On node and route choice models for high-dimensional road networks

Matthew Wright, Gabriel Gomes, Roberto Horowitz, Alex A. Kurzhanskiy

January 7, 2016

Abstract

This paper discusses first order node models for macroscopic multi-commodity traffic simulation. There are two parts: (1) input-output flow computation based on input demand, input link priorities, split ratios that define how incoming flow is distributed between output links, and output supply; and (2) a traffic assignment algorithm for the case when split ratios are not known a priori or are only partially known.

Input link priorities define how output supply is distributed between incoming flows. We deal with arbitrary input link priorities. Also, in this node model we specifically address the issue of the First-In-First-Out (FIFO) rule: in the case of one of the output links being overly congested the FIFO rule may be too restrictive, blocking the flow to other outputs and causing unreasonable spillback. We relax the FIFO rule through a parametrization that describes how output links influence each other and allows tuning the model to anything between full FIFO and no FIFO.

In the existing research the prevailing approach to local traffic assignment is to use a mode choice (logit or probit) model. In this paper we discuss the shortcomings of the logit model and introduce a nonparametric route choice algorithm that produces split ratios based only on link demands and supplies.

All methods in this paper are presented for multi-commodity traffic flows.

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 density and buildup and breakdown of congestion along the lineal direction of a road; these models are based on the “kinematic wave” abstraction of traffic as a fluid flow. Mesoscopic traffic models occupy the intermediate space. This paper focuses on the "node model" portion of first-order macroscopic models, where the state of the road is fully determined by the vehicle density along its length. Of the three, the highly-abstracted macroscopic models unsurprisingly have the lowest computational cost.

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. Individual links are taken to have constant vehicle density ρ along their length[1], and the link model describes, as a function of its current density, both the amount of vehicles trying to exit the link (called the sending function $S(\rho)$, or *demand*) and the amount of vehicles that the link is able to accept (called the receiving function $R(\rho)$, or *supply*) over the next simulation timestep. A traffic model may contain multiple classes of vehicles that share the road, and each class may have their own sending functions. Separate vehicle classes are often called "commodities." Exogenous to the link and node models are the so-called *turning* or *split ratios*, which define the driver behavior at the junction - the ratios of vehicles of each commodity that take each of the available movements. 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' sending functions and its outgoing links' receiving functions.

Of course, while a macroscopic model 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 (2015)). 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 junctions multi-input our multi-output junctions, that is, a junction 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 how modern node models can exacerbate computational complexity in high-dimensional networks by creating a dilemma between model accuracy and number of links; thankfully, this tradeoff can be overcome in a simple manner, as we will detail in Section 3.

Data needs can also be prohibitive in large and high dimensional networks. Similar to link models, node models have parameters that must be learned from data. In particular, the widely-used "logit" (McFadden, 1973) method for determining driver route choice at junctions learns a statistical model for drivers' route selection as a function of a route state (e.g., the expected route travel time). When one needs a route choice model for many nodes, the data needs can be prohibitive, or their collection infeasible. Arterial networks with many junctions may not have necessary penetration of sensors. We present a parameter-free route choice model to overcome problems arising from this dearth of driver behavior data in Section 4.

[1] Some authors use an alternative terminology of the macroscopic model elements, recalling the original terms used by Daganzo (1994). In this alternative description, links may not have homogeneous density, but may instead be further divided into smaller lengths of road called cells, which themselves have homogeneous density. These cells would be equivalent to links in our terminology.

2 Common junction 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 incoming links' per-commodity demands S_i^c, split ratios β_{ij}^c (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 are resolved to produce throughflows f_{ij}^c. During simulation of a traffic network, the link and node models are dynamically evaluated at each timestep, so the quantities S_i^c, R_j, β_{ij}^c, and f_{ij}^c are in fact time-dependent. In general, the parameters of the link and node models may be time-varying as well. To simplify the notation, in this paper we consider the node model evaluation in each timestep as an isolated problem, and omit any temporal notation.

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 $\max \sum_{i,j,c} f_{ij}^c$. 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, $f_{ij}^c \geq 0$ for all i, j, c.

4. Flow conservation: Total flow entering the node must be equal to total flow exiting the node. Mathematically, $\sum_i f_{ij}^c = \sum_j f_{ij}^c$ for all c.

5. Satisfaction of demand and supply constraints. Mathematically, $\sum_j f_{ij}^c \leq S_i^c$ and $\sum_i f_{ij}^c \leq R_j$.

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, $f_{ij}^c / \sum_i f_{ij}^c = \beta_{ij}^c$.

   We believe that in some situations, the FIFO constraint may be too restrictive. It should not be completely eliminated, however, but must 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 capacity F_i in an infinitesimal time, and therefore, a node model should yield solutions that are invariant to replacing S_i with C_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 presence in this link.

As opposed to the first seven, we do not view requirement 8 as a general principle. Rather, we list it here as a rule that we follow throughout this paper. Other modelers may decide another method to distribute supply restrictions across commodities, should there be evidence that spillback affects different vehicles classes in a link at different rates.

In addition to the above numbered requirements, two additional 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, which we will denote 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 $\sum_c f_{ij}^c$. 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. We will not make use of these 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_{ij}^c, but rather the quantities $1/(\sum_c f_{ij}^c)$, 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 an 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.

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).

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 exits. The natural network representation of this junction would have one node with one input link, with demand R_1, and three output links, with supplies R_1, R_2, and R_3, similar to Figure 2a.

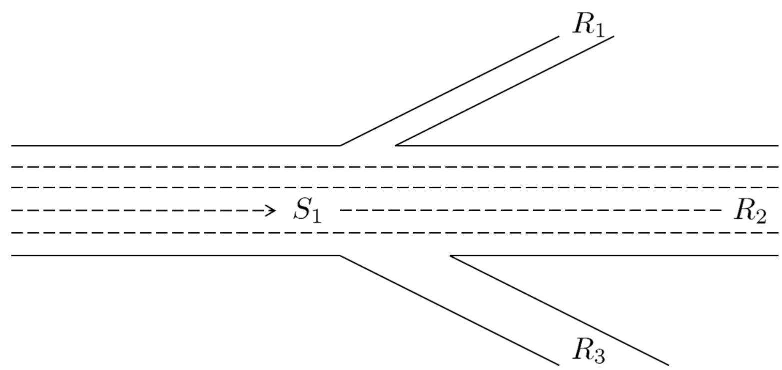

Figure 1: A node with one input and three output links, where congestion in output links 1 and 3 only partially affects flow into output link 2, while congestion in link 2 affects flows into output links 1 and 3 in full, and output links 1 and 3 do not affect each other.

