to assign flows along routes that are in route set \tilde{P} , as we can retrieve the route flows assigned over non-blocked routes $p \in P^{(i)} \setminus \tilde{P}$ from Step 1.

In this algorithm, we perform one complete network loading in Step 1, we cycle through all routes in Step 3, and we do an additional network loading of the potentially blocked routes in Step 7. These three steps are the most time consuming steps as they require calculations on large route sets. This implies that our inner level algorithm has the same computational complexity as a traditional static network loading, although the computation time will be in most cases two or three times longer. Since the fixed point iterations only involve potentially blocked turns, in practical networks they require only little computation time.

In the algorithm described above, we have used smart ways of reducing the size of the fixed point problem, but have further adopted a straightforward fixed point algorithm. Potentially further gains in computational efficiency can be achieved by adopting accelerated averaging procedures, see e.g. Bottom and Chabini (2001).

5. Numerical examples and case studies

In this section we will present some numerical examples on two small hypothesized networks to illustrate how the models and algorithms work. The first network (single OD pair with multiple routes) has been designed to illustrate predominantly the outer level algorithm, while the second network (multiple OD pairs with a single route) has been constructed to be able to illustrate the inner level algorithm. Further, we will demonstrate that it is feasible to apply our new methodology to large real life networks. For all studies, we assume a 2-hour peak period with a given OD matrix (i.e., $T = 2$), and we use the proposed QL fundamental diagram for all links. The model has been implemented in C++ in the OmniTRANS transport planning software using the StreamLine framework.

5.1 Hypothetical network 1: Single OD pair with multiple routes

Consider the network in Figure 4, in which there are four routes from origin r to destination s with a travel demand of 8000 veh/h. All links are assumed to have a length of 2 km, a maximum speed of 100 km/h, and a critical speed (speed at capacity) of 75 km/h. The link capacities are stated in Figure 4. The network has been constructed such that it causes severe congestion in the network. Our stochastic route set generator was able to find all four routes, which were included in the initial route set. Table 3 lists the stochastic user equilibrium solution after 20 iterations, using a scale parameter of $\theta = -7$ in the logit model.

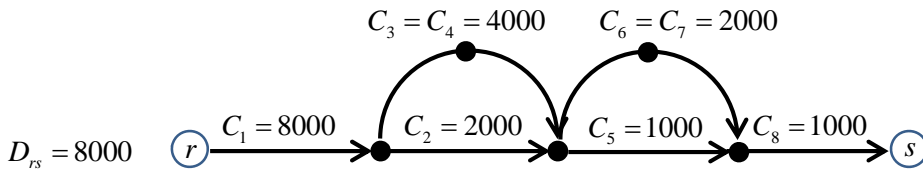


Figure 4: Hypothetical network with a single OD pair with multiple routes

**Capacity constrained stochastic static traffic assignment with residual point queues
incorporating a proper node model**

Bliemer, Raadsen, Smits, Zhou and Bell

Table 3: Route and link flows and costs for the single OD pair network

route	f_p	$\theta c_p + \ln(f_p)$	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8
1	952	31.84	952	686	--	--	336	--	--	112
2	1824	31.59	1824	1314	--	--	--	645	645	224
3	1792	31.59	1792	--	1291	1291	664	--	--	221
4	3432	31.72	3432	--	2473	2473	--	1271	1271	442
Total	8000		8000	2000	3764	3764	1000	1916	1916	1000
route	c_p		c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
1	3.546		0.412	1.081	--	--	2.027	--	--	0.027
2	3.453		0.412	1.081	--	--	--	0.026	1.907	0.027
3	3.456		0.412	--	0.026	0.964	2.027	--	--	0.027
4	3.362		0.412	--	0.026	0.964	--	0.026	1.907	0.027
α_a			0.72	0.49	1.00	0.51	0.33	1.00	0.35	1.00

The route flows can be derived using the travel demand and the logit choice probabilities in Equation (1). For example, $f_1 = e^{-7.3.546} / (e^{-7.3.546} + e^{-7.3.453} + e^{-7.3.456} + e^{-7.3.362}) \cdot 8000 \approx 952$. We can further see from the table that this result after 20 iterations approximates an SUE, since the value of $\theta c_p + \ln(f_p)$ is almost the same for all routes. The solution is converging to a SUE, as can be seen in Figure 5 where we have plotted the relative gap defined in Equation (20) on a logarithmic scale.

