

UC Berkeley

UC Berkeley Previously Published Works

Title

On node models for high-dimensional road networks

Permalink

<https://escholarship.org/uc/item/59x2m66n>

Authors

Wright, Matthew A.
Gomes, Gabriel
Horowitz, Roberto
[et al.](#)

Publication Date

2017-11-01

DOI

10.1016/j.trb.2017.09.001

Peer reviewed

On node models for high-dimensional road networks

Matthew A. Wright^{1,2}, Gabriel Gomes², Roberto Horowitz^{1,2}, Alex A. Kurzhanskiy²

¹Department of Mechanical Engineering, University of California, Berkeley

²Partners for Advanced Transportation Technologies (PATH), University of California, Berkeley

September 5, 2017

Abstract

Macroscopic traffic models are necessary for simulation and study of traffic’s complex macro-scale dynamics, and are often used by practitioners for road network planning, integrated corridor management, and other applications. These models have two parts: a link model, which describes traffic flow behavior on individual roads, and a node model, which describes behavior at road junctions. As the road networks under study become larger and more complex — nowadays often including arterial networks — the node model becomes more important. Despite their great importance to macroscopic models, however, only recently have node models had similar levels of attention as link models in the literature. This paper focuses on the first order node model and has two main contributions. First, we formalize the multi-commodity flow distribution at a junction as an optimization problem with all the necessary constraints. Most interesting here is the formalization of input flow priorities. Then, we discuss a very common “conservation of turning fractions” or “first-in-first-out” (FIFO) constraint, and how it often produces unrealistic spillback. This spillback occurs when, at a diverge, a queue develops for a movement that only a few lanes service, but FIFO requires that all lanes experience spillback from this queue. As we show, avoiding this unrealistic spillback while retaining FIFO in the node model requires complicated network topologies. Our second contribution is a “partial FIFO” mechanism that avoids this unrealistic spillback, and a (first-order) node model and solution algorithm that incorporates this mechanism. The partial FIFO mechanism is parameterized through intervals that describe how individual movements influence each other, can be intuitively described from physical lane geometry and turning movement rules, and allows tuning to describe a link as having anything between full FIFO and no FIFO. Excepting the FIFO constraint, the present node model also fits within the well-established “general class of first-order node models” for multi-commodity flows. Several illustrative examples are presented.

Keywords: macroscopic first order traffic model, first order node model, multi-commodity traffic, dynamic traffic assignment, dynamic network loading

1 Introduction

Traffic simulation models are vital tools for traffic engineers and practitioners. As in other disciplines focusing on complex systems, such as climate or population dynamics, traffic models have helped to deepen our understanding of traffic behavior. They are widely used in transportation planning projects in which capital investments must be justified with simulation-based studies (Caltrans, 2015). Recently, with the increased interest in Integrated Corridor Management (ICM) and Decision Support Systems (DSS), traffic models have also found a new role in real-time operations management. In both the planning and ICM/DSS contexts, the models have steadily grown larger - such as highway plans that append adjacent arterial

networks or managed lanes (Hadi et al., 2013) - and more complex - such as integration of new “smart” vehicle and communications technologies into real-time ICM.

Traffic models are typically divided into three categories based on their level of abstraction. At the most granular level, microscopic models simulate the motion and behavior of individual vehicles. At the other extreme, macroscopic models describe the evolution of traffic flows and buildup and breakdown of congestion along the lineal direction of a road in aggregate terms. Mesoscopic traffic models occupy the intermediate space. Of the three, the highly-abstracted macroscopic models unsurprisingly have the lowest computational cost, which makes them well-suited for study of large and complex networks of roads.

A macroscopic model is said to consist of a *link* model and a *node* model. A link model describes the evolution through time of the traffic flow along homogeneous sections of road. Several types of link models exist, such as those that give link flows as functions of link densities (e.g., the widely-used cell transmission model of Daganzo (1994) and its descendants), those that track the vehicles only at link boundaries (Yperman et al., 2005), and others (see Nie and Zhang (2005) for a broad overview). For the purposes of this paper, we do not specify a particular class of link model, but only require that the model describe, as a function of its state at time t , both the amount of vehicles trying to exit the link ($S(t)$, the link’s *demand*) and the amount of vehicles that the link is able to accept from upstream ($R(t)$, the link’s *supply*).

A traffic model may contain multiple classes of vehicles that share the road, and each class may have their own demands. Separate vehicle classes are often called *commodities*. In addition to the link and node models are the so-called *turning* or *split ratios*, which define the vehicles’ turning choice at the junction — the ratios of vehicles of each commodity that take each of the available movements. These split ratios might be measured, such as by manually counting flows of each movement at traffic intersections, estimated from some model, or prescribed by the modeler to make vehicles follow a certain path. In the context of this paper, we consider only the split ratios at particular junctions. Nodes join the links, and the node model computes the set of flows through a node for each commodity as a function of its incoming links’ demands and its outgoing links’ supplies.

Of course, while macroscopic models may be fast relative to more granular meso- and microscopic models, their computational needs are affected with the growth of network size and complexity. High computational cost can be exacerbated in modern ICM applications, as well. Many real-time traffic state estimation techniques follow an *ensemble method* approach, where many simulations describing different possible events are processed simultaneously (see e.g. Work et al. (2010), Wright and Horowitz (2016)). ICM decision-making can follow a similar approach, with multiple simulations being performed at a plan-evaluation step to project traffic outcomes under a range of possible future demands.

While the computational complexities stemming from links have been well-studied, node models can be sources of computational costs as well. These costs emerge when one models a network with many multi-input and/or multi-output junctions, or junctions where more than two links enter or exit. We adopt the term “high-dimensional” to describe these sorts of networks, to avoid ambiguity with similar terms such as “large,” which may also describe networks with many long roads and not many junctions. This paper draws on the authors’ experience in creating models for high-dimensional networks that describe a freeway, adjacent managed lanes (such as high-occupancy vehicle (HOV) lanes or tolled lanes), and/or the surrounding arterial grid for ICM purposes. We will discuss in Section 2 how modern node models can exacerbate computational complexity in high-dimensional networks by creating a trade-off between model accuracy and number of links; thankfully, this trade-off can be overcome in a simple manner, as we will detail in Section 3. Section 4 gives two examples, one being a slight modification of a well-studied example from Tampère et al. (2011), in which we demonstrate how our new node model considerations differ from previous node models in the junction-geometry information they consider and the flows produced. Appendix A summarizes the notation used in this paper.

2 Common node models and their drawbacks

2.1 Node models

The traffic node problem is defined on a junction of M input links, indexed by i , and N output links, indexed by j , with C vehicle commodities, indexed by c . As mentioned above, in first-order traffic models, the node model is said to consist of the mechanism by which, at time t , incoming links' per-commodity demands $S_i^c(t)$, split ratios $\beta_{i,j}^c(t)$ (which define the portion of vehicles of commodity c in link i that wish to exit to link j), and outgoing links' supplies $R_j(t)$ are resolved to produce throughflows $f_{i,j}^c(t)$. Nodes are generally infinitesimally small and have no storage, so all the flow that enters the node at time t must exit at time t . To simplify the notation, in the remainder of this paper we consider the node model evaluation in each time instant as an isolated problem (as we are assuming no storage), and omit the variable t for these quantities.

The node problem's history begins with the original formulation of discretized first-order traffic flow models (Daganzo, 1995). There have been many developments in the node model theory since, but we will reflect only on some more recent results.

We can divide the node model literature into pre- and post-Tampère et al. (2011) epochs. Tampère et al. (2011) drew from the literature several earlier-proposed node model requirements to develop a set of necessary conditions for first-order node models that they call the "general class of first-order node models." To review, these conditions are:

1. Applicability to general numbers of input links M and output links N . In the case of multi-commodity flow, this requirement also extends to general numbers of commodities c .
2. Maximization of the total flow through the node. Mathematically, this may be expressed as $\max \sum_{i,j,c} f_{i,j}^c$. According to Tampère et al. (2011), this means that "each flow should be actively restricted by one of the constraints, otherwise it would increase until it hits some constraint." When a node model is formulated as a constrained optimization problem, its solution will automatically satisfy this requirement. However, what this requirement really means is that constraints should be stated *correctly* and not be overly simplified and, thus, overly restrictive for the sake of convenient problem formulation. See the literature review in Tampère et al. (2011) for examples of node models that inadvertently do not maximize node throughput by oversimplifying their requirements.
3. Non-negativity of all input-output flows. Mathematically, $f_{i,j}^c \geq 0$ for all i, j, c .
4. Flow conservation: Total flow entering the node must be equal to total flow exiting the node. Mathematically, $\sum_i f_{i,j}^c = \sum_j f_{i,j}^c$ for all c .
5. Satisfaction of demand and supply constraints. Mathematically, $\sum_j f_{i,j}^c \leq S_i^c$ and $\sum_i f_{i,j}^c \leq R_j$.
6. Satisfaction of the first-in-first-out (FIFO) constraint: if a single destination j for a given i is not able to accept all demand from i to j , all flows from i are also constrained. This requirement is sometimes called "conservation of turning fractions" (CTF). Mathematically, $f_{i,j}^c / \sum_i f_{i,j}^c = \beta_{i,j}^c$.

We believe that in some situations, the FIFO constraint may be too restrictive. It should not be completely eliminated, but rather, could be relaxed through a parametrization. We will revisit this point in the following Section.

7. Satisfaction of the invariance principle. If the flow from some input link i is restricted by the available output supply, this input link enters a congested regime. This creates a queue in this input link and causes its demand S_i to jump to the link's capacity, which we denote F_i , in an infinitesimal time, and therefore, a node model should yield solutions that are invariant to replacing S_i with F_i when flow from input link i is supply-constrained (Lebacque and Khoshyaran, 2005).

This paper concerns multi-commodity traffic, which is unaddressed in the preceding list. To address multi-commodity traffic, we add another requirement,

8. Supply restrictions on a flow from any given input link are imposed on commodity components of this flow proportionally to their per-commodity demands.

Requirement 8 assumes that the commodities are mixed isotropically. This means that all vehicles attempting to take movement i, j will be queued in roughly random order, and not, for example, having all vehicles of commodity $c = 1$ queued in front of all vehicles of $c = 2$, in which case the $c = 2$ vehicles would be disproportionately affected by spillback. We feel this is a reasonable assumption for situations where the demand at the node is dependent mainly on the vehicles near the tail of the link (e.g., in a small cell at the tail).

In addition to the above numbered requirements, two other elements are required to define a node model. The first is a rule for the portioning of output link supplies R_j among the input links. Following Gentile et al. (2007), in Tampère et al. (2011) it was proposed to allocate supply for incoming flows proportionally to input link capacities F_i .

The second necessary element is a redistribution of “leftover supply.” Following the initial partitioning of supplies R_j , if one or more of the supply-receiving input links does not fill its allocated supply, some rule must redistribute the difference to other input links who may still fill it. This second element is meant to model the selfish behavior of drivers to take any space available, and ties in closely with requirement 2 above. Tampère et al. (2011) referred to these two elements collectively as a “supply constraint interaction rule” (SCIR).

Mentioned as an optional addition by Tampère et al. (2011) are “node supply constraints.” What is meant by these are supply quantities internal to the node, in addition to link supplies R_j . These node supplies are meant to model *shared resources* that multiple movements may make use of: each movement i, j through the node may or may not consume an amount of a node supply proportional to $\sum_c f_{i,j}^c$. A node supply being fully consumed would constrain these flows in a manner analogous to the link supplies. The obvious example of such a “shared resource” node supply is green time at a signalized intersection; see Tampère et al. (2011) for further discussion of this example. In Corthout et al. (2012) it was noted that these node supplies may lead to non-unique solutions. Very recently, Jabari (2016) revisited this non-uniqueness problem, introduced a relaxation of the flow-maximization objective termed a holding-free flow objective (i.e., not all holding-free flows are flow-maximizing) and proposed adding constraints concerning signal timing and the conflicts between movements (e.g., a through movement conflicts with the opposing left turn movement and if traffic drives on the right side of the road and the two cannot send vehicles at the same time) to achieve a unique flow solution. We will not make use of node supplies in the remainder of this paper.

The Tampère et al. (2011) requirements invite the modeler to design supply portioning and/or redistribution rules that, in some sense, model traffic behavior, add them to the numbered requirements, and create new node models. Recall that Tampère et al. (2011) suggested portioning the supplies R_j proportionally to the input links capacities F_i . Gibb (2011) suggested a supply portioning rule based on a so-called “capacity-consumption equivalence,” which states, qualitatively, that movements into a supply-constrained output link j' are characterized by drivers from individual input links i taking their movements i, j' one after the other in turn. The supply portioning rule thus says that the input links i , where drivers spend more time waiting for entry into congested links j , are assigned less supply from all links, including those not in congestion.

Smits et al. (2015) argue that the quantities of interest in the node model problem should not be the flows $f_{i,j}^c$, but rather the quantities $1/(\sum_c f_{i,j}^c)$, which are equal to the amount of time (relative to the simulation timestep) that a vehicle needs to make movement i, j through the node. They claim that this quantity is measurable in the field, and good node models would produce flows that emulate observed movement times. They review the node models proposed in Tampère et al. (2011) and Gibb (2011) in this re-parametrization and present others with different supply portioning and redistribution rules.

It is important to note that the node model in Gibb (2011) and the so-called “equal delay at outlink” model of Smits et al. (2015) define their supply portioning and redistribution rules *implicitly*. A consequence is that their solutions must be found through a fixed-point iterative procedure, whose iterations-until-convergence are potentially unbounded (this point was originally raised by Smits et al. (2015)). On the other hand, the capacity-proportional node model of Tampère et al. (2011) and “single server” model of Smits et al. (2015) are examples of *explicit* supply portioning rules.

For our purposes, we will focus on node models with explicit supply portioning and redistribution rules (that is, an explicit SCIR). Our SCIR, which will be introduced in Section 3, is similar to that of Tampère et al. (2011) and others in that it assigns leftover supply to other input links that can make use of it, but differs in what is considered “leftover” and how it is assigned.