Here, the FIFO constraint translates into:

$$\frac{f_{11}}{S_{11}} = \frac{f_{12}}{S_{12}} = \frac{f_{13}}{S_{13}},$$

where $f_{ij}(S_{ij})$ 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.

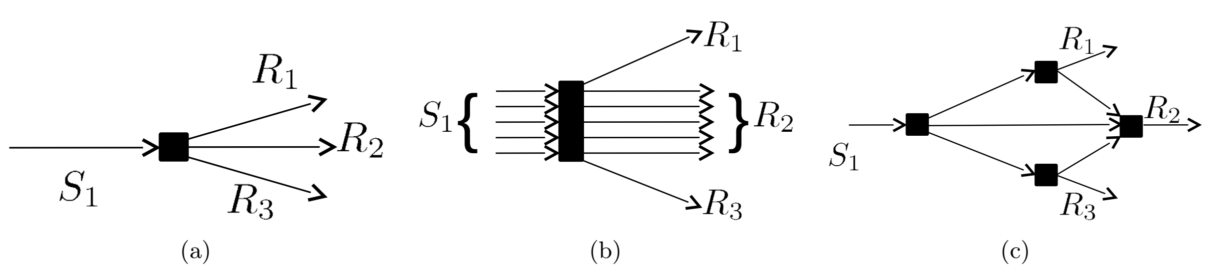

Figure 2: Three network representation of the junction in Figure 1.

Some authors (see for example 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. Secondly, 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 like in the network of Figure 2b. This approach 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 like in Figure 1.

2.3 The logit traffic assignment model

As mentioned in Section 1, the logit traffic assignment rule is a regression-based approach for determining the route choice proportions among a population. A typical logit model states that commodity c's split ratio from input link i to output link j is given by

$$\beta_{ij}^c = \frac{\exp(J_{ij}(x))}{\sum_{j'=1}^{N} \exp(J_{ij'}(x))} \tag{2.1}$$

where x is some state variable (for example, route lengths, costs, travel times, etc.) and $J_{ij}(\cdot)$ is a real-valued utility function for movement i, j. The function

$$\sigma : \mathcal{R}^K \to [0,1]^K; \quad \sigma(z)_j = \frac{\exp(z_j)}{\sum_{k=1}^{K} \exp(z_k)} \tag{2.2}$$

is often called the softmax function. It serves to "squash" the N real-valued utility functions in (2.1) to the N-dimensional simplex $[0,1]^N = \left\{\beta_i \in \mathcal{R}^N : 0 \le \beta_{ij} \le 1 \forall j \in \{1,\dots,N\}, \sum_{j=1}^N \beta_j = 1\right\}$. When the $J_{ij}(\cdot)$ are all linear or affine in their x_j, this becomes the multinomial logistic regression generalized linear model (McCullagh and Nedler, 1989).

If instead (2.1) is modified to read

$$\beta_{ij} = \frac{\exp(J_{ij}^c(x_j))}{\sum_{j'=1}^{N} \exp(J_{ij'}^c(x_j'))}, \tag{2.3}$$

the only argument to the functions $J_{ij}^c(\cdot)$ is the route-specific data x_j. This approach relies on an assumption that the utility of each lane is independent of all other lane states x_j, e.g., if the functions $J_{ij}^c(\cdot)$ in (2.1) are linear in x, then (2.3) assumes that the coefficients of all $J_{ij}^c(\cdot)$ for all elements of x not in x_j are zero. Route choice models in the transportation literature often use this formulation (a useful overview can be found in Bliemer and Bovy (2008)). Although the route choice literature often refers to (2.3) as the "multinomial logit model," (2.3) is more properly known as the "conditional logit model" (McFadden, 1973), with "multinomial logit model" reserved for the general case (2.1).

The logit route choice model is a statistical model: The $C \cdot N$ sets of parameters, one for each $J_{ij}^c(\cdot)$, are meant to be learned from data. These data take the form of input-output pairs (β_{ij}^c, x_j), i.e. observed link states x_j paired with their caused split ratios β_{ij}^c. In the use of a statistical model, the usual caveats must be kept in mind. In addition, it is desired that the data collected span much of the space of possible states x_j, or at least those subspaces the modeler would wish to query during simulation. Training a logit model with data from subspace of x_j, then querying in another subspace of x_j, is a statistical extrapolation, which is of course risky and unpredictable.

The continued development and use of the logit model is a testament to its predictive power when sufficient data is available. If studying a single intersection, collecting, analyzing, and storing enough data may be challenging but feasible. For a large or high-dimensional network, the task may be more onerous or unrealistically expensive. In a planning, prediction, or ICM scenario, it may in fact be impossible: one could not collect data from certain network states if the states or infrastructure under consideration are in fact the counterfactual.

In addition, the fact that the logit model describes route choice in terms of utility means that its parameters have no physical meaning, precluding making any sort of "sanity check." This means that hand-tuning the parameters of $J_{ij}^c(\cdot)$ to produce realistic results in the absence of sufficient data is not easy. In a related problem, the abstraction of route utility can lead to a "black-boxing" of the workings of the model. Observed irregularities in simulation behavior can be difficult to track down to the erroneous model parameters or components. These problems, of course, are only exacerbated when the network becomes high-dimensional, with many parameters.

One useful approach to apply the logit model in high-dimensional settings may be estimation of the utility functions J_{ij}^c on a data-rich link or junction, then transplanting these parameters to a similar, but poorly-instrumented, link. The hypothesis would be that their utility functions are similar. Unfortunately, we have some evidence for the instability of the logit model in these extrapolation scenarios from a study in a context of varying sets of available routes by Bliemer and Bovy (2008). They found that many basic and extended logit models are fragile in these situations. If one considers a junction with relevant and irrelevant routes (irrelevant, in this context, means that their utility values J_{ij}^c are small relative to the relevant routes), it turns out that calibrating the model with data collected while the irrelevant routes exist, then querying it without the irrelevant routes (or vice versa) will produce incorrect values of β_{ij}^c. This means that the utility functions J_{ij}^c may in fact not be independent of other output link states.