In the network, there is a diverge node, a cross node, and a merge node. The reduction factors α_a listed in the table are determined by the node model of Tampère et al. (2011). A reduction factor smaller than one implies that there is a queue on that link. Only links 3, 6, and 8 do not have a queue. For the diverge, link 2 is the bottleneck with a capacity of 2000, while the desired flow is $952+1824 = 2776$. Hence, the reduction factor is $\alpha_1 = 2000/2776 = 0.72$. For the merge, both links 5 and 7 are queuing due to competition for a capacity of only 1000 on link 8. The capacity will be distributed according to the proportions of the capacity of links 5 and 7. Hence, link 5 gets a capacity of $C_5 / (C_5 + C_7) \cdot C_8 = 333$, and link 7 obtains a capacity of $C_7 / (C_5 + C_7) \cdot C_8 = 667$. Given the desired flows of 1000 and 1921, this means that the reduction factors for links 5 and 7 are $\alpha_5 = 333/1000 = 0.33$ and $\alpha_7 = 667/1921 = 0.35$. For the cross-node, the calculation is somewhat more complicated, such that we refer to the algorithm provided in Tampère et al. (2011) in order to obtain reduction factors of 0.49 and 0.51 for links 2 and 4, respectively.

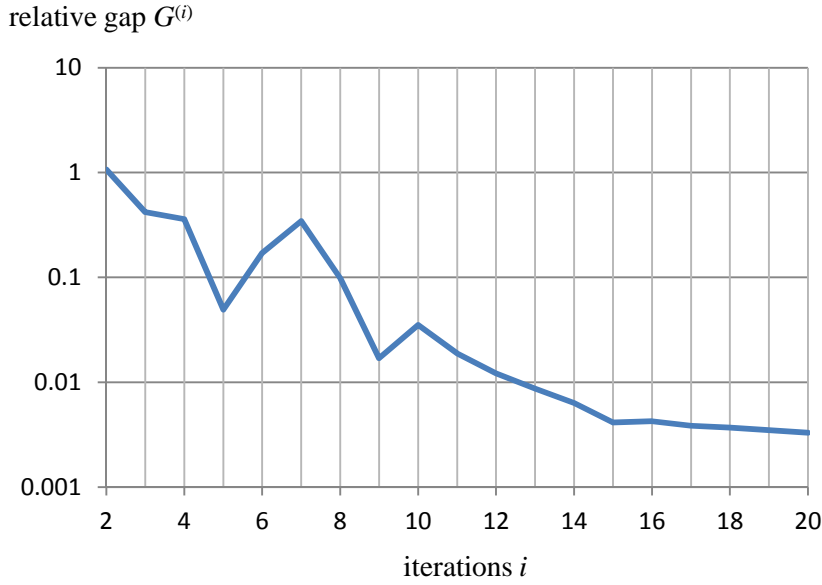


Figure 5: Relative gap (logarithmic vertical scale)

The link costs can be computed using Equations (17) and (19). For example, consider link 5, which is at capacity. The free-flow travel time for link 5 is $\tau_5^{\text{ff}}(1000) = 2/75 = 0.027$, while the queuing delay equals $\frac{1}{2}(1 - \alpha_5)/\alpha_5 \cdot 2 = 2$, hence the total travel time is $c_5 = 0.027 + 2 = 2.027$.

5.2 Hypothetical network 2: Multiple OD pairs with a single route

Now consider the network in Figure 6, which has three OD pairs – (r_1, s_1) , (r_2, s_2) , and (r_3, s_3) – each only having a single route. There are three cross-nodes in the network. The network has been constructed in such a way that each node is visited by each route, hence each turn reduction factor influences the flows in the entire network, which makes this network problem non-trivial to solve. We will demonstrate how the fixed point iterations in the inner level algorithm rapidly converge to flows that are consistent with the node model. In this network, we assume that all link lengths are 1 km, all maximum speeds are 100 km/h, and critical speeds are 75 km/h. All links have a capacity of 2000 veh/h. The travel demand for each OD pair is 2000 veh/h, such that the route flows are $f_1 = f_2 = f_3 = 2000$, in which route flow f_p corresponds with OD pair (r_p, s_p) .