While Tampère et al. (2011) used the capacity-proportional supply portioning rule, we will slightly generalize this to admit arbitrary per-input link priorities p_i in the spirit of Daganzo (1995), Ni and Leonard (2005), and Flötteröd and Rohde (2011).

2.2 Implications of the FIFO flow rule

We noted in Section 2.1 that the FIFO, or “conservation of turning fractions” rule in the Tampère et al. (2011) node requirements may be too restrictive. Consider the example junction depicted in Figure 1. It depicts a five-lane freeway with left and right off-ramps. The natural network representation of this junction would have one node with one input link, with demand S_1 , and three output links, with supplies R_1 , R_2 , and R_3 , similar to Figure 2a.

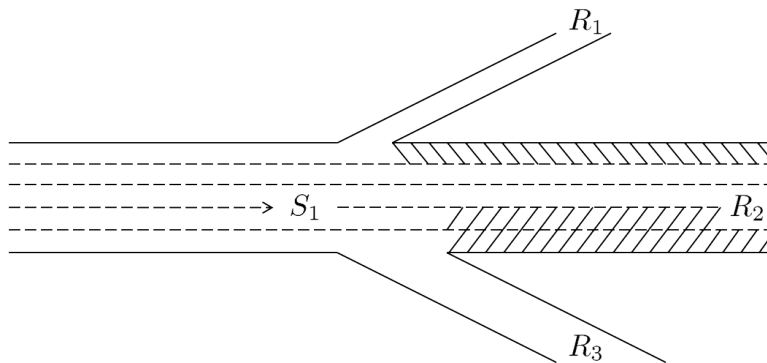


Figure 1: A node with one input and three output links, where congestion in output links 1 and 3 only partially affects flow into output link 2, while congestion in link 2 affects flows into output links 1 and 3 in full, and output links 1 and 3 do not affect each other.

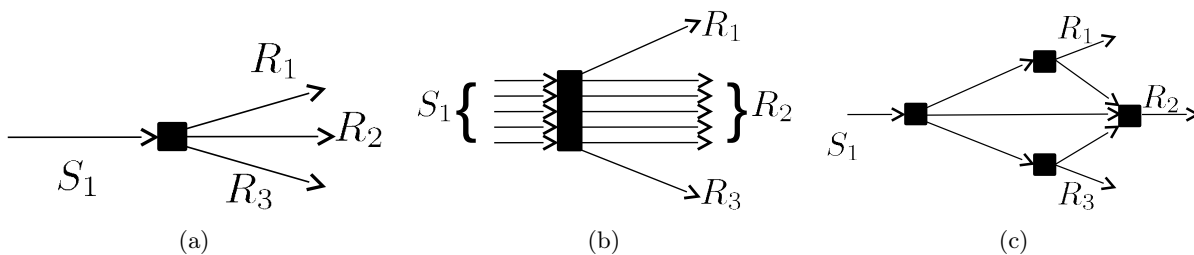


Figure 2: Three network representations of the junction in Figure 1.

Here, the FIFO constraint translates into

$$\frac{f_{1,1}}{S_{1,1}} = \frac{f_{1,2}}{S_{1,2}} = \frac{f_{1,3}}{S_{1,3}},$$

where $f_{i,j}(S_{i,j})$ is the flow (demand) from link i to j . This equality means that if, for instance, off-ramp 1 is jammed preventing traffic from coming in, there will be no flow of traffic to mainline 2 and off-ramp 3 either. This is in contrast to the fact that if congestion on off-ramp 1 spilled back onto the mainline, only one lane in Figure 1 would be blocked (with some knock-on effects in the adjacent lane if drivers were trying to enter the leftmost lane). More generally, FIFO imposes that a jam in any of the output links will block the flow to all output links. Transportation engineers find that for certain junctions the FIFO requirement is too strict to be realistic for this reason.

Some authors (e.g., Bliemer (2007) or Shiomi et al. (2015)) have suggested modeling each lane as a separate link to alleviate this problem, similar to the network in Figure 2b (in the Figure, the rectangular shape of the node signifies nothing, it is only stretched to fit all the input and output links). The demand S_1 and supply R_2 are split across the link lanes. This model resolves the unrealistic spillback problem: congestion in off-ramps 1 or 3 will only spill back to the links that are sending demand to an off-ramp; namely, the links representing the adjacent lanes. However, this approach has its own drawbacks. First and most obviously, it greatly complicates the size and dimensionality of the model, making every node in a two-or-more lane road a merge and diverge. Second, it invites the question of how to assign commodity traffic across link lanes, and provide a model for weaving or lane-change behavior, questions that are abstracted away in the “more macro” macroscopic model of Figure 2b.

Figure 2c shows an intermediate approach — the sending link S_1 in Figure 1 is split upstream of the off-ramps, at the leftmost node in Figure 2c. The top and bottom links exiting this leftmost node represent the left and right lanes in Figure 1, and carry all exiting vehicles, as well as the vehicles that take the left or right lanes but do not exit. The middle link then represents the middle three lanes. The non-exiting vehicles rejoin the same link at the rightmost node. This model limits the problems raised by separating each lane into a separate link, as in the network of Figure 2b. This approach, however, does not fully eliminate the unrealistic spillback problem, as heavy spillback that congests the “left lane” or “right lane” link will still spill back into link S_1 . Splitting the link apart earlier would give us more room to avoid the unrealistic spillback, but if we go too far upstream we can end up back at the network of Figure 2b.

At this point it may appear that we have found ourselves modeling between Scylla and Charybdis: keeping the network at manageable simplicity invites unrealistic spillback behavior, while preventing the overly aggressive spillback causes the network to explode in size. In Section 3 we present a method to overcome this dilemma, and will revisit this example in the context of our proposed solution in Section 3.2. Our method is relatively simple: it relies only on the lane geometry in junctions, as in Figure 1. It is also general and could be used with any sort of network configuration, for example, those of Figures 2b and 2c.

3 A node model for dimensionality management

Our notation for the node model problem has been introduced above. For a quick reference, we refer the reader to Appendix A.

This Section is organized as follows. First, we describe the merge problem — the multi-input-single-output (MISO) node in Section 3.1 to explain the concept of input link priorities. Then, we propose a way of relaxing the FIFO condition and explain it in the case of a single-input-multi-output (SIMO) node in Section 3.2. Finally, in Section 3.3 we proceed to the general multi-input-multi-output (MIMO) node putting all the concepts together.

3.1 Multiple-Input-Single-Output (MISO) Node: Explaining Input Priorities

We start by considering a node with M input links and 1 output link. The number of vehicles of type c that input link i wants to send is S_i^c . The flow entering from input link i has priority $p_i \geq 0$. Here $i = 1, \dots, M$ and $c = 1, \dots, C$. The output link can receive R_1 vehicles.

Let us briefly discuss the meaning of priorities. One intuitive way to understand them is that qualitatively speaking, p_i , the priority of link i , can be thought of the ability of link i to “claim” the supplies of the downstream links. If we consider the node problem as describing a scenario where, over some period of time, drivers from each incoming link “compete” to claim downstream supply, then an intuitive understanding of priorities is that p_i is proportional to the rate at which drivers are able to leave link i and claim downstream supply. We explore this idea further in a companion paper, Wright et al. (2016), where the node problems considered in this paper are explicitly recast as dynamic systems governed by ordinary differential equations. Note also that this “rate of leaving” characterization of priorities was mentioned by Tampère et al. (2011), in their justification of using capacities in place of priorities (their assumption being that vehicles will leave link i at a rate equal to its capacity). However, this characterization of priorities as flow rates is not meant to restrict the selection of priorities to flow rates. What is important is the ratio of a p_i to the other input links’ priorities. If, for example, a link 1 is to have twice the priority of a link 2, a priority set of $\{2000, 1000\}$ (i.e., on the scale of flow rates in terms of veh/hour) will produce the same results as a priority set of $\{2/3, 1/3\}$.

Casting the computation of input-output flows $f_{i,1}^c$ as a mathematical programming problem, we arrive at:

$$\max \left(\sum_{i=1}^M \sum_{c=1}^C f_{i,1}^c \right), \quad (3.1)$$

subject to:

$$f_{i,1}^c \geq 0, \quad i = 1, \dots, M, \quad c = 1, \dots, C \quad \text{— non-negativity constraint;} \quad (3.2)$$

$$f_{i,1}^c \leq S_i^c, \quad i = 1, \dots, M, \quad c = 1, \dots, C \quad \text{— demand constraint;} \quad (3.3)$$

$$\sum_{i=1}^M f_{i,1} \leq R_1 \quad \text{— supply constraint;} \quad (3.4)$$

$$\frac{f_{i,1}^c}{f_{i,1}} = \frac{S_i^c}{S_i}, \quad i = 1, \dots, M, \quad c = 1, \dots, C, \quad \text{— proportionality constraint for commodity flows;} \quad (3.5)$$

$$\left. \begin{array}{l} \text{(a)} \quad p_{i'} f_{i',1} = p_{i''} f_{i'',1} \quad \forall i', i'', \text{ such that } f_{i',1} < S_{i'}, f_{i'',1} < S_{i''}, \\ \text{(b)} \quad \text{If } f_{i,1} < S_i, \text{ then } f_{i,1} \geq \frac{p_i}{\sum_{i'=1}^M p_{i'}} R_1. \end{array} \right\} \quad \text{— priority constraint.} \quad (3.6)$$

where i' and i'' in (3.6) denote two generic input links. Constraint (3.5) defines how potential restrictions imposed on full input-output flows are translated to individual commodities. In this paper we assume that any restriction on the i -to-1 flow applies to commodity flows proportionally to their contribution to the demand, in accordance with requirement 8 introduced in Section 2.1.

Let us discuss the priority constraint (3.6) in more detail. It should be interpreted as follows. Input links $i = 1, \dots, M$, fall into two categories: (1) those whose flow is restricted by the allocated supply of the output link; and (2) those whose demand is satisfied by the supply allocated for them in the output link. Priorities define how supply in the output link is allocated for input flows. Condition (3.6)(a) says that flows from input links of category 1 are allocated proportionally to their priorities. Condition (3.6)(b) ensures that input flows of category 1 get *no less* than the portion of supply allocated for them based on their priorities. The inequality in condition (3.6)(b) becomes an equality when category 2 is empty.

There is a special case when some input link priorities are equal to 0. If there exists an input link \hat{i} such that $p_{\hat{i}} = 0$, while $f_{\hat{i},1} > 0$, then, due to condition (3.6)(a), all input links with nonzero priorities are in category

2. Thus, if category 1 contains only input links with zero priorities, one should evaluate condition (3.6)(a) with arbitrary positive, but *equal*, priorities: $p_{i'} = p_{i''} > 0$.

If the priorities p_i are proportional to the demands S_i , $i = 1, \dots, M$,¹ then condition (3.6) can be written as an equality constraint:

$$\frac{f_{1,1}}{S_1} = \dots = \frac{f_{i,1}}{S_i} = \dots = \frac{f_{M,1}}{S_M}, \quad (3.7)$$

and the optimization problem (3.1)-(3.6) turns into a *linear program* (LP).

Remark 3.1. *When selecting priorities p_i , they should be chosen such that they are not functions of the demands S_i . Otherwise, it can be shown that the resulting flows can violate the invariance principle (requirement 7 in Section 2.1) (Tampère et al., 2011). This is true for all priorities considered in this paper. For more detail on this, see the discussion in Section 2.1.3 of Tampère et al. (2011) and the references therein.*

Remark 3.2. *Note that constraint (3.7) cannot be trivially extended to the MIMO node by replacing subindex 1, where it denotes the output, with subindex j . Doing that, as is evident from Bliemer (2007), indeed leads to a convenient optimization problem formulation, but sacrifices the flow maximization objective of the node model, as was pointed out in Tampère et al. (2011), reducing the feasibility set more than necessary.*

For arbitrary priorities with $M = 2$, condition (3.6) becomes:

$$f_{1,1} \leq \max \left\{ \frac{p_1}{p_1 + p_2} R_1, R_1 - S_2 \right\}; \quad (3.8)$$

$$f_{2,1} \leq \max \left\{ \frac{p_2}{p_1 + p_2} R_1, R_1 - S_1 \right\}. \quad (3.9)$$

To give a hint how more complicated constraint (3.6) becomes as M increases, let us write it out for $M = 3$:

$$f_{1,1} \leq \max \left\{ \frac{p_1}{\sum_{i=1}^3 p_i} R_1, \frac{p_1}{p_1 + p_2} (R_1 - S_3), \frac{p_1}{p_1 + p_3} (R_1 - S_2), R_1 - \sum_{i=2,3} S_i \right\}; \quad (3.10)$$

$$f_{2,1} \leq \max \left\{ \frac{p_2}{\sum_{i=1}^3 p_i} R_1, \frac{p_2}{p_1 + p_2} (R_1 - S_3), \frac{p_2}{p_2 + p_3} (R_1 - S_1), R_1 - \sum_{i=1,3} S_i \right\}; \quad (3.11)$$

$$f_{3,1} \leq \max \left\{ \frac{p_3}{\sum_{i=1}^3 p_i} R_1, \frac{p_3}{p_1 + p_3} (R_1 - S_2), \frac{p_3}{p_2 + p_3} (R_1 - S_1), R_1 - \sum_{i=1,2} S_i \right\}. \quad (3.12)$$

As we can see, right hand sides of inequalities (3.10)-(3.12) contain known quantities, and so for arbitrary priorities, problem (3.1)-(3.6) is also an LP. For general M , however, building constraint (3.6) requires a somewhat involved algorithm. Instead, we develop an algorithm for computing input-output flows f_{i1}^c that solves the maximization problem (3.1)-(3.6). For the special case of the MISO node, our solution algorithm is similar to the solution algorithm of Tampère et al. (2011) (again, with a generalization of priorities to be arbitrary quantities rather than capacities), so in the interest of brevity we defer its statement to Appendix B.

The following theorem summarizes this section. The proof will follow in Section 3.3, where we prove a more general statement that has this problem as a special case.