The example used by Bliemer and Bovy (2008) did not feature the a transplanting of model parameters. Rather, the calibration and querying steps were on the same network; the irrelevant routes were highly circuitous. Still, it is evidence that logit parameters are not, in general, reusable across junctions that have different routes to choose from[2].

What is to be done, then? In Section 4 we present an alternative route choice algorithm for high-dimensional or data-deficient networks. Our algorithm is myopic to the node and agnostic to the link state: it only attempts to balance demand-supply ratios R/S across nodes. Our algorithm is inspired by classic integral control, which has historically been applied to "black-box" problems with success.

[2]Bliemer and Bovy (2008) did find that one extended logit model performed adequately in recreating the proper β. However, this model also encodes the length of road overlap between routes. We would not expect that parameters learned from one junction, then transplanted to another junction with different route overlap structures, would provide good prediction accuracy.

3 A node model for dimensionality management

Our notation for the node model problem has been introduced above. We summarize it here:

- M — number of input links;
- i — index of an input link ($i = 1, \dots, M$);
- N — number of output links;
- j — index of an output link ($j = 1, \dots, N$);
- C — number of commodities;
- c — index of a commodity ($c = 1, \dots, C$);
- S_i^c — how much traffic of commodity c input i wants to send (demand or sending function), $S_i = \sum_{c=1}^{C} S_i^c$;
- p_i — input link priority;
- R_j — how much traffic output j can receive (supply or receive function);
- $\beta_{ij}^c \in [0, 1]$ — split ratios that distribute traffic of a given commodity c coming from input link i between output links, $\sum_{j=1}^{N} \beta_{ij}^c = 1$;
- $S_{ij}^c = \beta_{ij}^c S_i^c$ — oriented demand for commodity c from input link i to output link j, $S_{ij} = \sum_{c=1}^{C} S_{ij}^c$;
- f_{ij}^c — input-output flows that the node model computes, $f_{ij} = \sum_{c=1}^{C} f_{ij}^c$.

This Section is organized as follows. We start by describing 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

To better explain the meaning of input link 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.

Casting the computation of input-output flows f_{i1}^c as a mathematical programming problem, we arrive at:

$$\max \left(\sum_{i=1}^{M} \sum_{c=1}^{C} f_{i1}^c \right), \tag{3.1}$$

subject to:

$$f_{i1}^c \geq 0, \;\; i = 1, \dots, M, \; c = 1, \dots, C \quad \text{— non-negativity constraint;} \tag{3.2}$$

$$f_{i1}^c \leq S_i^c, \;\; i = 1, \dots, M, \; c = 1, \dots, C \quad \text{— demand constraint;} \tag{3.3}$$

$$\sum_{i=1}^{M} f_{i1} \leq R_1 \quad \text{— supply constraint;} \tag{3.4}$$

$$\frac{f_{i1}^c}{f_{i1}} = \frac{S_i^c}{S_i}, \;\; i = 1, \dots, M, \; c = 1, \dots, C, \quad \text{— proportionality constraint for commodity flows;} \tag{3.5}$$

$$\left.\begin{array}{ll} \text{(a)} & p_{i''} f_{i'1} = p_{i'} f_{i''1} \;\; \forall i', i'', \text{ such that } f_{i'1} < S_{i'}, \; f_{i''1} < S_{i''}, \\ \text{(b)} & \text{If } f_{i1} < S_i, \text{ then } f_{i1} \geq \frac{p_i}{\sum_{i'=1}^{M} p_{i'}} R_1. \end{array}\right\} \text{— priority constraint.} \tag{3.6}$$

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. Certainly, other schemes of distributing the input-output flow restrictions between commodity flows may exist: e.g., selected commodities may have a preferential status that always allows them to travel first. Those alternative schemes, however, are beyond the scope of this paper.

Let us discuss 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 2 do not take up more output supply than was allocated for them in cases where category 1 is non-empty. 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 input link $\hat{\imath}$, such that $p_{\hat{\imath}} = 0$, while $f_{\hat{\imath}1} > 0$, then, due to condition (3.6)(a), all input links with non-zero 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 send functions S_i, $i = 1, \dots, M$,[3] then condition (3.6) can be written as an equality constraint:

$$\frac{f_{11}}{S_1} = \cdots = \frac{f_{i1}}{S_i} = \cdots = \frac{f_{M1}}{S_M}, \tag{3.7}$$

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

Remark. 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:

$$f_{11} \leq \max\left\{\frac{p_1}{p_1 + p_2} R_1, \; R_1 - S_2\right\}; \tag{3.8}$$

$$f_{21} \leq \max\left\{\frac{p_2}{p_1 + p_2} R_1, \; R_1 - S_1\right\}. \tag{3.9}$$

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

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

$$f_{11} \leq \max\left\{\frac{p_1}{\sum_{i=1}^3 p_i}R_1, \frac{p_1}{p_1+p_2}\left(R_1 - S_3\right), \frac{p_1}{p_1+p_3}\left(R_1 - S_2\right), R_1 - \sum_{i=2,3} S_i\right\}; \tag{3.10}$$

$$f_{21} \leq \max\left\{\frac{p_2}{\sum_{i=1}^3 p_i}R_1, \frac{p_2}{p_1+p_2}\left(R_1 - S_3\right), \frac{p_2}{p_2+p_3}\left(R_1 - S_1\right), R_1 - \sum_{i=1,3} S_i\right\}; \tag{3.11}$$

$$f_{31} \leq \max\left\{\frac{p_3}{\sum_{i=1}^3 p_i}R_1, \frac{p_3}{p_1+p_3}\left(R_1 - S_2\right), \frac{p_3}{p_2+p_3}\left(R_1 - S_1\right), R_1 - \sum_{i=1,2} S_i\right\}. \tag{3.12}$$

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 present the algorithm for computing input-output flows f_{i1}^c that solves the maximization problem (3.1)-(3.6).

1. Initialize:

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

   $U(k)$ is the set of unprocessed input links: 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 $U(k)$.