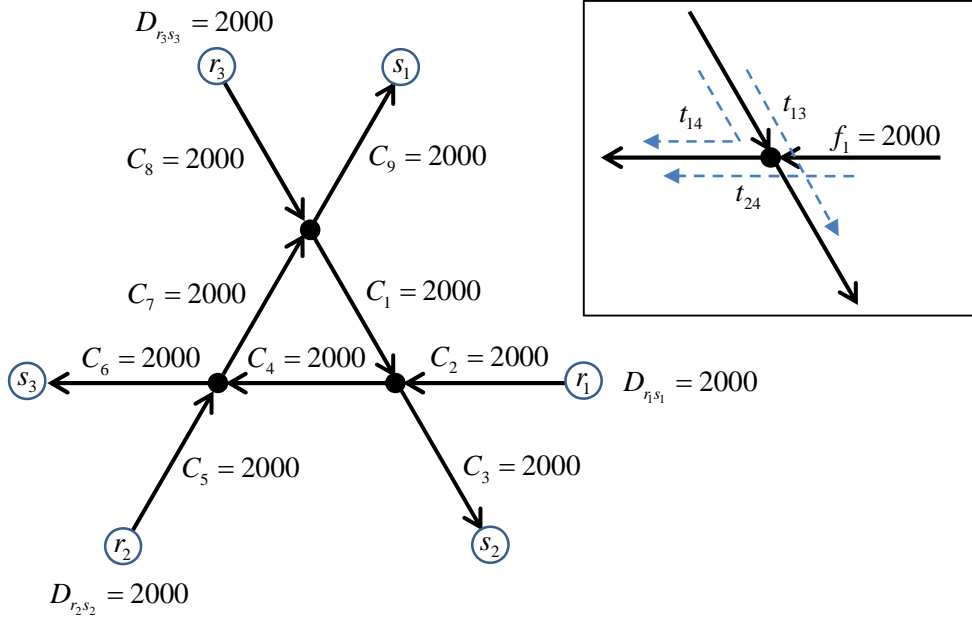


Figure 6: Hypothetical network with multiple OD pairs with a single route

We will concentrate our analyses on the node on the bottom right, with incoming links 1 and 2 and outgoing links 3 and 4, noting that the results for the other links are identical due to symmetry. Looking at the turn flows for this node, we note that t_{13} is influenced by route flow f_2 , turn flow t_{14} is influenced by route flow f_3 , and turn flow t_{24} is influenced by route flow f_1 . Furthermore, due to capacity constrained flow propagation, turn flows t_{13} and t_{14} are influenced by turn reduction factors at both other nodes. Finally, turns t_{14} and t_{24} are related because they are competing for the same link 4 capacity, while turns t_{13} and t_{14} are related because of the FIFO diverging rule. Hence, there exist many dependencies in this network, which makes this simple example interesting to solve.

Focussing on the bottom right node, in the first stage of the inner level algorithm, we determine the potentially bottleneck links and blocked turns by performing an unconstrained network loading. This yields a flow of 2000 veh/h on links 3 and 4, and a flow of 4000 veh/h on links 1 and 2. These latter links are therefore potential bottlenecks. This means that turns t_{14} and t_{24} are potentially directly blocked by this bottleneck, while turn t_{13} is indirectly potentially blocked due to the FIFO diverging rule.

Table 4 lists the turn flows and reduction factors until convergence is achieved on the inner level algorithm. Using Equations (11) and (12), it follows that $t_{13} = \alpha_5 \alpha_7 f_2$, $t_{14} = \alpha_8 f_3$, and $t_{24} = f_1$, and from Equation (14) it follows that (considering symmetry of the network) $\alpha_7 = \alpha_1 = \Gamma_1(t_{13}, t_{14}, t_{24} | C_1, C_2, C_3, C_4)$, and $\alpha_5 = \alpha_8 = \alpha_2 = \Gamma_2(t_{13}, t_{14}, t_{24} | C_1, C_2, C_3, C_4)$.

Table 4: Route and link flows and costs for the multiple OD pair network

iteration j	α_1	α_2	t_{13}	t_{14}	t_{24}
0	1.0000	1.0000	2000	2000	2000
1	0.3333	0.6667	444	1333	2000
2	0.6429	0.5714	735	1143	2000
3	0.6622	0.6216	823	1243	2000
4	0.6043	0.6244	755	1249	2000
5	0.6150	0.6160	758	1232	2000
6	0.6208	0.6176	767	1235	2000
7	0.6178	0.6184	764	1237	2000
8	0.6177	0.6180	763	1236	2000
9	0.6181	0.6180	764	1236	2000