Theorem 3.1. *The input-output flow computation algorithm of Appendix B constructs the unique solution of the maximization problem (3.1)-(3.6).*

¹Here we assume that $S_i > 0$, $i = 1, \dots, M$.

3.2 Single-Input-Multiple-Output (SIMO) Node: Relaxing the FIFO Condition

Recall our discussion of Figures 1 and 2, and the unrealistic spillback problems created by the FIFO condition. A more realistic model would allow to specify how output flows at any given junction can affect each other. In the context of our example, we could say that congestion in off-ramp 1 affects mainline flow in lane 1, congestion in off-ramp 3 affects mainline flow in lanes 4 and 5 (see striped areas in Figure 1), while traffic flow in lanes 2 and 3 of the mainline output link is not affected by the traffic state in output links 1 and 3. Additionally, we can say that congestion in the mainline output 2 affects flows directed to both off-ramps in full, while the traffic state in each of the off-ramps does not influence the flow directed to the other off-ramp.

For purposes that will become clear in a moment, we refer to this partial blocking in terms of intervals of the lane group (e.g., the leftmost fifth of the lane group), rather than specific lanes (e.g., lane 1). For example, referring again to Figure 1, off-ramp 1 becoming congested would affect the leftmost fifth of the mainline flow, and off-ramp 3 becoming congested would affect the rightmost two-fifths of the mainline flow. Further, if we parameterize the five lanes serving the mainline flow as a unit interval, $[0, 1]$, then we could say that off-ramp 1 becoming congested affects the interval $[0, \frac{1}{5}]$, and off-ramp 3 becoming congested affects the interval $[\frac{3}{5}, 1]$.

To formally describe this, we introduce *mutual restriction intervals* $\boldsymbol{\eta}_{j',j} = [y, z] \subseteq [0, 1]$. These parameters can be interpreted as follows:

- $\boldsymbol{\eta}_{j',j} = [0, 1]$ — congestion in the output link j' affects flow directed to the output link j in full. This is equivalent to the FIFO condition. Obviously, $\boldsymbol{\eta}_{j,j} \equiv [0, 1]$.
- $\boldsymbol{\eta}_{j',j} = [0, 0]$ (or any other interval of zero length) — traffic state in the output link j' does not influence the flow directed to the output link j . This is equivalent to no FIFO restriction.
- $\boldsymbol{\eta}_{j',j} = [y, z] \subset [0, 1]$ — traffic state in the output link j' affects a $|\boldsymbol{\eta}_{j',j}| = z - y$ portion of the flow directed to the output link j . Moreover, we specify this influence as an interval, not just a scalar, to capture the summary effect of multiple output links that may restrict flow to the output link j — how, will be explained shortly.

Recall again the node in Figure 1. Here, the mutual restriction intervals, in matrix form, may be written as

$$\boldsymbol{\eta} = \begin{pmatrix} [0, 1] & [0, \frac{1}{5}] & [0, 0] \\ [0, 1] & [0, 1] & [0, 1] \\ [0, 0] & [\frac{3}{5}, 1] & [0, 1] \end{pmatrix},$$

where $\boldsymbol{\eta}_{j',j}$ is the j', j element of the matrix. The diagonal ($j = j'$) elements of the matrix, $\boldsymbol{\eta}_{j,j}$, are always $[0, 1]$ because they indicate that a movement becomes fully restricted when its destination link is blocked. Note that the matrix is not symmetric in general, since a queue in link j will typically not affect the same portion $[y, z]$ of j' -serving lanes as a queue in link j' will affect the j -serving lanes (for example, $\boldsymbol{\eta}_{2,1} = [0, 1]$ in the above matrix because a queue in all five freeway lanes implies a queue in the leftmost freeway lane that serves offramp 1).

We have established how we encode the spatial (i.e., portion of lanes) extent of restriction into the restriction intervals. Now, we discuss how we use the restriction intervals to compute flows. Recalling our example, suppose $\frac{R_1}{S_{1,1}} < \frac{R_3}{S_{1,3}} < 1 \leq \frac{R_2}{S_{1,2}}$. In other words, demand for output links 1 and 3 exceeds the available supply with output link 1 being more restrictive, while the demand directed to the output link 2 can be satisfied. Since output links 1 and 3 do not affect each other, we get

$$f_{1,1} = R_1 \quad \text{and} \quad f_{1,3} = R_3.$$

Flow $f_{1,2}$ is partially restricted by both output links 1 and 3:

$$f_{1,2} = (1 - |\boldsymbol{\eta}_{1,2}| - |\boldsymbol{\eta}_{3,2} \setminus (\boldsymbol{\eta}_{3,2} \cap \boldsymbol{\eta}_{1,2})|) S_{1,2} + \frac{R_1}{S_{1,1}} |\boldsymbol{\eta}_{1,2}| S_{1,2} + \frac{R_3}{S_{1,3}} |\boldsymbol{\eta}_{3,2} \setminus (\boldsymbol{\eta}_{3,2} \cap \boldsymbol{\eta}_{1,2})| S_{1,2},$$

where $|\boldsymbol{\eta}_{1,2}|$ and $|\boldsymbol{\eta}_{3,2} \setminus (\boldsymbol{\eta}_{3,2} \cap \boldsymbol{\eta}_{1,2})|$ denote the lengths of intervals $\boldsymbol{\eta}_{1,2}$ and $\boldsymbol{\eta}_{3,2} \setminus (\boldsymbol{\eta}_{3,2} \cap \boldsymbol{\eta}_{1,2})$, respectively. In the second interval, the intersection of $\boldsymbol{\eta}_{3,2}$ and $\boldsymbol{\eta}_{1,2}$ is removed because, as output link 1 is more restrictive by assumption, the output 3 will restrict $f_{1,2}$ only when output 1 is already restricting $f_{1,2}$ (of course, in this example, the intersection is the empty set, but as we will see momentarily, that is not always the case). In this expression for $f_{1,2}$, the first term represents the unrestricted portion of flow (lanes 2 and 3); the second term represents the portion of flow restricted by the output 1 (lane 1); and the third term represents the portion of flow restricted by the output 3 (lanes 4 and 5). The expression for $f_{1,2}$ can be rewritten as

$$f_{1,2} = S_{1,2} - \left(1 - \frac{R_1}{S_{1,1}}\right) |\boldsymbol{\eta}_{1,2}| S_{1,2} - \left(1 - \frac{R_3}{S_{1,3}}\right) |\boldsymbol{\eta}_{3,2} \setminus (\boldsymbol{\eta}_{3,2} \cap \boldsymbol{\eta}_{1,2})| S_{1,2}. \quad (3.13)$$

We believe (3.13) is an intuitive representation of how the mutual restriction intervals affect flows. The first term, $S_{1,2}$, is the maximum possible flow (i.e., the demand). The second and third terms represent the portion of demand that cannot be fulfilled due to the restriction intervals. For both of them, $(1 - R_j/S_{1,j})$ represents the portion of flow that is affected by output link j becoming congested, and $|\cdot|$, the length of the mutual restriction interval, is the degree to which this flow portion is affected by relaxed, or partial, FIFO.

Further, the computation in (3.13) can be intuitively represented in the two-dimensional graphics of Figure 3. In both subfigures, the area of the striped region is the flow $f_{1,2}$. The entire rectangle, consisting of the region $[0, S_{1,2}]$ on the horizontal axis and $0, 1$ on the vertical axis, has the area $S_{1,2}$ and represents the maximum possible flow. The shaded regions correspond to the flow reductions in (3.13).

Focusing first on Figure 3(a), the top shaded region represents the flow reduction caused by congestion in output link 1. One can see that its extent on the vertical axis is $\boldsymbol{\eta}_{1,2} = [0, \frac{1}{5}]$, and its extent on the horizontal axis is $(1 - R_1/S_{1,1})S_{1,2} = (f_{1,1}/S_{1,1})S_{1,2}$. The area of this shape, then, is $(1 - R_1/S_{1,1})|\boldsymbol{\eta}_{1,2}|S_{1,2}$, which appears in (3.13). Computing (3.13), then, is equivalent to measuring the shaded area in Figure 3(a).

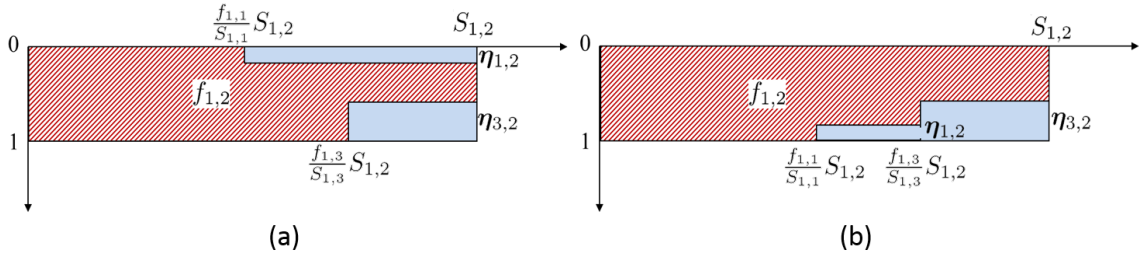


Figure 3: Computing $f_{1,2}$ for the 1-input-3-output node example with $\frac{R_1}{S_{1,1}} < \frac{R_3}{S_{1,3}} < 1 \leq \frac{R_2}{S_{1,2}}$, $\boldsymbol{\eta}_{13} = \boldsymbol{\eta}_{31} = [0, 0]$, $\boldsymbol{\eta}_{32} = [\frac{3}{5}, 1]$ and two cases for $\boldsymbol{\eta}_{1,2}$: (a) $\boldsymbol{\eta}_{1,2} = [0, \frac{1}{5}]$; (b) $\boldsymbol{\eta}_{1,2} = [\frac{4}{5}, 1]$.

Now, considering both Figures 3(a) and 3(b), we see why we subtracted the intersection in the third term of (3.13). In Figure 3(b), we have $\boldsymbol{\eta}_{1,2} = [\frac{4}{5}, 1]$. Here, restrictions from outputs 1 and 3 imposed on flow $f_{1,2}$ would overlap in their affected lanes. The intuition is that if traffic in lane 5 of the mainline output is already restricted by the output 1, then the output 3 cannot do anything more to restrict the flow in lane 5, it can only restrict flow in lane 4 of the mainline output. Since the intersection in Figure 3(a)'s problem is the empty set, (3.13) gives the area of its shaded area as well; (3.13) is the proper equation for both cases.

We now generalize from two restriction intervals to an arbitrary number. Define:

$$\mathcal{Q}_{j',j} = \boldsymbol{\eta}_{j',j} \times \left[\frac{f_{1,j'}}{S_{1,j'}} S_{1,j}, S_{1,j} \right], \quad (3.14)$$

where ‘ \times ’ denotes a Cartesian product. The Cartesian product of the two intervals gives us a rectangle $\mathcal{Q}_{j',j}$. For example, in Figure 3, $\mathcal{Q}_{1,2}$ and $\mathcal{Q}_{3,2}$ represent the grayed-out rectangles.

Denote $\mathcal{A}(\cdot)$ as the area of a two-dimensional shape. For example, the expression for flow $f_{1,2}$ in our example, (3.13), can be replaced with a single, more general formula:

$$f_{1,2} = S_{1,2} - \mathcal{A}(\mathcal{Q}_{1,2} \cup \mathcal{Q}_{3,2}).$$

Now, we are ready to formulate the optimization problem for the general SIMO node with N output links and C commodities:

$$\max \left(\sum_{j=1}^N \sum_{c=1}^C f_{1,j}^c \right), \quad (3.15)$$

subject to:

$$f_{1,j}^c \geq 0, \quad j = 1, \dots, N, \quad c = 1, \dots, C \quad \text{— non-negativity constraint;} \quad (3.16)$$

$$f_{1,j}^c \leq S_{1,j}^c, \quad j = 1, \dots, N, \quad c = 1, \dots, C \quad \text{— demand constraint;} \quad (3.17)$$

$$f_{1,j} \leq R_j, \quad j = 1, \dots, N \quad \text{— supply constraint;} \quad (3.18)$$

$$\frac{f_{i,j}^c}{f_{1,j}} = \frac{S_{1,j}^c}{S_{1,j}}, \quad j = 1, \dots, N, \quad c = 1, \dots, C \quad \text{— proportionality constraint for commodity flows;} \quad (3.19)$$

$$f_{1,j} \leq S_{1,j} - \mathcal{A} \left(\bigcup_{j' \neq j} \mathcal{Q}_{j',j} \right), \quad j = 1, \dots, N \quad \text{— relaxed FIFO constraint.} \quad (3.20)$$

For SIMO nodes with full FIFO, constraint (3.20) together with the supply constraint (3.18) translates into

$$f_{1,j} \leq S_{1,j} - \left(1 - \frac{f_{1,j^*}}{S_{1,j^*}} \right) S_{1,j} = \frac{f_{1,j^*}}{S_{1,j^*}} S_{1,j}, \quad (3.21)$$

where

$$j^* = \arg \min_j \frac{R_j}{S_{1,j}}, \quad (3.22)$$

and, since we are solving the flow maximization problem, (3.21) can be replaced with the equality constraint:

$$f_{1,j} = \frac{f_{1,j^*}}{S_{1,j^*}} S_{1,j}. \quad (3.23)$$

For SIMO nodes with no FIFO, $\mathcal{A} \left(\bigcup_{j' \neq j} \mathcal{Q}_{j',j} \right) = 0$, which simplifies (3.20) to the demand constraint, and thus, constraint (3.20) can be omitted in that case.

Next, we present the algorithm for solving the flow maximization problem (3.15)-(3.20).

1. Initialize:

$$\begin{aligned} \tilde{S}_{1,j}^c(0) &:= S_{1,j}^c; \\ \tilde{S}_{1,j}(0) &:= S_{1,j}; \\ \tilde{\eta}_j(0) &= [0, 0]; \\ V(0) &:= \{1, \dots, N\}; \\ k &:= 0; \\ & \quad j = 1, \dots, N, \quad c = 1, \dots, C. \end{aligned}$$

As before, k denotes the iteration index; $\tilde{S}_{1,j}^c(k)$ ($\tilde{S}_{1,j}(k)$) is the oriented demand per commodity (total across commodities) at iteration k ; $\tilde{\eta}_j(k)$ is the portion of output flow to j affected by the restricted supply of other output links at iteration k , and is the union of intervals that have become active as of iteration k ; and $V(k)$ is the set of output links still to be processed at iteration k .