3. 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:

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

   - Else, assign:

$$\begin{aligned} f_{i1}^c &= S_i^c \frac{\tilde{p}_i(k)}{\sum_{i' \in U(k)} \tilde{p}_{i'}(k)} \frac{\tilde{R}_1(k)}{S_i}, \quad i \in U(k); \\ U(k+1) &= \emptyset. \end{aligned}$$

4. If $U(k+1) = \emptyset$, then stop.
5. 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

$$f_{11}^c = \min\left\{S_1^c,\ S_1^c \frac{\max\left\{\frac{p_1}{p_1+p_2}R_1,\ R_1 - S_2\right\}}{S_1}\right\}; \tag{3.13}$$

$$f_{21}^c = \min\left\{S_2^c,\ S_2^c \frac{\max\left\{\frac{p_2}{p_1+p_2}R_1,\ R_1 - S_1\right\}}{S_2}\right\}. \tag{3.14}$$

The following theorem can be trivially proved, and we leave the proof to the reader. Further, in Section 3.3, it will become clear that this theorem is a special case of a more general statement.

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

3.2 Single-Input-Multiple-Output (SIMO) Node: Relaxing 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.

To formally describe the mutual influence of output links, 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, $\boldsymbol{\eta}_{jj} \equiv [0, 1]$.
- $\boldsymbol{\eta}_{j'j} = [0, 0]$ (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.

For our example in Figure 1, the mutual restriction intervals take the form of a 3×3 interval matrix:

$$\begin{pmatrix} [0,1] & \left[0, \frac{1}{5}\right] & [0,0] \\ [0,1] & [0,1] & [0,1] \\ [0,0] & \left[\frac{3}{5}, 1\right] & [0,1] \end{pmatrix}.$$

Suppose, $\frac{R_1}{S_{11}} < \frac{R_3}{S_{13}} < 1 \leq \frac{R_2}{S_{12}}$. 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:

$$f_{11} = R_1 \quad \text{and} \quad f_{13} = R_3.$$

Flow f_{12} is partially restricted by both output links 1 and 3:

$$f_{12} = (1 - |\boldsymbol{\eta}_{12}| - |\boldsymbol{\eta}_{32}|)\, S_{12} + \frac{R_1}{S_{11}}|\boldsymbol{\eta}_{12}|S_{12} + \frac{R_3}{S_{13}}|\boldsymbol{\eta}_{32}|S_{12},$$

where $|\boldsymbol{\eta}_{12}|$ and $|\boldsymbol{\eta}_{32}|$ denote lengths of intervals $\boldsymbol{\eta}_{12}$ and $\boldsymbol{\eta}_{32}$ respectively. In this expression for f_{12} 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_{12} can be rewritten as:

$$f_{12} = S_{12} - \left(1 - \frac{R_1}{S_{11}}\right)|\boldsymbol{\eta}_{12}|S_{12} - \left(1 - \frac{R_3}{S_{13}}\right)|\boldsymbol{\eta}_{32}|S_{12},$$

which corresponds to the striped area in Figure 3(a), while the second and the third terms of the right hand side correspond to the areas of the grayed out rectangles that represent restrictions imposed on flow f_{12} by outputs 1 and 3 respectively.

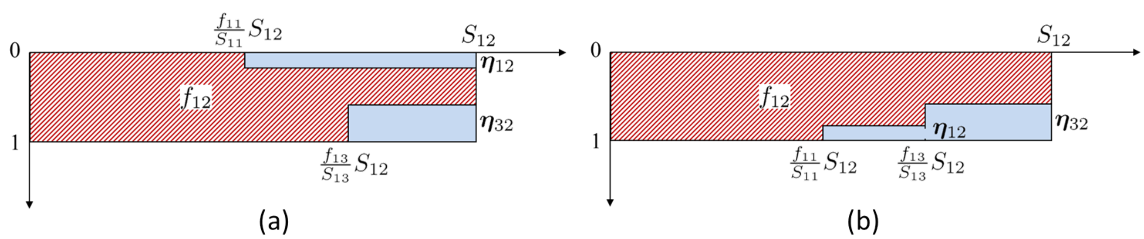

Figure 3: Computing f_{12} for the 1-input-3-output node example with $\frac{R_1}{S_{11}} < \frac{R_3}{S_{13}} < 1 \leq \frac{R_2}{S_{12}}$, $\boldsymbol{\eta}_{13} = \boldsymbol{\eta}_{31} = [0, 0]$, $\boldsymbol{\eta}_{32} = \left[\frac{3}{5}, 1\right]$ and two cases for η_{12}: (a) $\eta_{12} = \left[0, \frac{1}{5}\right]$; (b) $\eta_{12} = \left[\frac{4}{5}, 1\right]$.

We come to an important point of why we use intervals, not just scalars, to represent mutual restrictions of output links. Suppose, in our example $\boldsymbol{\eta}_{12} = \left[\frac{4}{5}, 1\right]$. Then, restrictions from outputs 1 and 3 imposed on flow f_{12} would overlap, and their summary effect would be smaller. 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. Thus, the expression for f_{12} in this case will be:

$$f_{12} = S_{12} - \left(1 - \frac{R_1}{S_{11}}\right)|\boldsymbol{\eta}_{12}|S_{12} - \left(1 - \frac{R_3}{S_{13}}\right)(|\boldsymbol{\eta}_{32}| - |\boldsymbol{\eta}_{12}|)\, S_{12},$$

where the second term of the right hand side corresponds to the restriction imposed by the output 1, as before, and the third term corresponds to the *additional* restrictive effect imposed by the output 3. This expression is illustrated by Figure 3(b).

Define:

$$\mathcal{Q}_{j'j} = \boldsymbol{\eta}_{j'j} \times \left[\frac{f_{1j'}}{S_{1j'}}S_{1j}, S_{1j}\right], \tag{3.15}$$

where '$\times$' denotes a Cartesian product. The Cartesian product of the two intervals gives us a rectangle $\mathcal{Q}_{j'j}$. In Figure 3 $\mathcal{Q}_{12}$ and $\mathcal{Q}_{32}$ represent the grayed out rectangles.