In the initialization ($j=0$), a normal (unconstrained) static assignment is performed, yielding all turn flows equal to 2000 veh/h. The node model by Tampère et al. (2011) yields $\alpha_1 = 0.33$, and $\alpha_2 = 0.67$, such that the inflow into link 4 is restricted to 2000 veh/h. The desired turn flows are then updated to $t_{13} = 0.67 \cdot 0.33 \cdot 2000 = 444$, $t_{14} = 0.67 \cdot 2000 = 1333$, and $t_{24} = 2000$. The turn flows have stabilized after nine iterations, which yields a solution to our fixed point problem in Equation (15). Using the resulting turn reduction factors to conduct the final capacity constrained network loading yields link flows $q_1 = \alpha_5 \alpha_7 f_2 + \alpha_8 f_3 = 2000$, $q_2 = f_1 = 2000$, $q_3 = \alpha_5 \alpha_7 \alpha_1 f_2 = 472$, and $q_4 = \alpha_1 f_1 + \alpha_8 \alpha_1 f_3 = 2000$. In other words, for each route flow of 2000 veh/h that enters a route, only 472 vehicles have arrived after 1 hour while the remaining vehicles are waiting in one of the queues on the network.

5.2 Large real life networks

Figure 7 shows two real life networks from the Netherlands (networks of the cities of Amsterdam and Rotterdam) and two from Australia (networks of the cities Gold Coast and Sydney) that we considered. Table 5 presents the size of the networks, where the Amsterdam network is the smallest and the Sydney network the largest (for which we generated more than 2 million routes). The table also states the CPU time for each iteration in the outer (route choice) level, which includes applying the logit model (route choice) and solving the fixed point problem (network loading). Note that even for large networks, the current prototype implementation takes less than 4 minutes per iteration on a standard computer, which is fairly fast.

Consider for example the Sydney network. The CPU times reported in Table 5 are per route choice iteration, including calculating the route choice proportions for over 1 million OD pairs, and solving the fixed point capacity constrained network loading problem in the capacity constrained traffic assignment submodel. In the first iteration, 1,333 nodes were potentially blocked, yielding 1,799,407 blocked routes. This number decreased over the number of route

Capacity constrained stochastic static traffic assignment with residual point queues incorporating a proper node model

Bliemer, Raadsen, Smits, Zhou and Bell

choice iterations, as the travel demand is spread more over routes. The maximum number of blocked turns on a single route turned out to be 152. The computations required 3.5 GB of RAM. Note that the CPU times are based on a prototype implementation. We can likely achieve efficiency gains that will bring down the required CPU time and memory usage.



Figure 7: Real life networks

Table 5: Network data, computation time, and memory use

Network	Number of TAZs	Number of links	Number of nodes	Number of routes	Number of OD pairs	Number of vehicles	CPU time per iteration ³
Amsterdam ¹	418	9,408	4,281	266,505	275,722	271,772	8 sec.
Rotterdam ¹	1,744	17,187	6,422	1,394,853	737,415	260,324	53 sec.
Gold Coast ²	1,067	9,565	2,987	1,221,524	592,856	243,838	70 sec.
Sydney ²	3,264	75,379	30,573	2,394,496	1,045,156	1,569,698	205 sec.

¹ Network and OD matrix kindly provided by Goudappel Coffeng BV, The Netherlands

² Network and OD matrix kindly provided by Veitch Lister Consulting Pty Ltd, Australia

³ Using a notebook computer with Intel Core i7 @ 2.80Ghz running Windows 7

Even though we have not calibrated the OD matrices, inspecting the results of our traffic assignment after 20 iterations, we notice in all four case studies that traffic flows are at realistic levels and queues appear upstream at expected bottleneck locations. The outcomes seem to be a significant improvement over the outcomes of a traditional static traffic assignment model in which traffic flows exceed link capacities and delays are predicted in the wrong locations. We do note that we have not yet included signalized intersections, which in an urban setting will further constrain the turn capacities at intersections. Tampère et al. (2011) propose a relatively simple extension of the node model to include such further restrictions on the turn capacities by taking the proportion of green time per turn into account. As mentioned earlier, our model framework can accept any node model and including signalized intersections is therefore a relatively straightforward extension (although data on green times may not be readily available in existing static network models).

6. Conclusions and discussion

In this paper we have proposed a novel capacity constrained static traffic assignment model with residual point queues. Our novel model is able to produce more realistic results compared to existing static assignment techniques due to incorporating a proper node model into the network loading procedure. Such a node model distributes link capacities to turns more accurately, resulting in queue formation in the appropriate locations. To our knowledge, this is the first attempt to include proper first order node models into static traffic assignment and obtain consistent capacity constrained network flows.