2. If $V(k) = \emptyset$, stop.
3. For all output links $j \in V(k)$, find flow reduction factors:

$$\alpha_j(k) = \frac{R_j}{\tilde{S}_{1,j}(k)}, \quad (3.24)$$

and find the most restrictive output link out of the remaining ones:

$$j^*(k) = \arg \min_{j \in V(k)} \alpha_j(k). \quad (3.25)$$

- If $\alpha_{j^*(k)}(k) \geq 1$, then assign:

$$\begin{aligned} f_{1,j}^c &= \tilde{S}_{1,j}^c(k), \quad \forall j \in V(k), \quad c = 1, \dots, C; \\ V(k+1) &= \emptyset. \end{aligned} \quad (3.26)$$

- Else, assign:

$$\tilde{S}_{1,j^*}^c(k+1) = \alpha_{j^*}(k) \tilde{S}_{1,j^*}^c(k), \quad c = 1, \dots, C; \quad (3.27)$$

$$\tilde{S}_{1,j}^c(k+1) = \frac{\tilde{S}_{1,j}(k+1)}{S_{1,j}} S_{1,j}^c, \quad j \in V(k) \setminus \{j^*\}, \quad c = 1, \dots, C, \quad (3.28)$$

where

$$\tilde{S}_{1,j}(k+1) = \tilde{S}_{1,j}(k) - S_{1,j} (|\boldsymbol{\eta}_{j^*,j}| - |\tilde{\eta}_j(k) \cap \boldsymbol{\eta}_{j^*,j}|) \left(1 - \frac{\sum_{c=1}^C \tilde{S}_{1,j^*}^c(k+1)}{S_{1,j^*}} \right), \quad (3.29)$$

$$\begin{aligned} \tilde{\eta}_j(k+1) &= \tilde{\eta}_j(k) \cup \boldsymbol{\eta}_{j^*,j}, \quad j \in V(k); \\ f_{1,j}^c &= \tilde{S}_{1,j}^c(k+1), \quad j : \tilde{\eta}_j(k+1) = [0, 1], \quad c = 1, \dots, C; \\ V(k+1) &= V(k) \setminus \{j : \tilde{\eta}_j(k+1) = [0, 1]\}, \end{aligned} \quad (3.30)$$

where $|\tilde{\eta}_j(k) \cap \boldsymbol{\eta}_{j^*,j}|$ denotes the measure of the interval intersection.²

4. Set $k := k + 1$ and return to step 2.

This algorithm takes no more than N iterations to complete.

The algorithm can be understood as follows. $V(k)$ is the set of output links who have not had their incoming flows determined as of iteration k . In (3.24) and (3.25), the output link whose remaining supply is most demanded (that is, whose ratio of remaining supply to total incoming demand is lowest) is identified and labeled $j^*(k)$. If this most-demanded link is actually able to handle all of its demand, then we reach (3.26) and all node flows are in freeflow. Otherwise, we reach (3.27)-(3.30). Equation (3.27) assigns the flow to the most-restricted output link $j^*(k)$ (this flow will be R_{j^*} , which can be seen by examining (3.24)). Equations (3.28) and (3.29) perform the application of the mutual restriction interval for the just-filled link j^* to all other output links j in a two-step process. The quantity $\tilde{S}_{1,j}^c(k+1)$ is a “running demand” for commodity c to link j after relaxed FIFO has been applied to the original demand $S_{1,j}^c$.

This “running demand” that takes the activation of mutual restriction intervals is calculated in (3.29). Note that this calculation is exactly the calculation of a shaded area in the diagrams of Figure 3. In particular, $|\boldsymbol{\eta}_{j^*,j}| - |\tilde{\eta}_j(k) \cap \boldsymbol{\eta}_{j^*,j}|$ defines the height of the union of a just-activated mutual restriction interval $\boldsymbol{\eta}_{j^*,j}$ with the current active mutual restriction interval(s) $\tilde{\eta}_j(k)$, and $1 - \left(\sum_{c=1}^C \tilde{S}_{1,j^*}^c(k+1) \right) / (S_{1,j^*})$ defines the

²This set may be disjoint.

width of the new rectangle. Equation (3.28) simply scales each commodity's demand by the same "blockage" represented by the rectangle. Similar to $\tilde{S}_{1,j}^c(k+1)$, $\tilde{\eta}_j(k+1)$ is a "running mutual restriction interval" that represents the union of mutual restriction intervals acting on flow to j up to this point. When link j has a running mutual restriction interval of $[0, 1]$, it has been fully blocked, and so its flow is assigned by its running demand in (3.30). This construction of the heights and widths of the rectangles will also be used in the multi-input-multi-output case in the next Section.

Finally, we state the result of this Section as a theorem. As with Theorem 3.1, the proof is deferred to Section 3.3.

Theorem 3.2. *The SIMO input-output flow computation algorithm constructs the unique solution of the maximization problem (3.15)-(3.20).*

3.3 Multiple-Input-Multiple-Output (MIMO) Node

To formally state the flow maximization problem for the MIMO node, we need to extend the MISO optimization problem (3.1)-(3.6) and the SIMO optimization problem (3.15)-(3.20). While the generalization of the objective (3.1), (3.15) and constraints (3.2)-(3.5), (3.16)-(3.19) is straightforward, extending the priority constraint (3.6) and the relaxed FIFO constraint (3.20) to the MIMO case requires a little more work.

First, we introduce the concept of oriented priorities:

$$p_{i,j} = p_i \frac{\sum_{c=1}^C S_{i,j}^c}{\sum_{c=1}^C S_i^c}, \quad i = 1, \dots, M, \quad j = 1, \dots, N. \quad (3.31)$$

Recall our discussion about the physical meaning of priorities in Section 3.1. Qualitatively, priorities mean that drivers from each link i "compete" to claim supply at rate p_i . The oriented priority $p_{i,j}$ can be thought of as the rate at which drivers from a link i claim supply from the *particular* link j . This means that the ability of vehicles from i to claim j 's supply is proportional to the portion of i vehicles that are actually trying to claim j 's supply.

Before we present the algorithm, we need to define two items.

Definition 3.1. *If, for a given input link i , whose demand cannot be satisfied ($\sum_{j=1}^N f_{i,j} < S_i$), there exists at least one output link j^* , such that: (1) $S_{i,j^*} > 0$; and (2) $p_{i',j^*} f_{i',j^*} \geq p_{i,j^*} f_{i,j^*}$ for any $i' \neq i$, we say that such output j^* is restricting for input i .*

We need this definition to formulate the priority constraint, and it is only valid for $M > 1$. If the output j^* is restricting for input links i' and i'' , then, according to this definition, $p_{i'',j^*} f_{i'',j^*} = p_{i',j^*} f_{i',j^*}$.

Definition 3.2. *For a given input link i , define the set of restricting output links (as defined by Definition 3.1) for that i as*

$$W_i \triangleq \left\{ j^* : R_{j^*} > S_{i,j^*} > 0, \exists i' \neq i \text{ s.t. } p_{i',j^*} f_{i',j^*} \geq p_{i,j^*} f_{i,j^*} \right\}.$$

Now, we can formulate the flow maximization problem for the general MIMO node:

$$\max \left(\sum_{i=1}^M \sum_{j=1}^N \sum_{c=1}^C f_{i,j}^c \right), \quad (3.32)$$

subject to:

$$f_{i,j}^c \geq 0, \quad i = 1, \dots, M, \quad j = 1, \dots, N, \quad c = 1, \dots, C \quad \text{— non-negativity constraint;} \quad (3.33)$$

$$f_{i,j}^c \leq S_{i,j}^c, \quad i = 1, \dots, M, \quad j = 1, \dots, N, \quad c = 1, \dots, C \quad \text{— demand constraint;} \quad (3.34)$$

$$\sum_{i=1}^M f_{i,j} \leq R_j, \quad j = 1, \dots, N \quad \text{— supply constraint;} \quad (3.35)$$

$$\frac{f_{i,j}^c}{f_{i,j}} = \frac{S_{i,j}^c}{S_{i,j}}, \quad i = 1, \dots, M, \quad j = 1, \dots, N, \quad c = 1, \dots, C \quad \text{— proportionality constraint}$$

for commodity flows; (3.36)

$$\left. \begin{array}{l} \text{(a) For each input link } i \text{ such that} \\ \quad \sum_{j=1}^N f_{i,j} < S_i, \quad W_i \neq \emptyset; \\ \text{(b) For each input link } i \text{ such that } W_i \neq \emptyset, \\ \quad f_{i,j} \geq \frac{p_{i,j}}{\sum_{i'=1}^M p_{i',j}} R_j, \quad \forall j \in W_i, \end{array} \right\} \text{— priority constraint;} \quad (3.37)$$

$$f_{i,j} \leq S_{i,j} - \mathcal{A} \left(\bigcup_{j' \in W_i \setminus \{j\}} \mathcal{Q}_{j',j}^i \right), \quad i = 1, \dots, M, \quad j = 1, \dots, N \quad \text{— relaxed FIFO}$$

constraint. (3.38)

Constraint (3.37) generalizes the MISO priority constraint (3.6), with W_i being the set of restricting outputs for input link i as defined by Definition 3.2. For each output link j , $j = 1, \dots, N$, flows $\sum_{c=1}^C f_{i,j}^c$ fall into two categories: (1) restricted by *this* output link; and (2) not restricted by *this* output link. Condition (3.37)(a) states that if the flow from an input i is supply-constrained, there do exist output links j , such that flows $f_{i,j}$ are of category 1; and for a given output j , input-output flows of category 1 are allocated proportionally to their priorities. For every output link j , $j = 1, \dots, N$, condition (3.37)(b) says that these category 1 flows may take “leftover” supply after the category 2 flows into j have been resolved. For a SIMO ($M = 1$) node, there is no competition between input flows, and constraint (3.37) is satisfied automatically.

Constraint (3.38) generalizes the SIMO relaxed FIFO constraint (3.20). Here,

$$\mathcal{Q}_{j',j}^i = \eta_{j',j}^i \times \left[\frac{f_{i,j'}}{S_{i,j'}} S_{i,j}, S_{i,j} \right], \quad (3.39)$$

which is a generalization of (3.14), and $\mathcal{A}(\cdot)$ denotes the area of a two-dimensional object. For MIMO nodes with full FIFO, constraint (3.38) together with the supply constraint (3.35) translates to:

$$f_{i,j} \leq \min_{j' \in W_i} \left\{ \frac{f_{i,j'}}{S_{i,j'}} \right\} S_{i,j}, \quad (3.40)$$

and, since we are solving the flow maximization problem, (3.40) can be replaced with the equality constraint:

$$f_{i,j} = \min_{j^* \in W_i} \frac{f_{i,j^*}}{S_{i,j^*}} S_{i,j}. \quad (3.41)$$

For MIMO nodes with no FIFO and for MISO ($N = 1$) nodes, $\mathcal{A} \left(\bigcup_{j' \in W_i \setminus \{j\}} \mathcal{Q}_{j',j}^i \right) = 0$, and thus, constraint (3.38) degenerates into the demand constraint.

Remark 3.3. Note that in the case of multiple input links mutual restriction intervals are to be specified per input link. This can be justified by the following example. In the node representing a junction with 2 input and 3 output links, shown in Figure 4, consider the influence of the output link 5 on the output link 4. If vehicles enter links 4 and 5 from link 1, then it is reasonable to assume that once link 5 is jammed

and cannot accept any vehicles, there is no flow from 1 to 4 either, since vehicles queueing for a U-turn into link 4 would use the same lanes as those vehicles queueing for a left turn into link 5. In other words, $\eta_{5,4}^1 = [0, 1]$. On the other hand, if vehicles arrive from link 2, blockage of the output link 5 may hinder, but not necessarily prevent, traffic from flowing into the output link 4, and so $\eta_{5,4}^2 = [1 - \epsilon, 1]$, with some $0 < \epsilon < 1$ that depends on the portion of 2's of lanes that allow both a right-turn and through movement. So, from now on we will write $\eta_{j',j}^i$ and $\mathcal{Q}_{j',j}^i$ with index i identifying the input link. Example 1 in Section 4 discusses a situation like this in detail.

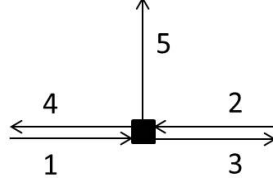


Figure 4: Intersection node.

Next, we present the algorithm that solves the flow maximization problem (3.32)-(3.38).

1. Initialize:

$$\begin{aligned}
\tilde{R}_j(0) &:= R; \\
U_j(0) &:= \left\{ i \in \{1, \dots, M\} : \sum_{c=1}^C S_{i,j}^c > 0 \right\}; \\
\tilde{S}_{i,j}^c(0) &:= S_{i,j}^c; \\
\tilde{S}_{i,j}(0) &:= \sum_{c=1}^C \tilde{S}_{i,j}^c(0); \\
\tilde{\eta}_j^i(0) &:= [0, 0]; \\
k &:= 0; \\
& \quad i = 1, \dots, M, \quad j = 1, \dots, N, \quad c = 1, \dots, C.
\end{aligned}$$

As before, k denotes iteration; $\tilde{R}_j(k)$ is the remaining supply of output link j at iteration k ; $U_j(k)$ is the set of input links i that have nonzero demands towards j (that is, at least one nonzero $S_{i,j}^c$) and have not had their flows assigned as of iteration k ; $\tilde{S}_{1,j}^c(k)$ ($\tilde{S}_{1,j}(k)$) is the oriented demand per commodity (total) at iteration k ; $\tilde{\eta}_j^i(k)$ is the portion of input- i -to-output- j flow affected by the restricted supply of other output links at iteration k and is the union of intervals affecting that particular flow that have become active as of iteration k .