Denote $\mathcal{A}(\cdot)$ an area of a two-dimensional shape. Then, the expressions for flow f_{12} for different values of $\boldsymbol{\eta}_{12}$ in our example can be replaced with a single, more general formula:

$$f_{12} = S_{12} - \mathcal{A}\left(\mathcal{Q}_{12} \cup \mathcal{Q}_{32}\right).$$

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

$$\max\left(\sum_{j=1}^{N}\sum_{c=1}^{C} f_{1j}^c\right), \tag{3.16}$$

subject to:

$$f_{1j}^c \geq 0, \quad j = 1, \ldots, N, \; c = 1, \ldots, C \text{ — non-negativity constraint;} \tag{3.17}$$

$$f_{1j}^c \leq S_{1j}^c, \quad j = 1, \ldots, N, \; c = 1, \ldots, C \text{ — demand constraint;} \tag{3.18}$$

$$f_{1j} \leq R_j, \quad j = 1, \ldots, N \text{ — supply constraint;} \tag{3.19}$$

$$\frac{f_{ij}^c}{f_{1j}} = \frac{S_{1j}^c}{S_{1j}}, \quad j = 1, \ldots, N, \; c = 1, \ldots, C \text{ — proportionality constraint for commodity flows;} \tag{3.20}$$

$$f_{1j} \leq S_{ij} - \mathcal{A}\left(\bigcup_{j' \neq j} Q_{j'j}\right), \quad j = 1, \ldots, N \text{ — relaxed FIFO constraint.} \tag{3.21}$$

For SIMO nodes with full FIFO, constraint (3.21) together with the supply constraint (3.19) translates into:

$$f_{1j} \leq S_{1j} - \left(1 - \frac{f_{1j^*}}{S_{1j^*}}\right) S_{1j} = \frac{f_{1j^*}}{S_{1j^*}} S_{1j}, \tag{3.22}$$

where

$$j^* = \arg\min_j \frac{R_{j^*}}{S_{1j^*}}, \tag{3.23}$$

and, since we are solving the flow maximization problem, (3.22) can be replaced with the equality constraint:

$$f_{1j} = \frac{f_{1j^*}}{S_{1j^*}} S_{1j}. \tag{3.24}$$

For SIMO nodes with no FIFO, $\mathcal{A}\left(\bigcup_{j' \neq j} \mathcal{Q}_{j'j}\right) = 0$, which simplifies (3.21) to the demand constraint, and thus, constraint (3.21) can be omitted.

Next, we present the algorithm for solving the flow maximization problem (3.16)-(3.21).

1. Initialize:

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

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

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

$$\alpha_j(k) = \frac{R_j}{\tilde{S}_{1j}(k)}, \tag{3.25}$$

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

$$j^*(k) = \arg\min_{j \in V(k)} \alpha_j(k). \tag{3.26}$$

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

$$\begin{aligned} f_{1j}^c &= \tilde{S}_{1j}^c(k), \quad \forall j \in V(k), \quad c = 1, \ldots, C; \\ V(k+1) &= \emptyset. \end{aligned} \tag{3.27}$$

- Else, assign:

$$\tilde{S}_{1j^*}^c(k+1) = \alpha_{j^*}(k)\tilde{S}_{1j}^c(k), \quad c = 1, \ldots, C; \tag{3.28}$$

$$\tilde{S}_{1j}^c(k+1) = \frac{\tilde{S}_{1j}(k+1)}{S_{1j}} S_{1j}^c, \quad j \in V(k) \setminus \{j^*\}, \quad c = 1, \ldots, C, \tag{3.29}$$

where

$$\tilde{S}_{1j}(k+1) = \tilde{S}_{1j}(k) - S_{1j}\left(|\boldsymbol{\eta}_{j^*j}| - |\tilde{\boldsymbol{\eta}}_j(k) \cap \boldsymbol{\eta}_{j^*j}|\right)\left(1 - \frac{\sum_{c=1}^C \tilde{S}_{1j^*}^c(k+1)}{S_{1j^*}}\right), \tag{3.30}$$

$$\begin{aligned} \tilde{\boldsymbol{\eta}}_j(k+1) &= \tilde{\boldsymbol{\eta}}_j(k) \cup \boldsymbol{\eta}_{j^*j}, \quad j \in V(k); \\ f_{1j}^c &= \tilde{S}_{ij}^c(k+1), \quad j : \tilde{\boldsymbol{\eta}}_j(k+1) = [0,1], \quad c = 1, \ldots, C; \\ V(k+1) &= V(k) \setminus \{j : \tilde{\boldsymbol{\eta}}_j(k+1) = [0,1]\}, \end{aligned} \tag{3.31}$$

where $|\tilde{\boldsymbol{\eta}}_j(k) \cap \boldsymbol{\eta}_{j^*j}|$ denotes the measure of the interval intersection.[4]

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

This algorithm takes no more than N iterations to complete.

Finally, we state the result of this Section as a theorem.

Theorem 3.2. *The SIMO input-output flow computation algorithm constructs the unique solution of the maximization problem* (3.16)-(3.21).

Remark. 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 an intersection 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. In other words, $\boldsymbol{\eta}_{54}^1 = [0, 1]$. 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 $\boldsymbol{\eta}_{54}^2 = [1 - \epsilon, 1]$, with some $0 < \epsilon < 1$. So, from now on we will write $\boldsymbol{\eta}_{j'j}^i$ and $\mathcal{Q}_{j'j}^i$with index i identifying the input link.

[4]This set may be disjoint.

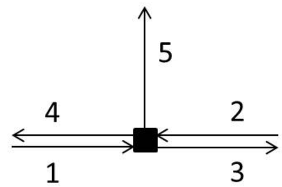

Figure 4: Intersection node.

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 (3.16)-(3.21). While the generalization of the objective (3.1), (3.16) and constraints (3.2)-(3.5), (3.17)-(3.20) is straight forward, extending the priority constraint (3.6) and the relaxed FIFO constraint (3.21) to the MIMO case requires a little more work.

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

If the output j^* is restricting for input links i' and i'', then, according to this definition, $p_{i''j^*} f_{i'j^*} = p_{i'j^*} f_{i''j^*}$.