Our model is route based, adopting a logit based assignment, formulated as a variational inequality problem. The capacity constrained network loading model that propagates the flow along the routes is formulated as a fixed point problem that can take any node model into account. In this paper we have adopted the node model by Tampère et al. (2011), which is a first order macroscopic node model that satisfies a list of required properties.

We have proposed a solution algorithm and illustrated how to apply it to our model on hypothetical networks. Further, we demonstrated feasibility on large scale real-life networks. We believe that our new model is able to replace existing traditional static traffic assignment models, because it offers a way to include capacity constraints more realistically and practically than any other static model to date and as a consequence is able to obtain much more realistic

traffic flows and travel times, while maintaining a rigorous but elegant mathematical problem formulation.

Clearly, our novel model formulation opens up a whole new stream of research and there are still many interesting properties and model extensions to be investigated. We will mention a few. First, although we have rigorously formulated our model, we have not investigated solution properties like existence and uniqueness, as this is a whole study in itself. We have run the model on many networks and implemented several algorithms. In all cases we were able to find a solution, and in all cases we found the same solution no matter what algorithm we adopted. This gives us confidence that the model and the solution are likely to have nice properties, but this remains a subject of further research. Secondly, while our algorithm converged on a series of test networks, we have not proven convergence in general. It may be possible to construct networks in which our simple inner level algorithm is not able to converge to a fixed point due to some interdependencies in the turn flows along routes. It should be noted that such situations in general networks are rare. Our algorithm can be adapted to take such dependencies into account, but it would lead to a more complex algorithm, which is outside the scope of this paper. Thirdly, we have only considered vertical queues in this paper instead of horizontal (physical) queues, which may lead to spillback. An extension to include physical queues and spillback would move the model away from a static model towards a quasi-dynamic model. However, in this paper we already adopted a realistic fundamental diagram instead of standard link performance functions, which opens the door to possible extensions that consistently include physical queuing without the need to rely on post processing as in Bliemer et al. (2012). Fourthly, the node model currently adopted does not take into account internal node constraints. The paper by Tampère et al. (2011) touches lightly on the subject and there is other research that indicates that including these kinds of constraints can potentially lead to problems in finding unique solutions (Corthout et al., 2012). Further research on this subject is needed in order to be able to successfully integrate signalized- and other forms of complex intersection configurations. Finally, in this paper we have focused on methodology. A closer look at the traffic assignment outcomes by comparing predicted traffic flows with link counts and predicted travel times with measured travel times in a calibrated model will tell us more about the level of improvement over a traditional assignment model, and how it compares to outcomes of dynamic assignment models.

Acknowledgments

We would like to thank Luuk Brederode and Luc Wismans for their comments on the draft version of the paper, and Omnitrans International for providing the Streamline framework. Furthermore, we would like to thank Wenlong Jin for the useful discussions on the node model. The paper solely expresses the opinion of the authors. This research is partly sponsored by funding from the Australian Research Council (LP130101048).

Appendix A

Tampère et al. (2011) provide an iterative algorithm for determining the reduction factors α_a in the case of a general cross-node. In this appendix we provide a quick summary in our own notation, and refer to Tampère et al. (2011) for details. The maximum number of iterations needed in the algorithm equals the number of links leading into the node n under consideration, i.e., the cardinality of the set A_n^{in} . Denote each iteration with index m . Further, let $Y^{(m)}$ and $Z^{(m)}$ denote the set of inlinks that are either demand constrained or capacity constrained in iteration m , respectively, and let $X^{(m)}$ contain the inlinks that have not yet been processed in iteration m . Assume that turn flows (demand) t_{ab} and all link capacities (supply) C_a and C_b are given for each $a \in A_n^{\text{in}}$ and $b \in A_n^{\text{out}}$. Define the following scaling factor, $\lambda_a = C_a / \sum_b t_{ab}$. Finally, let $R_b^{(m)}$ denote the available capacity of outlink $b \in A_n^{\text{out}}$ in iteration m . The algorithm below describes the iterative process to determine the reduction factors.

Algorithm (node model)

Step 1: *Initialization.* Set $m := 1$ and set $X^{(1)} = A_n^{\text{in}}$, $Y^{(1)} = Z^{(1)} = \emptyset$, and $R_b^{(1)} = C_b$.

Step 2: *Determine current most restricting outlink.* Find $\bar{b} = \arg \min_{b \in A_n^{\text{out}}} \left\{ \frac{R_b^{(m)}}{\sum_{a \in X^{(m)}} \lambda_a t_{ab}} \right\}$.