2. Define the set of output links that still need processing:

$$V(k) = \{j : U_j(k) \neq \emptyset\}.$$

If $V(k) = \emptyset$, stop.

3. Check that at least one of the unprocessed input links has nonzero priority, otherwise, assign equal positive priorities to all the unprocessed input links:

$$\tilde{p}_i(k) = \begin{cases} p_i, & \text{if there exists } i' \in \bigcup_{j \in V(k)} U_j(k) : p_{i'} > 0, \\ \frac{1}{|\bigcup_{j \in V(k)} U_j(k)|}, & \text{otherwise,} \end{cases} \quad (3.42)$$

where $\left| \bigcup_{j \in V(k)} U_j(k) \right|$ denotes the number of elements in the union $\bigcup_{j \in V(k)} U_j(k)$; and for each output link $j \in V(k)$ and input link $i \in U_j(k)$ compute oriented priority:

$$\tilde{p}_{i,j}(k) = \tilde{p}_i(k) \frac{\sum_{c=1}^C S_{i,j}^c}{\sum_{c=1}^C S_i^c}. \quad (3.43)$$

4. For each $j \in V(k)$, compute factors:

$$a_j(k) = \frac{\tilde{R}_j(k)}{\sum_{i \in U_j(k)} \tilde{p}_{i,j}(k)}, \quad (3.44)$$

and find the smallest of these factors:

$$a_{j^*}(k) = \min_{j \in V(k)} a_j(k). \quad (3.45)$$

These factors $a_j(k)$ describe the ratio of the supply $R_j(k)$ to the demand placed upon j by way of the oriented priorities $\tilde{p}_{i,j}(k)$ ³. The link j^* has the most demanded supply of all output links.

5. Define the set of input links whose demand does not exceed the allocated supply:

$$\tilde{U}(k) = \left\{ i \in U_{j^*}(k) : \left(\sum_{j \in V(k)} \tilde{S}_{i,j}(k) \right) \leq \tilde{p}_i(k) a_{j^*}(k) \right\}.$$

- If $\tilde{U}(k) \neq \emptyset$, then for all output links $j \in V(k)$ assign:

$$\begin{aligned} f_{i,j}^c &= \tilde{S}_{i,j}^c(k), \quad i \in \tilde{U}(k), \quad c = 1, \dots, C; \\ \tilde{R}_j(k+1) &= \tilde{R}_j(k) - \sum_{i \in \tilde{U}(k)} f_{i,j}; \\ U_j(k+1) &= U_j(k) \setminus \tilde{U}(k). \end{aligned} \quad (3.46)$$

- Else, for all input links $i \in U_{j^*}(k)$, output links $j \in V(k)$ and commodities $c = 1, \dots, C$, assign:

$$\tilde{S}_{i,j^*}^c(k+1) = \tilde{S}_{i,j^*}^c(k) \frac{\tilde{p}_{i,j^*}(k) a_{j^*}(k)}{\tilde{S}_{i,j^*}(k)}, \quad c = 1, \dots, C; \quad (3.47)$$

$$\tilde{S}_{i,j}^c(k+1) = \tilde{S}_{i,j}^c(k) \frac{\tilde{S}_{i,j}(k+1)}{\tilde{S}_{i,j}(k)}, \quad i \in U_j(k) \cap U_{j^*}(k), \quad j \in V(k) \setminus \{j^*\}, \quad (3.48)$$

where

$$\tilde{S}_{i,j}(k+1) = \tilde{S}_{i,j}(k) - S_{i,j} (|\boldsymbol{\eta}_{j^*,j}^i| - |\tilde{\boldsymbol{\eta}}_j^i(k) \cap \boldsymbol{\eta}_{j^*,j}^i|) \left(1 - \frac{\sum_{c=1}^C \tilde{S}_{i,j^*}^c(k+1)}{S_{i,j^*}} \right); \quad (3.49)$$

$$\tilde{S}_{i,j}^c(k+1) = \tilde{S}_{i,j}^c(k), \quad i \notin U_j(k) \cap U_{j^*}(k);$$

$$\tilde{\boldsymbol{\eta}}_j^i(k+1) = \tilde{\boldsymbol{\eta}}_j^i(k) \cup \boldsymbol{\eta}_{j^*,j}^i;$$

$$f_{i,j}^c = \tilde{S}_{i,j}^c(k+1),$$

$$i \in U_j(k) \cap U_{j^*}(k), \quad j \in V(k) : \tilde{\boldsymbol{\eta}}_j^i(k+1) = [0, 1], \quad c = 1, \dots, C; \quad (3.50)$$

$$\tilde{R}_j(k+1) = \tilde{R}_j(k) - \sum_{i \in U_{j^*}(k) : \boldsymbol{\eta}_j^i(k+1) = [0, 1]} f_{i,j};$$

$$U_j(k+1) = U_j(k) \setminus \{i \in U_{j^*}(k) : \tilde{\boldsymbol{\eta}}_j^i(k+1) = [0, 1]\}.$$

³ If, as before, we consider oriented priorities as describing the “rate” at which supply is claimed, then $a_j(k) < a_{j'}(k)$ means that $R_j(k)$, which is being filled at rate $\sum_{i \in U_j(k)} \tilde{p}_{i,j}(k)$, is filled “before” $R_{j'}(k)$, which is being filled at rate $\sum_{i \in U_{j'}(k)} \tilde{p}_{i,j'}(k)$ (see Wright et al. (2016) for more on this).

6. Set $k := k + 1$, and return to step 2.

This algorithm takes no more than $M + N$ iterations to complete.

This algorithm can be best understood as a generalization of the algorithm for solving the SIMO node problem (Section 3.2). Similar to the SIMO problem, $V(k)$ is the set of output links whose input flows are not yet determined as of iteration k . At each iteration k , $j^* \in V(k)$ is the output whose supply is most-demanded by its oriented priorities. In step 5, it is determined whether any of the links that still want to send demand to j^* (i.e., the $i \in U_{j^*}(k)$) will be able to send all their vehicles before this most-demanded link j^* 's supply is exhausted. If so, then $\tilde{U}(k)$ is nonempty, the first case of step 5 is entered, and these ‘‘lucky’’ links $\tilde{U}(k)$ are able to satisfy all their demand in (3.46).

If not, then all links $i \in U_{j^*}(k)$ will still have leftover demand after j^* is filled, and we enter the second case of step 5. In (3.47), the demands into j^* are scaled so that when their flows are assigned in (3.50), each $i \in U_{j^*}(k)$ will fill its priority-proportional share of $\tilde{R}_{j^*}(k)$ (see the relevant calculation in the proof of Theorem 3.3 to check this). In (3.48) and (3.49), relaxed FIFO is enforced on all flows into *different* $j \neq j^*$ in accordance with their mutual restriction intervals $\eta_{j^*,j}^i$ (this is the same procedure as in (3.28) and (3.29) in the SIMO case). Finally, (3.50) sets all flows that can be found at this point, due to them being constrained by a FIFO constraint in the form of a running mutual restriction interval of $[0, 1]$ (note that this includes the flows into j^* , as $\eta_{j^*,j^*}^i = [0, 1] \forall i$ by definition).

The following lemma states that in the case of $N = 1$, the MIMO algorithm produces the same result as the MISO algorithm described in Section 3.1.

Lemma 3.1. *The MISO algorithm is a special case of the MIMO algorithm with $N = 1$.*

Proof. The proof follows from the fact that for $N = 1$, formulae (3.44)-(3.45) result in $a_{j^*}(k) = a_1(k) = \frac{\tilde{R}_1(k)}{\sum_{i \in U_1(k)} \tilde{p}_i(k)}$ and $j^* = 1$. \square

The following lemma states that in the case $M = 1$, the MIMO algorithm with relaxed FIFO condition produces the same result as the SIMO algorithm with relaxed FIFO condition described in Section 3.2.

Lemma 3.2. *The SIMO algorithm with the relaxed FIFO condition is a special case of the MIMO algorithm with $M = 1$.*

Proof. The proof follows from the fact that for $M = 1$, factors $a_j(k)$, defined in (3.44), reduce to:

$$a_j(k) = \frac{R_j \tilde{S}_1(k)}{\tilde{S}_{1,j}(k)},$$

and

$$j^* = \arg \min_{j \in V(k)} \frac{R_j \tilde{S}_1(k)}{\tilde{S}_{1,j}(k)} = \arg \min_{j \in V(k)} \frac{R_j}{\tilde{S}_{1,j}(k)}. \quad \square$$

The main result of this Section can be stated as the following theorem.

Theorem 3.3. *Given a set of input links $i \in 1, \dots, M$, output links $j \in 1, \dots, N$, commodities $c \in 1, \dots, C$, priorities $\{p_i\}$, split ratios $\{\beta_{i,j}^c\}$, and mutual restriction intervals $\{\eta_{j',j}^i\}$ ($j' \neq j$), the algorithm of Section 3.3 obtains the unique solution of the optimization problem (3.32)-(3.38).*

Proof. The priority constraint (3.37) makes this optimization problem non-convex, except in the special cases mentioned in Section 3.1. In fact, we conjecture that in the MIMO case, both with and without relaxation of the FIFO constraint, there is no way to verify a solution faster than re-solving the problem (3.32)-(3.38). Thus, we will prove optimality by showing that as our algorithm proceeds through iterations, it constructs the unique optimal solution.

We may decompose the problem into finding the $M \cdot N \cdot C$ interrelated quantities $\{f_{i,j}^c\}$. The C flows for each (i, j) are further constrained by our commodity flow proportionality constraint (3.36); solving for one of $\{f_{i,j}^1, \dots, f_{i,j}^C\}$ also finds them all. Our task then becomes finding optimal values for each of $M \cdot N$ subsets $\{f_{i,j}^1, \dots, f_{i,j}^C\}$. Our algorithm finds at least one of these $M \cdot N$ subsets per iteration k . The assignments are done by either equation (3.46) or equation (3.50). Over iterations, subsets are assigned to build up the unique optimal solution. We can show that each subset assigned is optimal; that is, at least one of the constraints is tight.

Consider (3.49)-(3.50), our implementation of the relaxed FIFO constraint. In step 4 of our algorithm, we identify a single output link as j^* . The minimization in this step picks out a single link as the most restrictive of all output links. By the relaxed FIFO construction, all $i \in U_{j^*}$ will feel a partial FIFO effect instigated by j^* as the most restrictive link. For a generic $j \neq j^*$, equation (3.49) enforces the relaxed FIFO constraint by decaying the oriented demand $\tilde{S}_{i,j}(k)$. In fact, this modified oriented demand acts as a proxy for the relaxed FIFO constraint. Since a $f_{i,j}^c$ that is restricted by partial FIFO will, by construction, never obtain $f_{i,j}^c = S_{i,j}^c$, the running quantity $\tilde{S}_{i,j}^c(k)$ represents an ‘‘effective’’ oriented demand after application of partial FIFO. A flow that is not constrained by partial FIFO will nevertheless not exceed its demand, as $\tilde{S}_{i,j}^c(0) = S_{i,j}^c$, and $\tilde{S}_{i,j}^c(k)$ only decreases across k as relaxed FIFO constraints are considered (ensuring compliance with the demand constraint (3.34)). On the other hand, if it turns out that a flow is found $f_{i,j}^c = \tilde{S}_{i,j}^c(k) \leq S_{i,j}^c$ for some k , then this flow value may have some constraint imposed by partial FIFO.

Flows assigned by equation (3.50) for $j \neq j^*$ are constrained by a relaxed FIFO constraint (3.38). To see this, note that

$$\mathcal{A}(\mathcal{Q}_{j',j}^i) = |\boldsymbol{\eta}_{j',j}^i| \left(1 - \frac{\sum_{c=1}^C f_{i,j'}^c}{S_{i,j'}} \right) S_{i,j} \quad (3.51)$$

where, recall, $\mathcal{Q}_{j',j}^i$ is defined by (3.39). As applied in (3.49), $j' = j^*$ always, and $f_{i,j^*}^c = \tilde{S}_{i,j^*}^c$ by (3.50).

For a union of two rectangles $\mathcal{Q}_{j',j}^i$ and $\mathcal{Q}_{j'',j}^i$, we have

$$\mathcal{A}(\mathcal{Q}_{j',j}^i \cup \mathcal{Q}_{j'',j}^i) = \mathcal{A}(\mathcal{Q}_{j',j}^i) + \mathcal{A}(\mathcal{Q}_{j'',j}^i) - \mathcal{A}(\mathcal{Q}_{j',j}^i \cap \mathcal{Q}_{j'',j}^i) \quad (3.52)$$

and

$$\mathcal{A}(\mathcal{Q}_{j',j}^i \cap \mathcal{Q}_{j'',j}^i) = (|\boldsymbol{\eta}_{j',j}^i \cap \boldsymbol{\eta}_{j'',j}^i|) \min_{j^\# \in \{j', j''\}} \left(1 - \frac{\sum_{c=1}^C f_{i,j^\#}^c}{S_{i,j^\#}} \right) S_{i,j}. \quad (3.53)$$

Note that $j^\#$ as defined in (3.53) is the less-restricted of the two output links j' and j'' . This means that in the context of our algorithm, since the most-restricted link j^* is picked at each iteration, subsequent links j^* at later iterations will always be less restricted than those in previous iterations. Equation (3.49) thus takes the union (with the subtraction of the intersection) as done in (3.52)-(3.53), and incorporates the relaxed FIFO constraint (3.38).