Define the set of restricting outputs for input link i:

$$W_i = \{j^* : \ \ S_{ij^*} > 0; \ \ p_{i'j^*} f_{ij^*} \geq p_{ij^*} f_{i'j^*}, \ \ \forall i' \neq i\}. \tag{3.32}$$

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

$$\max \left(\sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{c=1}^{C} f_{ij}^{c} \right), \tag{3.33}$$

subject to:

$$f_{ij}^{c} \geq 0, \ \ i = 1, \ldots, M, \ j = 1, \ldots, N, \ c = 1, \ldots, C \ \ \text{— non-negativity constraint;} \tag{3.34}$$

$$f_{ij}^{c} \leq S_{ij}^{c}, \ \ i = 1, \ldots, M, \ j = 1, \ldots, N, \ c = 1, \ldots, C \ \ \text{— demand constraint;} \tag{3.35}$$

$$\sum_{i=1}^{M} f_{ij} \leq R_j, \ \ j = 1, \ldots, N \ \ \text{— supply constraint;} \tag{3.36}$$

$$\frac{f_{ij}^{c}}{f_{ij}} = \frac{S_{ij}^{c}}{S_{ij}}, \ \ i = 1, \ldots, M, \ j = 1, \ldots, N, c = 1, \ldots, C \ \ \text{— proportionality constraint for commodity flows;} \tag{3.37}$$

$$\left. \begin{array}{ll} \text{(a)} & \text{For each input link } i \text{, such that} \\ & \sum_{j=1}^{N} f_{ij} < S_i, \ \ W_i \neq \emptyset; \\ \text{(b)} & \text{For each input link } i \text{, such that } W_i \neq \emptyset, \\ & f_{ij} \geq \frac{p_{ij}}{\sum_{i'=1}^{M} p_{i'j}} R_j, \ \ \forall j \in W_i. \end{array} \right\} \text{— priority constraint;} \tag{3.38}$$

$$f_{ij} \leq S_{ij} - \mathcal{A}\left(\bigcup_{j' \in W_i \setminus \{j\}} \mathcal{Q}_{j'j}^{i} \right), \ \ i = 1, \ldots, M, \ j = 1, \ldots, N \ \ \text{— relaxed FIFO constraint.} \tag{3.39}$$

Constraint (3.38) generalizes the MISO priority constraint (3.6). For each output link j, $j = 1, \dots, N$, flows $\sum_{c=1}^{C} f_{ij}^c$ fall into two categories: (1) restricted by *this* output link; and (2) not restricted by *this* output link. Condition (3.38)(a) states that if the flow from an input i is supply-constrained, there do exist output links j, such that flows f_{ij} 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 (3.38)(b) ensures that input-output flows of category 2 do not take up more output supply than was allocated for them when category 1 is non-empty. For a SIMO ($M = 1$) node, where there is no competition between input flows, constraint (3.38) is satisfied automatically.

Constraint (3.39) generalizes the SIMO relaxed FIFO constraint (3.21). Here,

$$\mathcal{Q}_{j'j}^i = \boldsymbol{\eta}_{j'j} \times \left[\frac{f_{ij'}}{S_{ij'}} S_{ij}, S_{ij}\right], \tag{3.40}$$

which is a generalization of (3.15), and $\mathcal{A}(\cdot)$ denotes the area of a two-dimensional object. For MIMO nodes with full FIFO, constraint (3.39) together with the supply constraint (3.36) translates to:

$$f_{ij} \leq \min_{j' \in W_i} \left\{\frac{f_{ij'}}{S_{ij'}}\right\} S_{ij}, \tag{3.41}$$

and, since we are solving the flow maximization problem, (3.41) can be replaced with the equality constraint:

$$f_{ij} = \min_{j^* \in W_i} \frac{f_{ij^*}}{S_{ij^*}} S_{ij}. \tag{3.42}$$

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 (3.39) degenerates into the demand constraint.

Next, we present the algorithm that solves the flow maximization problem (3.33)-(3.39)

1. Initialize:

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

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:

$$\tilde{p}_i(k) = \begin{cases} p_i, & \text{if there exists } i' \in \bigcup_{j \in V(k)} U_j(k) : \; p_{i'} > 0, \\ \frac{1}{\left|\bigcup_{j \in V(k)} U_j(k)\right|}, & \text{otherwise,} \end{cases} \tag{3.43}$$

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:

$$\tilde{p}_{ij}(k) = \tilde{p}_i(k)\frac{\sum_{c=1}^{C} S_{ij}^c}{\sum_{c=1}^{C} S_i^c}. \tag{3.44}$$

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

$$a_j(k) = \frac{\tilde{R}_j(k)}{\sum_{i\in U_j(k)} \tilde{p}_{ij}(k)}, \tag{3.45}$$

and find the smallest of these factors:

$$a_{j^*}(k) = \min_{j\in V(k)} a_j(k). \tag{3.46}$$

The link j^* has the most restricted supply of all output links.

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

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

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

$$\begin{aligned} f_{ij}^c &= \tilde{S}_{ij}^c(k), \quad i \in \tilde{U}(k), \quad c = 1,\ldots,C; \\ \tilde{R}_j(k+1) &= \tilde{R}_j(k) - \sum_{i\in\tilde{U}(k)} f_{ij}; \\ U_j(k+1) &= U_j(k) \setminus \tilde{U}(k). \end{aligned} \tag{3.47}$$

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

$$\tilde{S}_{ij^*}^c(k+1) = \tilde{S}_{ij^*}^c(k)\frac{\tilde{p}_{ij^*}(k)a_{j^*}(k)}{\tilde{S}_{ij^*}(k)}, \quad c = 1,\ldots,C; \tag{3.48}$$

$$\tilde{S}_{ij}^c(k+1) = \tilde{S}_{ij}^c(k)\frac{\tilde{S}_{ij}(k+1)}{\tilde{S}_{ij}(k)}, \quad i \in U_j(k)\cap U_{j^*}(k), \quad j \in V(k)\setminus\{j^*\}, \tag{3.49}$$

where

$$\tilde{S}_{ij}(k+1) = \tilde{S}_{ij}(k) - S_{ij}\left(|\boldsymbol{\eta}_{j^*j}^i| - |\tilde{\boldsymbol{\eta}}_j^i(k)\cap\boldsymbol{\eta}_{j^*j}^i|\right)\left(1 - \frac{\sum_{c=1}^{C}\tilde{S}_{ij^*}^c(k+1)}{S_{ij^*}}\right); \tag{3.50}$$