Step 3: *Calculate iteration reduction factor.* Determine $\beta^{(m)} = \frac{R_{\bar{b}}^{(m)}}{\sum_{a \in X^{(m)}} \lambda_a t_{a\bar{b}}}$.

Step 4: *Update sets.* Update demand constrained set $Y^{(m)} = \{a \in X^{(m)} \mid t_{a\bar{b}} > 0, \lambda_a \beta^{(m)} \leq 1\}$. If $Y^{(m)} = \emptyset$, then set capacity constrained set to $Z^{(m)} = \{a \in X^{(m)} \mid t_{a\bar{b}} > 0\}$, otherwise set $Z^{(m)} = \emptyset$. Update the set of unprocessed inlinks to $X^{(m+1)} = X^{(m)} \setminus \{Y^{(m)} \cup Z^{(m)}\}$.

Step 5: *Determine reduction factors.* Set $\alpha_a := \begin{cases} 1, & \text{if } a \in Y^{(m)}, \\ \lambda_a \beta^{(m)}, & \text{if } a \in Z^{(m)}, \\ \lambda_a, & \text{otherwise.} \end{cases}$

Step 6: *Update remaining outlink capacities.* Set $R_b^{(m+1)} = R_b^{(m)} - \sum_{a \in \{Y^{(m)} \cup Z^{(m)}\}} \alpha_a t_{ab}$, for all $b \in A_n^{\text{out}}$.

Step 7: *Termination.* If $X^{(m+1)} = \emptyset$, then stop. Otherwise, set $m := m + 1$ and return to Step 2.

References

- 4Cast (2009) Qblok_2004: toedelingsprocedure LMS/NRM (in Dutch). Working report.
- Bakker, D., P.H. Mijjer, and F. Hofman (1994) QBLOK: An assignment technique for modeling the dependency between bottlenecks and the prediction of 'grid lock'. *Proceedings of Colloquium Vervoersplanologisch Speurwerk*, Delft, pp. 313-332.
- Bar-Gera H. (2002) Origin-based algorithm for the transportation assignment problem. *Transportation Science* 36, pp. 398-417.
- Bar-Gera, H. (2010) Traffic assignment by paired alternative segments. *Transportation Research Part B*, Vol. 44, No. 8/9, pp. 1022-1046.
- Bar-Gera, H., D. Boyce, and Y. Nie (2012) User-equilibrium route flows and the condition of proportionality. *Transportation Research Part B*, Vol. 46, pp. 440-462.
- Beckmann, M., C. McGuire, and C. Winsten (1956) *Studies in the Economics of Transportation*. Yale University Press, Connecticut, USA.
- Bekhor, S., and T. Toledo (2005) Investigating path-based solution algorithms to the stochastic user equilibrium problem. *Transportation Research Part B*, Vol. 39, pp. 279-295.
- Bell, M.G.H. (1995) Stochastic user equilibrium in networks with queues. *Transportation Research Part B*, Vol. 29, No. 2, pp. 115-112.
- Ben-Akiva, M. and M. Bierlaire (1999) Discrete choice methods and their applications to short term travel decisions. In: R.W. Hall (ed.) *Handbook of transportation science*. USA: Kluwer Publishers, 5-34.
- Bifulco, G., and U. Crisalli (1998) Stochastic user equilibrium and link capacity constraints: formulation and theoretical evidences. *Proceedings of the European Transport Conference*.
- Bliemer, M.C.J. (2007) Dynamic queuing and spillback in an analytical multiclass dynamic network loading model. *Transportation Research Record* 2029, pp. 14-21.
- Bliemer, M.C.J., and P.H.L. Bovy (2008) Impact of route choice set on route choice probabilities. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2076, pp. 10-19.
- Bliemer, M., L. Brederode, L. Wismans, and E.-S. Smits (2012) Quasi-dynamic traffic assignment: static traffic assignment with queuing and spillback. *Presented at the Annual Meeting of the Transportation Research Board*, Washington DC, USA.
- Bottom, J. and I. Chabini (2001) Accelerated averaging methods for fixed point problems in transportation analysis and planning. *Proceedings of the Triennial Symposium on Transportation Analysis (TRISTAN IV)*, São Miguel, Azores, pp. 69-74.
- Boyce, D., B.N. Janson, R.W. Eash (1981) The effect on equilibrium trip assignment of different link congestion functions. *Transportation Research Part A*, Vol. 15, pp. 223-232.
- Bundschuh, M., P. Vortisch, and T. Van Vuuren (2006) Modelling queues in static traffic assignment. *Proceedings of the European Transport Conference*.