Now consider equations (3.47) and (3.50), which apply in the special case where $j = j^*$. We have

$$\begin{aligned}
\sum_{c=1}^C \sum_{i \in U_{j^*}(k)} f_{i,j^*}^c &= \sum_{c=1}^C \sum_{i \in U_{j^*}(k)} \tilde{S}_{i,j^*}(k) \frac{\tilde{p}_{i,j^*}(k) a_{j^*}(k)}{\sum_{c'=1}^C \tilde{S}_{i,j^*}^{c'}(k)} \\
&= \sum_{i \in U_{j^*}(k)} \tilde{p}_{i,j^*}(k) a_{j^*}(k) \\
&= \left(\sum_{i \in U_{j^*}(k)} \tilde{p}_{i,j^*}(k) \right) \frac{\tilde{R}_{j^*}(k)}{\sum_{i \in U_{j^*}(k)} \tilde{p}_{i,j^*}(k)} \\
&= \tilde{R}_{j^*}(k),
\end{aligned}$$

so all the flows into link j^* take up all available supply. These flows are thus constrained by the supply constraint, (3.35). \square

Corollary 3.1. *Theorems 3.1 and 3.2 follow from Theorem 3.3 as special cases.*

Remark 3.4. *Note that setting the priorities equal to input link capacities, $p_i = F_i$, and all restriction intervals $\eta_{j',j}^i = [0, 1]$, we recover the original node model and algorithm of Tampère et al. (2011).*

4 Examples

We present two examples to demonstrate the computation of node flows with our node model. The first example is an extension of the example with four input links and four output links from Tampère et al. (2011), where we have extended it to include the partial FIFO construction for some input links.

The second example represents an onramp to a freeway with a parallel managed lane facility as a node with three input links (onramp, freeway and managed lane) and two output links (freeway and managed lane). In this example, we vary the input links' priorities to demonstrate how this affects the resulting node flows.

4.1 Example One: An example with partial FIFO

Consider the node schematically presented in Figure 5a. This node has four input links (links 1 through 4) and four output links (links 5 through 8).

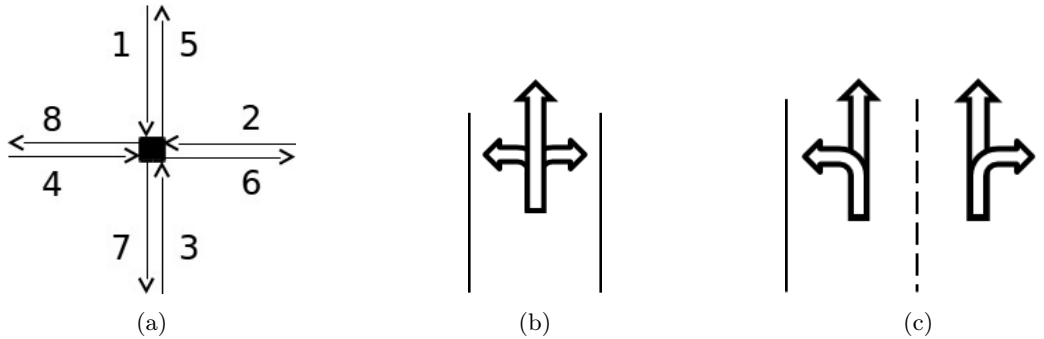


Figure 5: Items used in Example One: (a) four-input, four-output node used, (b) one-lane structure and turn movements for input links 1 and 3 and (c) two-lane structure and turn movements for input links 2 and 4.

Suppose that our relevant parameters are:

$C = 1$			
$S_1 = 500$	$S_2 = 2000$	$S_3 = 800$	$S_4 = 1700$
$p_1 = F_1 = 1000$	$p_2 = F_2 = 2000$	$p_3 = F_3 = 1000$	$p_4 = F_4 = 2000$
$R_5 = 1000$	$R_6 = 2000$	$R_7 = 1000$	$R_8 = 2000$
$\beta_{1,5} = 0$	$\beta_{1,6} = 0.1$	$\beta_{1,7} = 0.3$	$\beta_{1,8} = 0.6$
$\beta_{2,5} = 0.05$	$\beta_{2,6} = 0$	$\beta_{2,7} = 0.15$	$\beta_{2,8} = 0.8$
$\beta_{3,5} = 0.125$	$\beta_{3,6} = 0.125$	$\beta_{3,7} = 0$	$\beta_{3,8} = 0.75$
$\beta_{4,5} = 1/17$	$\beta_{4,6} = 8/17$	$\beta_{4,7} = 8/17$	$\beta_{4,8} = 0$

where F_i denotes the capacity of link i and, we have omitted the commodity index c on the demands and split ratios for readability, since there is only one commodity.

As stated above, these values are taken from the example of Tampère et al. (2011). Note that input links 2 and 4 have twice the capacity of input links 1 and 3. Let us say that links 2 and 4 are two-lane roads, and links 1 and 3 are one-lane roads (see Figures 5b and 5c). Then, we can create a partial FIFO characterization of the multilane links 2 and 4.

4.1.1 Defining restriction intervals

Let us define restriction intervals

$$\eta_{j',j}^2 = \begin{cases} j' = 5 & \begin{matrix} j = 5 & j = 7 & j = 8 \\ [0, 1] & \emptyset & [1/2, 1] \end{matrix} \\ j' = 7 & \begin{matrix} \emptyset & [0, 1] & [0, 1/2] \end{matrix} \\ j' = 8 & \begin{matrix} [0, 1] & [0, 1] & [0, 1] \end{matrix} \end{cases}$$

$$\eta_{j',j}^4 = \begin{cases} j' = 5 & \begin{matrix} j = 5 & j = 6 & j = 7 \\ [0, 1] & [0, 1/2] & \emptyset \end{matrix} \\ j' = 6 & \begin{matrix} [0, 1] & [0, 1] & [0, 1] \end{matrix} \\ j' = 7 & \begin{matrix} \emptyset & [1/2, 1] & [0, 1] \end{matrix} \end{cases}$$

To read the above tables, recall that as written, j' is the restricting link, and j is the restricted link. When j' runs out of supply, the restriction intervals in that row become active. Recall that the total restriction for a single movement (i, j) is the union of all active intervals $\eta_{j',j}^i$.

We will describe the meaning of each element in the table for $\eta_{j',j}^2$ as part of this example. This set of restriction intervals describe the behavior of partial FIFO blocking on input link 2 when an output link runs out of supply and becomes congested. The intervals are meant to describe a two-lane road with left (onto link 7), straight (onto link 8), and right (onto link 5) movements. The left turn movement is allowed on the left lane, the right turn movement is allowed on the right lane, and the straight movement is allowed on both lanes (see Fig 5c).

The interval $\eta_{5,7}^2$ is the empty interval because, when the right turn movement is blocked (due to link 5 running out of supply), the right lane will begin to queue. However, no vehicles intending to turn left (i.e., to link 7) will be in this lane, since the left turn movement is not allowed in the right lane. Instead, the left-turning vehicles will all be in the left lane, which will not be blocked from link 7's spillback since no right-turning vehicles will be in the left lane (as the right turn movement is not allowed from the left lane). Therefore, when the right turn movement is blocked, no left-turning vehicles will be in the blocked lanes, and the left turn movement (link 7) is unaffected by spillback in the right turn movement (link 5), or $\eta_{5,7}^2 = \emptyset$. This argument also applies to $\eta_{7,5}^2$, with the left and right lanes and turn movements switched.

The interval $\eta_{5,8}^2 = [1/2, 1]$ encodes the effect of the blocking of the right lane described in the previous paragraph on the through movement. The through movement is allowed in both lanes, so when the right

lane is queueing from spillback from the right turn, the right lane will also have vehicles that are trying to go straight stuck in this queue. Therefore, the right half-interval, or $[1/2, 1]$, of the through-movement-allowing-lanes will be blocked by queueing to take the right turn when link 5 spills back onto link 2. This same argument explains how $\boldsymbol{\eta}_{7,8}^2 = [0, 1/2]$, as the left half-interval will be blocked when the left turn spills back. Note that since the total restriction interval for a movement is the union of all active intervals, this means that when both links 5 and 7 are congested and spill back onto link 2, the total restriction interval for movement (2, 8) is $\boldsymbol{\eta}_{5,8}^2 \cup \boldsymbol{\eta}_{7,8}^2 = [0, 1]$, which means that the through movement becomes fully blocked when vehicles in link 2 are queueing to take both the left and right turn movements, and queueing in both the left and right lanes.

We have $\boldsymbol{\eta}_{8,5}^2 = [0, 1]$ and $\boldsymbol{\eta}_{8,7}^2 = [0, 1]$ to describe the effects of spillback from the through movement, link 8, on the turn movements. Since the through movement is allowed in both lanes, vehicles trying to enter a blocked lane 8 will queue in both lanes, and all vehicles trying to take the turn movements will also be stuck in these queues.

Finally, the diagonal entries, $\boldsymbol{\eta}_{5,5}^2$, $\boldsymbol{\eta}_{7,7}^2$, and $\boldsymbol{\eta}_{8,8}^2$ are $[0, 1]$ by definition. This is because when j' has no supply and is restricting, it is obviously also blocked by its own queue.

Input link 4 is assumed to have the same lane movement rules as link 2, so its restriction intervals are mirrored. The one-lane input links, 1 and 3, are assumed to operate under full FIFO, as any queue would block the only lane. That is, $\boldsymbol{\eta}_{j',j}^i = [0, 1]$, for all j', j and $i = 1, 3$.

4.1.2 Solution

We will now demonstrate the use of our node flow model solution algorithm to resolve the node flows from these supplies and demands.

First, we compute the oriented demands, with $S_{i,j} = \beta_{i,j} S_i$.

$S_{1,5} = 0$	$S_{1,6} = 50$	$S_{1,7} = 150$	$S_{1,8} = 300$
$S_{2,5} = 100$	$S_{2,6} = 0$	$S_{2,7} = 300$	$S_{2,8} = 1600$
$S_{3,5} = 100$	$S_{3,6} = 100$	$S_{3,7} = 0$	$S_{3,8} = 600$
$S_{4,5} = 100$	$S_{4,6} = 800$	$S_{4,7} = 800$	$S_{4,8} = 0$

We now outline step-by-step how our algorithm proceeds. The numbers in each iteration k correspond to the numbered steps in the algorithm in Section 3.3.

$k = 0$:

1. $U_5(0) = \{2, 3, 4\}$; $U_6(0) = \{1, 3, 4\}$; $U_7(0) = \{1, 2, 4\}$; $U_8(0) = \{1, 2, 3\}$
2. $V(0) = \{5, 6, 7, 8\}$
3. No adjustments to the \tilde{p}_i are made. The oriented priorities are:

$\tilde{p}_{1,5} = 0$	$\tilde{p}_{1,6} = 100$	$\tilde{p}_{1,7} = 300$	$\tilde{p}_{1,8} = 600$
$\tilde{p}_{2,5} = 100$	$\tilde{p}_{2,6} = 0$	$\tilde{p}_{2,7} = 300$	$\tilde{p}_{2,8} = 1600$
$\tilde{p}_{3,5} = 125$	$\tilde{p}_{3,6} = 125$	$\tilde{p}_{3,7} = 0$	$\tilde{p}_{3,8} = 750$
$\tilde{p}_{4,5} = 118$	$\tilde{p}_{4,6} = 941$	$\tilde{p}_{4,7} = 941$	$\tilde{p}_{4,8} = 0$
4. $a_5(0) = 2.92$; $a_6(0) = 1.72$; $a_7(0) = 0.649$; $a_8(0) = 0.678$
 $a_{j^*}(0) = a_7(0) = 0.649$
5. $\tilde{U}(0) = \{1\}$, as $S_1 = 500 \leq \tilde{p}_1(0)a_7(0) = 1000 \times 0.649$

- $f_{1,6} = S_{1,6} = 50$; $f_{1,7} = S_{1,7} = 150$; $f_{1,8} = S_{1,8} = 300$;
- $\tilde{R}_6(1) = 2000 - 50 = 1950$; $\tilde{R}_7(1) = 1000 - 150 = 850$; $\tilde{R}_8(1) = 2000 - 300 = 1700$
- $U_5(1) = \{2, 3, 4\}$; $U_6(1) = \{3, 4\}$; $U_7(1) = \{2, 4\}$; $U_8(1) = \{2, 3\}$

In this iteration, we calculated that output link 7 was the most-demanded output link, and that of its demanding links, only input link 1 would be able to send its full demand before link 7 would run out of supply and become congested. Input link 1 is able to send its full demand.

$k = 1$:

2. $V(1) = \{5, 6, 7, 8\}$
3. No changes are made to priorities or oriented priorities.
4. $a_5(1) = 2.92$; $a_6(1) = 1.83$; $a_7(1) = 0.685$; $a_8(1) = 0.723$
 $a_{j^*}(1) = a_7(1) = 0.685$
5. $\tilde{U}(1) = \emptyset$, as $\sum_{j \in V(k)} \tilde{S}_{2j}(2) = 2000 \not\leq \tilde{p}_2(1)a_7(1) = 1370$,
and $\sum_{j \in V(k)} \tilde{S}_{4j} = 1700 \not\leq \tilde{p}_4(1)a_7(1) = 1370$,
 - $f_{2,7} = \tilde{p}_{2,7}(1)a_7(1) = 205.5 = \tilde{S}_{2,7}(2)$
 $f_{4,7} = \tilde{p}_{4,7}(1)a_7(1) = 644.5 = \tilde{S}_{4,7}(2)$
 - $\tilde{\eta}_5^2(2) = \tilde{\eta}_5^2(1) \cup \eta_{7,5}^2 = \emptyset \cup \emptyset = \emptyset$
 $\tilde{\eta}_8^2(2) = \tilde{\eta}_8^2(1) \cup \eta_{7,8}^2 = \emptyset \cup [0, 1/2] = [0, 1/2]$
 $\tilde{\eta}_5^4(2) = \tilde{\eta}_5^4(1) \cup \eta_{7,5}^4 = \emptyset \cup \emptyset = \emptyset$
 $\tilde{\eta}_6^4(2) = \tilde{\eta}_6^4(1) \cup \eta_{7,6}^4 = \emptyset \cup [1/2, 1] = [1/2, 1]$
 - $\tilde{S}_{2,8}(2) = 1600 - 1600^{(1/2)} \left(1 - \frac{205.5}{300}\right) = 1348$
 $\tilde{S}_{4,6}(2) = 800 - 800^{(1/2)} \left(1 - \frac{644.5}{800}\right) = 772.25$
 - $\tilde{R}_7(2) = 850 - 644.5 - 205.5 = 0$
 - $U_5(2) = \{2, 3, 4\}$; $U_6(2) = \{3, 4\}$; $U_7(2) = \emptyset$; $U_8(2) = \{2, 3\}$

In this iteration, we found that, after accounting for input link 1's flow in the previous iteration, output link 7 remains the most-demanded output link. However, none of the input links will be able to send all of their demand before link 7's supply is exhausted. Therefore, the remaining supply of link 7 is assigned priority-proportional to input links 2 and 4, and restriction intervals $\eta_{7,j}^i$ become active. The restriction intervals for links 2 and 4 reflect the overlapping lanes of the restricted movement (i, j) with the restricted movement $(i, 7)$. Since $(i, 7)$ is a turn movement for $i = 2, 4$, this means that the through movement has its overlapping half-interval restricted, and the other turn movement is not restricted. The new values for $\tilde{S}_{2,8}(2)$ and $\tilde{S}_{4,6}(2)$ describe the new maximum flow for these movements, the subtracted term representing the vehicles that will be stuck in the queued lane when the congestion in the turn movement spills back.