$$\begin{aligned} \tilde{S}_{ij}^c(k+1) &= \tilde{S}_{ij}^c(k), \quad i \notin U_j(k)\cap U_{j^*}(k); \\ \tilde{\boldsymbol{\eta}}_j^i(k+1) &= \tilde{\boldsymbol{\eta}}_j^i(k) \cup \boldsymbol{\eta}_{j^*j}^i; \\ f_{ij}^c &= \tilde{S}_{ij}^c(k+1), \\ & \quad i \in U_j(k)\cap U_{j^*}(k),\ j \in V(k):\ \tilde{\boldsymbol{\eta}}_j^i(k+1) = [0,1],\ c = 1,\ldots,C; \\ \tilde{R}_j(k+1) &= \tilde{R}_j(k) - \sum_{i\in U_{j^*}(k):\boldsymbol{\eta}_j^i(k+1)=[0,1]} f_{ij}; \\ U_j(k+1) &= U_j(k)\setminus\left\{i \in U_{j^*}(k):\ \tilde{\boldsymbol{\eta}}_j^i(k+1) = [0,1]\right\}. \end{aligned} \tag{3.51}$$

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

This algorithm takes no more than $(M+N-2)$ iterations to complete.

The following lemma states that in the case of $N = 1$, the MIMO algorithm produces exactly 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 (3.45)-(3.46) result in $a_{j^*}(k) = a_1(k) = \frac{\tilde{R}_1(k)}{\sum_{i \in U_1(k)} \tilde{p}_i(k)}$ and $j^* = 1$. □

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

Lemma 3.2. *The SIMO algorithm with 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 (3.45), reduce to:

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

and

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

□

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

Theorem 3.3. *Given a set of input links* $i \in 1, \ldots, M$, *output links* $j \in 1, \ldots, N$, *commodities* $c \in 1, \ldots, C$, *priorities* p_i, *split ratios* β_i, *and mutual restriction coefficients* $\{\eta^i_{j,j'}\}$, *the algorithm of Section 3.3 obtains the unique solution of the optimization problem* (3.33)-(3.39).

Proof. The priority constraint (3.38) 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 (3.33)-(3.39). 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 $\{f^c_{ij}\}$. The C flows for each (i, j) are further constrained by our commodity flow proportionality constraint (3.37); solving for one of $\{f^1_{ij}, \ldots, f^C_{ij}\}$ also finds them all. Our task then becomes finding optimal values for each of $M \cdot N$ subsets $\{f^1_{ij}, \ldots, f^C_{ij}\}$. Our algorithm finds at least one of these $M \cdot N$ subsets per iteration k. The assignments are done by either equation (3.47) or equation (3.51). Over iterations, subsets are assigned to build up the unique optimum solution. We can show that each subset assigned is optimal; that is, at least one of the constraints is tight.

Consider formulae (3.50)-(3.51), our implementation of the relaxed FIFO constraint. In step 5 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 relaxed FIFO effect instigated by j^* as the most restrictive link. For a generic $j \neq j^*$, equation (3.50) enforces the relaxed FIFO constraint by decaying the oriented demand $\tilde{S}_{ij}$. In fact, this modified oriented demand acts as a proxy for the relaxed FIFO constraint. Since a f^c_{ij} that is restricted by relaxed FIFO will, by construction, never obtain $f^c_{ij} = S^c_{ij}$, the running quantity $\tilde{S}^c_{ij}(k)$ represents an "effective" oriented demand after application of relaxed FIFO. A flow that is not constrained by relaxed FIFO will nevertheless not exceed its demand, as $\tilde{S}^c_{ij}(0) = S^c_{ij}$, and $\tilde{S}^c_{ij}(k)$ only decreases across k as relaxed FIFO constraints are considered (ensuring compliance with the demand constraint (3.35)). On the other hand, if it turns out that a flow is found $f^c_{ij} = \tilde{S}^c_{ij}(k) \leq S^c_{ij}$ for some k, then this flow value may have an some constraint imposed by relaxed FIFO.

Flows assigned by equation (3.51) for $j \neq j^*$ are constrained by a relaxed FIFO constraint (3.39). To see this, note that

$$\mathcal{A}(\mathcal{Q}^i_{j'j}) = \left|\eta^i_{j'j}\right| \left(1 - \frac{\sum_c f^c_{ij'}}{S_{ij'}}\right) S_{ij} \tag{3.52}$$

where, recall, $\mathcal{Q}^i_{j'j}$ is defined by (3.40). As applied in (3.50), $j' = j^*$ always, and $f^c_{ij^*} = \tilde{S}^c_{ij^*}$ by (3.51).

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

$$\mathcal{A}\left(\mathcal{Q}^i_{j'j} \cup \mathcal{Q}^i_{j''j}\right) = \mathcal{A}\left(\mathcal{Q}^i_{j'j}\right) + \mathcal{A}\left(\mathcal{Q}^i_{j''j}\right) - \mathcal{A}\left(\mathcal{Q}^i_{j'j} \cap \mathcal{Q}^i_{j''j}\right) \tag{3.53}$$

and

$$\mathcal{A}\left(\mathcal{Q}^i_{j'j} \cap \mathcal{Q}^i_{j''j}\right) = \left(\left|\eta^i_{j'j} \cap \eta^i_{j''j}\right|\right) \min_{j^\sharp \in \{j', j''\}} \left(1 - \frac{\sum_c f^c_{ij^\sharp}}{S_{ij^\sharp}}\right) S_{ij}. \tag{3.54}$$

Note that $j^\sharp$ as defined in (3.54) 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 then those in previous iterations. Equation (3.50) thus takes the union (with the subtraction of the intersection) as done in (3.53)-(3.54), and incorporates the relaxed FIFO constraint (3.39).