Capacity constrained stochastic static traffic assignment with residual point queues incorporating a proper node model

Bliemer, Raadsen, Smits, Zhou and Bell

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

Cascetta, E., A. Nuzzolo, F. Russo, and A. Vitetta (1996) A Modified Logit Route Choice Model Overcoming Path Overlapping Problems: Specification and Some Calibration Results for Interurban Networks. In: J.B. Lesort (ed.) *Proceedings of the International Symposium on Transportation and Traffic Theory*, Lyon, pp. 697-711.

Chudak, F., V.D.S. Eleuterio, and Y. Nesterov (2007) Static Traffic Assignment Problem. A comparison between Beckmann (1956) and Nesterov & de Palma (1998) models. Conference paper STRC 2007.

Corthout, R., G. Flötteröd, F. Viti, and C.M.J. Tampère (2012) Non-unique flows in macroscopic first-order intersection models. *Transportation Research Part B*, Vol., 46, pp. 343-359.

Dafermos, S. (1980) Traffic equilibrium and variational inequalities. *Transportation Science*, Vol. 14, pp. 42-54.

Daganzo, C.F. (1977a) On the traffic assignment problem with flow dependent costs—I. *Transportation Research*, Vol. 11, pp. 433-437.

Daganzo, C.F. (1977b) On the traffic assignment problem with flow dependent costs—II. *Transportation Research*, Vol. 11, pp. 439-441.

Daganzo, C.F. (1995) The cell transmission model II: network traffic. *Transportation Research Part B*, Vol. 29, No. 2, pp. 79-93.

Daganzo, C.F., and Y. Sheffi (1977) On stochastic models of traffic assignment. *Transportation Science*, Vol. 11, No. 3, pp. 253-274.

Fiorenzo-Catalano, S., R. van Nes, and P.H.L. Bovy (2004) Choice set generation for multi-modal travel analysis. *European Journal of Transport and Infrastructure Research*, Vol. 4, No. 2, pp. 195-209.

Fisk, C. (1980) Some developments in equilibrium traffic assignment. *Transportation Research Part B*, Vol. 13, pp. 243-255.

Flötteröd, G., and J. Rohde (2011) Operational macroscopic modeling of complex urban road intersections. *Transportation Research Part B*, Vol. 45, No. 6, pp. 903-922.

Frank, M. and P. Wolfe (1956) An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, Vol. 3, pp. 95-110.

Gentile G., and K. Noekel (2009) Linear User Cost Equilibrium: the new algorithm for traffic assignment in VISUM. *Proceedings of European Transport Conference 2009*, The Netherlands.

Gibb, J. (2011) Model of traffic flow capacity constraint through nodes for dynamic network loading with queue spillback. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2263, pp. 113-122.

Guo, X., H. Yang, and T.L. Liu (2010) Bounding the inefficiency of logit-based stochastic user equilibrium. *European Journal of Operational Research*, Vol. 201, No. 2, pp. 463-469.

Capacity constrained stochastic static traffic assignment with residual point queues incorporating a proper node model

Bliemer, Raadsen, Smits, Zhou and Bell

Han, D.R., and H.K. Lo (2002) New alternating direction method for a class of nonlinear variational inequality problems. *Journal of Optimization Theory and its Applications*, Vol. 112, No. 3, pp. 549–560.

Hearn, D.W. (1980) Bounding flows in traffic assignment models. Research Report 80-4, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL.

Hearn, D.W., and J. Ribera (1980) Bounded flow equilibrium problems by penalty methods. In: *Proceedings of IEEE International Conference on Circuits and Computers*, pp. 162–166.

Inouye, H. (1987) Traffic equilibria and its solution in congested road networks. In: R. Genser (ed.) *Proceedings of IFAC Conference on Control in Transportation Systems*, pp. 267–272.

Jin, W.-L. (2012a) The traffic statics problem in a road network. *Transportation Research Part B*, Vol. 46, pp. 1360-1373.

Jin, W.-L. (2012b) A kinematic wave theory of multi-commodity network traffic flow. *Transportation Research Part B*, Vol. 46, pp. 1000-1022.

Jin, W.-L., and H.M. Zhang (2003) On the distribution schemes for determining flows through a merge. *Transportation Research Part B*, Vol. 37, No. 6, pp. 521–540.