Since input link 3 had no demand for the movement $(3, 7)$, there are no vehicles that will queue for that movement when output link 7 becomes congested, and so there are no restriction intervals that become active for input link 3 at this time.

$k = 2$:

2. $V(2) = \{5, 6, 8\}$
3. No changes are made to priorities or oriented priorities.

4. $a_5(2) = 2.82$; $a_6(2) = 1.83$; $a_8(2) = 0.723$
 $a_{j^*}(2) = a_8(2) = 0.723$
5. $\tilde{U}(2) = \emptyset$, as $\sum_{j \in V(k)} \tilde{S}_{2j}(2) = 205.5 + 1348 = 1553.5 \not\leq \tilde{p}_2(2)a_8(2) = 1446$,
and $\sum_{j \in V(k)} \tilde{S}_{3j}(2) = 800 \not\leq \tilde{p}_3(2)a_8(2) = 723$
 - $f_{2,8} = \tilde{p}_{2,8}(2)a_8(2) = 1157.4 = \tilde{S}_{2,8}(3)$
 $f_{3,8} = \tilde{p}_{3,8}(2)a_8(2) = 542.6 = \tilde{S}_{3,8}(3)$
 - $\tilde{\eta}_5^2(3) = \tilde{\eta}_5^2(2) \cup \eta_{8,5}^2 = \emptyset \cup [0, 1] = [0, 1]$
 $\tilde{\eta}_5^3(3) = \tilde{\eta}_5^3(2) \cup \eta_{8,5}^3 = \emptyset \cup [0, 1] = [0, 1]$
 $\tilde{\eta}_6^3(3) = \tilde{\eta}_6^3(2) \cup \eta_{8,6}^3 = \emptyset \cup [0, 1] = [0, 1]$
 - $\tilde{S}_{2,5}(3) = 100 - 100(1) \left(1 - \frac{1157.4}{1600}\right) = 72.3$
 $\tilde{S}_{3,5}(3) = 100 - 100(1) \left(1 - \frac{542.6}{800}\right) = 67.8$
 $\tilde{S}_{3,6}(3) = 100 - 100(1) \left(1 - \frac{542.6}{800}\right) = 67.8$
 - $f_{2,5} = \tilde{S}_{2,5}(3) = 72.3$
 $f_{3,5} = \tilde{S}_{3,5}(3) = 67.8$
 $f_{3,6} = \tilde{S}_{3,6}(3) = 67.8$
 - $\tilde{R}_8(3) = 1700 - 1157.4 - 542.6 = 0$
 $\tilde{R}_5(3) = 1000 - 72.3 - 67.8 = 859.9$
 $\tilde{R}_6(3) = 1950 - 67.8 = 1882.2$
 - $U_5(3) = \{4\}$; $U_6(3) = \{4\}$; $U_8(3) = \emptyset$

This iteration is similar to the previous iteration in that an output link, this time link 8, has its supply exhausted. However, the two input links under consideration, links 2 and 3, have a full FIFO restriction on their flows when link 8 runs out of supply. For link 2, this is because (2, 8) is the through movement whose queue takes up both lanes as described in Section 4.1.1. For link 3, this is because it has only one lane, so any queue blocks all traffic. The remaining flows for input links 2 and 3 are therefore determined by this FIFO restriction.

$k = 3$:

2. $V(3) = \{5, 6\}$
3. No changes are made to priorities or oriented priorities.
4. $a_5(3) = 7.29$; $a_6(3) = 2.35$
 $a_{j^*}(3) = a_6(3) = 2.35$
5. $\tilde{U}(3) = \{4\}$, as $\sum_{j \in V(k)} \tilde{S}_{4j}(3) = 100 + 772.25 = 872.25 \leq \tilde{p}_4(3)a_6(3) = 4705.5$,
 - $f_{4,5} = \tilde{S}_{4,5}(3) = 100$
 $f_{4,6} = \tilde{S}_{4,6}(3) = 772.25$
 - $\tilde{R}_5(4) = 859.9 - 100 = 759.9$
 $\tilde{R}_6(4) = 1882.2 - 759.9 = 1122.3$
 - $U_5(4) = \emptyset$; $U_6(4) = \emptyset$

This iteration finds the remaining flows. Note that link 4's through movement, (4,6), is already under effect of partial FIFO, so it is not able to fill its entire demand $S_{4,6}$.

$k = 4$:

2. $V(4) = \emptyset$, so the algorithm terminates. All flows have been found.

It is useful to compare the solution to this example problem to that of the example in Tampère et al. (2011) (Tables 1 and 2). As noted above, all values for this example are the same as those in Tampère et al. (2011), except in the present example we have added the partial FIFO relaxation to some movements. In Tampère et al. (2011), with the full FIFO restriction, iteration 0 proceeded the same, but at iteration 1, when link 7’s supply is exhausted, *all* of links 2 and 4’s flows were set due to the full FIFO restriction. Then, the full-FIFO example finds that input link 3 is able to have all of its demand satisfied, as the high amount of spillback from link 7 to links 2 and 4 left more supply available for link 3 in link 8 than in the present example.

	$j = 5$	$j = 6$	$j = 7$	$j = 8$
$i = 1$	0	50	150	300
$i = 2$	72.3	0	205.5	1157.4
$i = 3$	67.8	67.8	0	542.6
$i = 4$	100	772.3	644.5	0

Table 1: Flows $f_{i,j}$ resulting from enforcing partial FIFO as specified in the present example

	$j = 5$	$j = 6$	$j = 7$	$j = 8$
$i = 1$	0	50	150	300
$i = 2$	68.5	0	205.5	1096
$i = 3$	100	100	0	600
$i = 4$	80.6	644.5	644.5	0

Table 2: Flows $f_{i,j}$ resulting from enforcing full FIFO (from Tampère et al. (2011, Table 8))

In other words, if this example is meant to model a junction where two input links (2 and 4) have two lanes and twice the supply as the other two, a full FIFO construction may have led to the “unrealistic spillback” phenomenon we discussed in Section 2.2, with queueing for a turning movement causing spillback into through movements and non-competing turning movements. Further, as this example demonstrates, an overly aggressive spillback model can lead to knock-on effects in differences in other input links’ flows, as shown by link 3 being able to send all its supply when the two-lane links suffer full FIFO from a turn movement queue, but is not able to when the two-lane links have only a partial FIFO restriction.

4.2 Example Two: Variation of input links’ priorities

Consider the node schematically presented in Figure 6. This node has three input links (links 1 through 3) and two output links (4 and 5). This node might represent an onramp (link 3) joining a freeway (links 1 and 4) with a parallel managed lane facility (links 2 and 5). In this situation, we will have two vehicle commodities, with $c = 1$ representing vehicles that cannot enter the managed lane facility, and $c = 2$ representing vehicles that may, but are not required to, enter the managed lane facility.

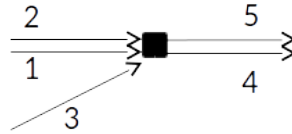


Figure 6: Three-input, two-output node used in Example Two.

Let us define our parameters as follows:

$C = 2$			
$S_1^1 = 1700$	$S_2^1 = 0$	$S_3^1 = 400$	
$S_1^2 = 200$	$S_2^2 = 500$	$S_3^2 = 200$	
$F_1 = 4000$	$F_2 = 2000$	$F_3 = 1000$	
$R_4 = 2000$	$R_5 = 1000$		
$\beta_{1,4}^1 = 1$	$\beta_{1,5}^1 = 0$	$\beta_{1,4}^2 = 0.2$	$\beta_{1,5}^2 = 0.8$
$\beta_{2,4}^1 = 1$	$\beta_{2,5}^1 = 0$	$\beta_{2,4}^2 = 0.1$	$\beta_{2,5}^2 = 0.9$
$\beta_{3,4}^1 = 1$	$\beta_{3,5}^2 = 0$	$\beta_{3,4}^2 = 0.5$	$\beta_{3,5}^2 = 0.5$
$\eta_{j',j}^i = [0, 1]$ for all i, j, j' (Full FIFO).			

To demonstrate the effect of varying input link priorities on node flows, we will use three different priority-assignment schemes with the above values. This also demonstrates the usefulness of our node model in studying the effect of priorities on node flows: For the sake of brevity, we will not show intermediate calculations here as was done in the previous example, but simply show and discuss the final node flows.

Capacity-proportional priorities: Tampère et al. (2011) recommended assigning downstream supply among the node flows in proportion to the input links' capacities. As we mentioned in Remark 3.4, setting the priorities in our node model equal to capacities and using strict FIFO recovers the Tampère et al. (2011) node model as a special case. So, setting $p_i = F_i$ for $i = 1, 2, 3$ and calculating node flows, we obtain the flows shown in Table 3.

		$j = 4$	$j = 5$
$i = 1$	$c = 1$	1552.1	0
	$c = 2$	36.52	146.1
$i = 2$	$c = 1$	0	0
	$c = 2$	50	450
$i = 3$	$c = 1$	289.1	0
	$c = 2$	72.28	72.28
$R_j - \sum_i \sum_c f_{i,j}^c$		0	331.6

Table 3: Flows $f_{i,j}^c$ from Example Two with capacity-proportional priorities

Here, the GP link ($j = 4$) had its supply filled, while the managed lane link ($j = 5$) had some leftover supply. The input GP link ($i = 1$) and onramp link ($i = 3$) both had some unfulfilled special ($c = 2$) vehicles that could have taken this leftover supply in the managed lane link, but they were restricted by strict FIFO from link 1's supply exhausting. Note that all of the demand of the input managed lane link ($i = 2$) was able to be satisfied. This will be in contrast to the next set of priorities:

Demand-proportional priorities: As we mentioned in Remark 3.1 and elsewhere, in practice one should not assign priorities in proportion to the input links' demand, as this would produce flows that would violate nodal invariance principles (Tampère et al., 2011). However, for the purpose of this example, we will use priorities equal to the total demands purely for demonstration purposes, so that the resulting flows can be contrasted with other priority examples.

Setting $p_1 = 1900, p_2 = 500, p_3 = 600$, and calculating node flows, we obtain the flows shown in Table 4.

Note that, while similar to the previous example in that link $j = 4$ has its supply exhausted and link $j = 5$ has some extra $c = 2$ special vehicles that cannot reach it, this example is different from the previous example in that the input managed lane link ($i = 2$) is not able to send all of its vehicles. Vehicles exiting this link encounter a queue entering link $j = 4$ when that link fills, which blocks vehicles entering link $j = 5$ due to this example's strict FIFO construction. This blockage leads to more spillback and less flow through the node, as evidenced by the higher amount of leftover supply in link $j = 5$.

Onramp-preference priorities: Some authors (e.g., Coogan and Arcaç (2014), Gomes and Horowitz (2006)) have suggested that the particular problem of a relatively small onramp merging with a larger

		$j = 4$	$j = 5$
$i = 1$	$c = 1$	1484.7	0
	$c = 2$	34.93	139.7
$i = 2$	$c = 1$	0	0
	$c = 2$	43.67	393.0
$i = 3$	$c = 1$	349.3	0
	$c = 2$	87.33	87.33
$R_j - \sum_i \sum_c f_{i,j}^c$		0	379.9

Table 4: Flows $f_{i,j}^c$ from Example Two with demand-proportional priorities

freeway should be modeled such that vehicles from the onramp are able to claim any available downstream supply *first*, so that a queue would always appear on the mainline before the onramp. In addition, this type of node model at onramp junctions is implicit in many ramp metering control schemes, from classic schemes such as ALINEA (Papageorgiou et al., 1991) to recent neural-network-based reinforcement learning approaches (e.g., Rezaee (2014); Belletti et al. (2017)). In particular, a ramp metering control problem often defines the onramp *flow* as an exogenous input that is controllable by metering, rather than the onramp *demand* (that is, the freeway is assumed to always accept all demand that the meter sends from the onramp). In our node model, we can create this special case by proper initialization of the priorities. In particular, setting $p_1 = p_2 = 0$ and $p_3 > 0$, our node model will assign all necessary supply to the onramp on its first iteration. Then, on subsequent iterations, in (3.42) input links $i = 1, 2$ will have nonzero priority and be able to claim the leftover supply. Following this procedure, we obtain the flows shown in Table 5.

		$j = 4$	$j = 5$
$i = 1$	$c = 1$	1416.7	0
	$c = 2$	33.33	133.3
$i = 2$	$c = 1$	0	0
	$c = 2$	50	450
$i = 3$	$c = 1$	400	0
	$c = 2$	100	100
$R_j - \sum_i \sum_c f_{i,j}^c$		0	316.7

Table 5: Flows $f_{i,j}^c$ from Example Two with onramp-preference priority scheme

Note that we have indeed satisfied all demand of the onramp link ($i = 3$) with this priority scheme. In this particular example, we have $p_1 = p_2 = 0.5$ when we reset their priorities, as prescribed in (3.42). However, a different version of (3.42) that resets the priorities to nonequal values (e.g., $p_i = F_i$) is possible. In this case, assigning higher priority to input link $i = 1$ may prevent $i = 2$ from sending its full demand. This further demonstrates the variability of node flows to the priority-assignment scheme.

In summary, Example Two shows that input priorities determine not only define the upstream directions of queue formations, but can also affect the node throughput.

5 Conclusion

This paper discussed and addressed several node-rooted modeling issues that the authors have encountered in macroscopic simulation of large and/or high-dimensional road networks. First, the node model framework of Tampère et al. (2011) was extended to the case of arbitrary input priorities, with discussion of what input priorities might represent. Our node model also allows priorities to have zero values, and, in the style of the literature, is formulated as a mathematical optimization problem.

Second, we discussed a tradeoff between unrealistic spillback behavior and a need to model a road with multiple links, and traced this tradeoff back to the first-in-first-out (FIFO) constraint that is present in common macroscopic node models. To resolve this dilemma, we proposed a generalization of the FIFO constraint that is easily applicable to node models of the common Tampère et al. (2011) framework. In addition, our FIFO relaxation is intuitive due to the ability to illustrate its effects with simple geometrical shapes.

We believe our node model, and our relaxed FIFO construction, is widely applicable across link models. Like any node model in the Tampère et al. (2011) framework, it can be applied with any link model that defines supplies, demands, and some form of priority. In addition, while this paper’s discussion of the node flow problem strictly deals with discrete-time simulations (i.e., at some timestep t , some fixed supplies $S_{i,j}^c(t)$ and demands $R_j(t)$ are given, and the node model computes some set of throughflows $f_{i,j}^c$ at that timestep), which, like the classic CTM are usually of fixed temporal step size, we have written a companion paper (Wright et al., 2016) where the flows computed by this paper’s node model are shown to be equal to those computed by a particular hybrid dynamic system. In the dynamic system setting, the flows $f_{i,j}^c$ are given in terms of a time differential, $\frac{d}{dt} f_{i,j}^c(t)$. Thus, this paper’s node model (and all other node models of the Tampère et al. (2011) framework, of which this paper’s node model is a generalization) and its relaxed FIFO construction should be applicable in simulations with continuous-time link models (where the supplies and demands are given by differential equations) or variable-step-size simulations such as in Raadsen et al. (2016).

As a final note, we want to emphasize the usefulness of our generalization of the Tampère et al. (2011)-style “general class of node models” framework to multi-commodity flows. Many other works in the node model literature (e.g., Gibb (2011); Corthout et al. (2012); Smits et al. (2015); Jabari (2016)⁴), but as noted, we considered multi-commodity flows, with several commodities in the same link and taking the same movement, in this paper. It turns out that our expression of multi-commodity flows, while simple (Requirement 8 in Section 2.1), can be quite powerful. In a forthcoming paper (Wright and Horowitz, 2017), we make use of this construction, and give each commodity its own fundamental diagram, to extend this paper’s node model to the second order of macroscopic traffic models.

All of our results have been presented in the form of constructive computational algorithms that are readily implementable in macroscopic traffic simulation software.

Acknowledgements

We would like to express great appreciation to our colleagues Elena Dorogush and Ajith Muralidharan for sharing ideas, Ramtin Pedarsani, Brian Phegley and Pravin Varaiya for their critical reading and their help in clarifying some theoretical issues, as well as the anonymous reviewers for their many useful comments.

This research was funded by the California Department of Transportation.

⁴ Note, however, that Jabari (2016), in order to specify the non-concurrence of conflicting turn movements, defined different movements as different commodities with their own fundamental diagrams.

A Notation

Symbol	Definition
k	Iteration index
i	Index of links entering a node
M	Number of links entering a node
j	Index of links exiting a node
N	Number of links exiting a node
c	Vehicle commodity index
C	Number of vehicle commodities
S_i^c	Demand for commodity c of link i
S_i	Total demand for input link i , $S_i = \sum_{c=1}^C S_i^c$
R_j	Supply of link j
p_i	Priority of input link i
$f_{i,j}^c$	Flow of vehicle commodity c from link i to link j
$\eta_{j,j'}^i$	Mutual restriction interval of link j onto link j' for link i
$U(k)$	Set of input links whose flows have yet to be fully determined as of iteration k
$V(k)$	Set of output links whose flows have yet to be fully determined as of iteration k
$\tilde{S}_i^c(k)$	Adjusted demand for commodity c of link i as of iteration k
$\tilde{S}_i(k)$	Total adjusted demand for input link i as of iteration k
$\beta_{i,j}^c$	Split ratio of c vehicles from link i to link j
$\tilde{p}_i(k)$	Adjusted priority of link i at iteration k
$\tilde{R}_j(k)$	Adjusted supply of link j at iteration k
$U_j(k)$	Set of input links contributing to link j whose flows are undetermined as of iteration k
$\tilde{p}_{i,j}(k)$	Oriented priority from link i to j at iteration k
$a_j(k)$	Restriction term of link j at iteration k
$a_{j^*}(k)$	Smallest (most restrictive) restriction term at iteration k
$\alpha_j(k)$	Reduction factor of link j at iteration k
$\tilde{U}(k)$	Set of input links whose demand can be fully met by downstream links at iteration k

B MISO node solution algorithm

1. Initialize:

$$\begin{aligned} \tilde{R}_1(0) &:= R_1; \\ U(0) &:= \{1, \dots, M\}; \\ k &:= 0. \end{aligned}$$

Here, k is the iteration index; $\tilde{R}_1(k)$ is the remaining supply of the output link at iteration k ; and $U(k)$ is the set of still unprocessed input links at iteration k : input links whose input-output flows have not been assigned yet.

2. Check that at least one of the unprocessed input links has nonzero priority, otherwise, assign equal positive priorities to all the unprocessed input links:

$$\tilde{p}_i(k) = \begin{cases} p_i, & \text{if there exists } i' \in U(k) : p_{i'} > 0, \\ \frac{1}{|U(k)|}, & \text{otherwise,} \end{cases}$$

where $|U(k)|$ denotes the number of elements in set $U(k)$.

- Define the set of input links that want to send fewer vehicles than their allocated supply and whose flows are still undetermined:

$$\tilde{U}(k) = \left\{ i \in U(k) : \sum_{c=1}^C S_i^c \leq \tilde{p}_i(k) \frac{\tilde{R}_1(k)}{\sum_{i' \in U(k)} \tilde{p}_{i'}(k)} \right\}.$$

- If $\tilde{U}(k) \neq \emptyset$, assign:

$$\begin{aligned} f_{i1}^c &= S_i^c, \quad i \in \tilde{U}(k); \\ \tilde{R}_1(k+1) &= \tilde{R}_1(k) - \sum_{i \in \tilde{U}(k)} \sum_{c=1}^C f_{i,1}^c; \\ U(k+1) &= U(k) \setminus \tilde{U}(k). \end{aligned}$$

- Else, assign:

$$\begin{aligned} f_{i,1}^c &= S_i^c \frac{\tilde{p}_i(k)}{\sum_{i' \in U(k)} \tilde{p}_{i'}(k)} \frac{\tilde{R}_1(k)}{S_i}, \quad i \in U(k); \\ U(k+1) &= \emptyset. \end{aligned}$$

- If $U(k+1) = \emptyset$, then stop.

- Set $k := k + 1$, and return to step 2.

This algorithm finishes after no more than M iterations, and in the special case of $M = 2$ it reduces to

$$f_{1,1}^c = \min \left\{ S_1^c, S_1^c \frac{\max \left\{ \frac{\tilde{p}_1}{\tilde{p}_1 + \tilde{p}_2} R_1, R_1 - S_2 \right\}}{S_1} \right\}; \quad (\text{B.1})$$

$$f_{2,1}^c = \min \left\{ S_2^c, S_2^c \frac{\max \left\{ \frac{\tilde{p}_2}{\tilde{p}_1 + \tilde{p}_2} R_1, R_1 - S_1 \right\}}{S_2} \right\}, \quad (\text{B.2})$$

with \tilde{p}_i computed per step 2.

References

- F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen. Expert level control of ramp metering based on multi-task deep reinforcement learning. *arXiv preprint arXiv:1701.08832*, 2017.
- M. Bliemer. Dynamic queueing and spillback in an analytical multiclass dynamic network loading model. *Transportation Research Record*, 2029:14–21, 2007.
- California Department of Transportation (Caltrans). Office of System, Freight, & Rail Planning, 2015. URL <http://www.dot.ca.gov/hq/tpp/corridor-mobility/>.
- S. Coogan and M. Arcaç. Freeway traffic control from linear temporal logic specifications. In *ICCPs'14: ACM/IEEE 5th International Conference on Cyber-Physical Systems*, pages 36–47. IEEE Computer Society, 2014.
- R. Corthout, G. Flötteröd, F. Viti, and C. M. Tampère. Non-unique flows in macroscopic first-order intersection models. *Transportation Research Part B: Methodological*, 46(3):343–359, Mar. 2012. ISSN 01912615. doi: 10.1016/j.trb.2011.10.011. URL <http://linkinghub.elsevier.com/retrieve/pii/S0191261511001652>.

- C. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994.
- C. Daganzo. The cell transmission model, Part II: Network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995. doi: 10.1016/0191-2615(94)00022-R.
- G. Flötteröd and J. Rohde. Operational macroscopic modeling of complex urban road intersections. *Transportation Research Part B: Methodological*, 45(6):903–922, July 2011. ISSN 01912615. doi: 10.1016/j.trb.2011.04.001.
- G. Gentile, L. Meschini, and N. Papola. Spillback congestion in dynamic traffic assignment: a macroscopic flow model with time-varying bottlenecks. *Transportation Research Part B: Methodological*, 41(10):1114–1138, 2007.
- J. Gibb. Model of Traffic Flow Capacity Constraint Through Nodes for Dynamic Network Loading with Queue Spillback. *Transportation Research Record: Journal of the Transportation Research Board*, 2263: 113–122, Dec. 2011. ISSN 0361-1981. doi: 10.3141/2263-13. URL <http://trrjournalonline.trb.org/doi/10.3141/2263-13>.
- G. Gomes and R. Horowitz. Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research, Part C*, 14(4):244–262, 2006.
- M. Hadi, S. Shabaniyan, H. Ozen, Y. Xiao, M. Doherty, C. Segovia, and H. Ham. Application of Dynamic Traffic Assignment to Advanced Managed Lane Modeling. Technical report, Florida International University, 2013.
- S. E. Jabari. Node modeling for congested urban road networks. *Transportation Research Part B: Methodological*, 91:229–249, Sept. 2016. ISSN 01912615. doi: 10.1016/j.trb.2016.06.001.
- J. Lebacque and M. Khoshyaran. First-order macroscopic traffic flow models: intersection modeling, network modeling. In *The 16th International Symposium on Transportation and Traffic Theory (ISTTT)*, pages 365–386, 2005.
- D. Ni and J. D. Leonard. A simplified kinematic wave model at a merge bottleneck. *Applied Mathematical Modelling*, 29(11):1054–1072, Nov. 2005. ISSN 0307904X. doi: 10.1016/j.apm.2005.02.008.
- X. Nie and H. M. Zhang. A comparative study of some macroscopic link models used in dynamic traffic assignment. *Networks and Spatial Economics*, 5(1):89–115, 2005.
- M. Papageorgiou, H. Hadj-Salem, and J. Blosseville. ALINEA: a local feedback control law for on-ramp metering. *Transportation Research Record*, 1320:58–64, 1991.
- M. P. Raadsen, M. C. Bliemer, and M. G. Bell. An efficient and exact event-based algorithm for solving simplified first order dynamic network loading problems in continuous time. *Transportation Research Part B: Methodological*, 92:191–210, Oct. 2016. ISSN 01912615. doi: 10.1016/j.trb.2015.08.004.
- K. Rezaee. *Decentralized coordinated optimal ramp metering using multi-agent reinforcement learning*. PhD thesis, University of Toronto, 2014.
- Y. Shiomi, T. Taniguchi, N. Uno, H. Shimamoto, and T. Nakamura. Multilane first-order traffic flow model with endogenous representation of lane-flow equilibrium. *Transportation Research Part C: Emerging Technologies*, 59:198–215, Oct. 2015. ISSN 0968090X. doi: 10.1016/j.trc.2015.07.002. URL <http://linkinghub.elsevier.com/retrieve/pii/S0968090X15002429>.
- E.-S. Smits, M. C. Bliemer, A. J. Pel, and B. van Arem. A family of macroscopic node models. *Transportation Research Part B: Methodological*, 74:20–39, Apr. 2015. ISSN 01912615. doi: 10.1016/j.trb.2015.01.002. URL <http://linkinghub.elsevier.com/retrieve/pii/S0191261515000053>.

- C. M. J. Tampère, R. Corthout, D. Cattrysse, and L. H. Immers. A generic class of first order node models for dynamic macroscopic simulation of traffic flows. *Transportation Research, Part B*, 45(1):289–309, 2011. doi: 10.1016/j.trb.2010.06.004.
- D. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. Bayen. A traffic model for velocity data assimilation. *Applied Mathematics Research eXpress*, 2010(1):1–35, 2010.
- M. Wright and R. Horowitz. Fusing Loop and GPS Probe Measurements to Estimate Freeway Density. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3577–3590, Dec 2016. ISSN 1524-9050. doi: 10.1109/TITS.2016.2565438.
- M. Wright, R. Horowitz, and A. A. Kurzhanskiy. A dynamic system characterization of road network node models. In *Proceedings of the 10th IFAC Symposium on Nonlinear Control Systems*, volume 49, pages 1054–1059, August 2016. doi: 10.1016/j.ifacol.2016.10.307.
- M. A. Wright and R. Horowitz. Generic second-order macroscopic traffic node model for general multi-input multi-output road junctions via a dynamic system approach. *arXiv preprint arXiv:1707.09346*, 2017.
- I. Yperman, S. Logghe, and B. Immers. The link transmission model: An efficient implementation of the kinematic wave theory in traffic networks. In *Proceedings of the 10th EWGT Meeting*, pages 122–127, Poznan, Poland, 2005.