Jin, W.L. and H.M. Zhang (2004) Multicommodity kinematic wave simulation model for network traffic flow. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1883, pp. 59-67.

Jorgensen, N.O. (1963) Some Aspects of the Urban Traffic Assignment Problem. Graduate Report, Institute of Transportation and Traffic Engineering, University of California, Berkeley, CA.

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

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

Lebacque, J.P., and M.M. Khoshyaran (2005) First-order macroscopic traffic flow models: Intersection modeling, network modeling. *Proceedings of the 16th International Symposium on Transportation and Traffic Theory (ISTTT)*, pp. 365-386.

Liu, H.X., X. He and B. He (2009) Method of successive weighted averages (MSWA) and self-regulated averaging schemes for solving stochastic user equilibrium problem. *Networks and Spatial Economics*, Vol. 9, pp. 485-503.

Luo, A., Z. Ruan, C. Huang, and H. Chen (2012) Feasible and descent direction method for continuous equilibrium network design problem. *Research Journal of Applied Sciences, Engineering and Technology*, Vol. 4, No. 22, pp. 4819-4824.

Lv, Q., A. Salis, G. Skrobanski, M.J. Smith, J. Springham, A. Woods, and A. Gordon (2007) Two-direction methods for variable demand traffic assignment. In: B.G. Heydecker (ed.), *Selected Proceedings of the 4th IMA International Conference on Mathematics in Transport*, Elsevier, pp. 71–96.

Capacity constrained stochastic static traffic assignment with residual point queues incorporating a proper node model

Bliemer, Raadsen, Smits, Zhou and Bell

Marcotte, P., S. Nguyen, and A. Schoeb (2004) A strategic flow model of traffic assignment in static capacitated networks. *Operations Research*, Vol. 52, pp. 191–212.

McFadden, D. (1974) Conditional Logit Analysis of Qualitative Choice Behavior. In: Zarembka (ed.) *Frontiers in Econometrics*, Academic Press, New York, 105-142.

Meng, Q., W.H.K. Lam, and L. Yang (2008) General stochastic user equilibrium traffic assignment problem with link capacity constraints. *Journal of Advanced Transportation*, Vol. 42, pp. 429-465.

Nesterov, Y., and A. de Palma (1998) Optimization formulations and static equilibrium in congested transportation networks, Technical report, CORE.

Nesterov, Y., and A. de Palma (2000) Stable dynamics solutions in transportation systems, Technical report, CORE.

Nesterov, Y., and A. de Palma (2003) Stationary dynamics solutions in congested transportation networks: Summary and perspectives. *Networks and Spatial Economics*, Vol. 3, pp. 371-395.

Newell, G.F. (1993) A simplified theory of kinematic waves in highway traffic, Part I: General theory, Part II: Queuing at freeway bottlenecks, Part III: Multi-destination flows, *Transportation Research Part B*, Vol. 27, pp. 281-313.

Nie, Y., H. Zhang, D. Lee (2004) Models and algorithms for the traffic assignment problem with link capacity constraints. *Transportation Research Part B*, Vol. 38, pp. 285-312.

Payne, H.J., and W.A. Thompson (1975) Traffic assignment on transportation networks with capacity constraints and queueing. Paper presented at the 47th National ORSA Meeting/TIMS 1975 North-American Meeting, Chicago, IL.

Prashker, J.N., and T. Toledo (2004) A gradient projection algorithm for side-constrained traffic assignment. *European Journal of Transport and Infrastructure Research*, Vol. 4, No. 2, pp. 177-193.

Shahpar, A., H. Aashtiani, and A. Babazadeh (2008) Dynamic penalty function method for the side constrained traffic assignment problem. *Applied Mathematics and Computation*, Vol. 206, pp. 332-345.

Sheffi, Y. and W.B. Powell (1982) An algorithm for the equilibrium assignment problem with random link times. *Networks*, Vol. 12, No. 2, pp. 191–207.

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

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

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

**Capacity constrained stochastic static traffic assignment with residual point queues
incorporating a proper node model**

Bliemer, Raadsen, Smits, Zhou and Bell

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

Wardrop, J.G. (1952) Some theoretical aspects of road traffic research. *Proceedings of the Institute of Civil Engineers*, Part II, pp. 325-378.

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

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

Zhou, Z., A. Chen, and S. Bekhor (2012) C-logit stochastic user-equilibrium model: formulations and solution algorithm. *Transportmetrica*, Vol. 8, No. 1, pp. 17-41.