Now consider equations (3.48) and (3.51), which apply in the special case where $j = j^*$. have

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

so all the flows into link j^* take up all available supply. These flows are thus constrained by the supply constraint, (3.36). □

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

Remark 3.1. *Note that setting the priorities equal to input link capacities, $p_i = F_i$, and all restriction intervals $\eta^i_{j'j} = [0, 1]$, we recover the original node model and algorithm of Tampère et al. (2011).*

4 Stable and nonparametric traffic assignment

In this section we consider a MIMO node with M input links, N output links and C commodities, where some of the split ratios β^c_{ij} are not defined a priori and must be computed as functions of the input demand S^c_i, priorities p_i and the output supply R_j, $i = 1, \ldots, M$, $j = 1, \ldots, N$ and $c = 1, \ldots, C$. This may occur if the modeler chooses to let drivers at a node select a route to their destination in response to changing conditions; a specific example of such a node is given in an example describing an interchange between a freeway and a managed lane to be discussed after presentation of the algorithm. Here we present the algorithm for computing undefined split ratios based on the following definitions and assumptions:

- Define the set of commodity movements, for which split ratios are known, as $\mathcal{B} = \left\{\{i,j,c\} : \beta_{ij}^c \in [0,1]\right\}$, and the set of commodity movements, for which split ratios are to be computed, as $\overline{\mathcal{B}} = \left\{\{i,j,c\} : \beta_{ij}^c \text{ are unknown}\right\}$.
- For a given input link i and commodity c such that $S_i^c = 0$, assume that all split ratios are known: $\{i,j,c\} \in \mathcal{B}$.[5]
- Define the set of output links, for which there exist unknown split ratios, as $V = \left\{j : \exists \{i,j,c\} \in \overline{\mathcal{B}}\right\}$.
- Assuming that for a given input link i and commodity c split ratios must sum up to 1, define the unassigned portion of flow by $\overline{\beta}_i^c = 1 - \sum_{j:\{i,j,c\}\in\mathcal{B}} \beta_{ij}^c$.
- For a given input link i and commodity c such that there exist $\{i,j,c\} \in \overline{\mathcal{B}}$, assume $\overline{\beta}_i^c > 0$, otherwise the undefined split ratios can be trivially set to 0.
- For every output link $j \in V$, define the set of input links that have an unassigned demand portion directed toward this output link, by $U_j = \left\{i : \exists \{i,j,c\} \in \overline{\mathcal{B}}\right\}$.
- For a given input link i and commodity c define the set of output links, split ratios for which are to be computed by $V_i^c = \{j : \exists i \in U_j\}$, and assume that if nonempty, this set contains at least two elements, otherwise a single split ratio can be trivially set to $\overline{\beta}_i^c$.
- Assume that input link priorities are nonnegative, $p_i \geq 0$, $i = 1, \ldots, M$, and $\sum_{i=1}^M p_i = 1$. Note that, although in section 3 we did not require the input priorities to sum to one, in this section we assume this normalization is done to simplify the notation.
- Define the set of input links with zero priority: $U_{zp} = \{i : p_i = 0\}$. To avoid dealing with zero input priorities, perform regularization:

  $$\tilde{p}_i = p_i\left(1 - \frac{|U_{zp}|}{M}\right) + \frac{1}{M}\frac{|U_{zp}|}{M} = p_i\frac{M - |U_{zp}|}{M} + \frac{|U_{zp}|}{M^2}, \tag{4.1}$$

  where $|U_{zp}|$ denotes the number of elements in set U_{zp}. Expression (4.1) implies that the regularized input priority $\tilde{p}_i$ consists of two parts: (1) the original input priority p_i normalized to the portion of input links with nonzero priorities; and (2) uniform distribution among M input links, $\frac{1}{M}$, normalized to the portion of input links with zero priorities.

  Note that $\tilde{p}_i \geq 0$, $i = 1, \ldots, M$, and $\sum_{i=1}^M \tilde{p}_i = 1$.

The algorithm for distributing $\overline{\beta}_i^c$ among the commodity movements in $\overline{\mathcal{B}}$, that is assigning values to the a priori unknown split ratios, aims at maintaining output links as uniformly occupied as possible. It is described next.

1. Initialize:

   $$\begin{aligned}
   \tilde{\beta}_{ij}^c(0) &:= \begin{cases} \beta_{ij}^c, & \text{if } \{i,j,c\} \in \mathcal{B}, \\ 0, & \text{otherwise;} \end{cases} \\
   \overline{\beta}_i^c(0) &:= \overline{\beta}_i^c; \\
   \tilde{U}_j(0) &= U_j; \\
   \tilde{V}(0) &= V; \\
   k &:= 0,
   \end{aligned}$$

   Here $\tilde{U}_j(k)$ is the remaining set of input links with some unassigned demand, which may be directed to output link j; and $\tilde{V}(k)$ is the remaining set of output links, to which the still unassigned demand may be directed.

[5] If split ratios were undefined in this case, they could be assigned arbitrarily.

2. If $\tilde{V}(k) = \emptyset$, stop. The sought-for split ratios are $\left\{\tilde{\beta}_{ij}^c(k)\right\}$, $i = 1, \ldots, M$, $j = 1, \ldots, N$, $c = 1, \ldots, C$.

3. Calculate the remaining unallocated demand:

$$\overline{S}_i^c(k) = \overline{\beta}_i^c(k) S_i^c, \quad i = 1, \ldots, M, \quad c = 1, \ldots, C.$$

4. For all input-output link pairs calculate oriented demand as introduced in Section 3:

$$\tilde{S}_{ij}^c(k) = \tilde{\beta}_{ij}^c(k) S_i^c.$$

5. For all input-output link pairs calculate oriented priorities:

$$\tilde{p}_{ij}(k) = \tilde{p}_i \frac{\sum_{c=1}^C \gamma_{ij}^c S_i^c}{\sum_{c=1}^C S_i^c} \tag{4.2}$$

with

$$\gamma_{ij}^c(k) = \begin{cases} \beta_{ij}^c, & \text{if split ratio is defined a priori: } \{i, j, c\} \in \mathcal{B}, \\ \tilde{\beta}_{ij}^c(k) + \frac{\overline{\beta}_i^c(k)}{|V_i^c|}, & \text{otherwise,} \end{cases} \tag{4.3}$$

where $|V_i^c|$ denotes the number of elements in the set V_i^c. Comparing the expression (4.2)-(4.3) with (3.44), one can see that split ratios $\tilde{\beta}_{ij}^c(k)$, which are not fully defined yet, are complemented with a fraction of $\overline{\beta}_i^c(k)$ inversely proportional to the number of output links among which the flow of commodity c from input link i can be distributed.

Note that in this step we are using *regularized* priorities $\tilde{p}_i$ as opposed to the original p_i, $i = 1, \ldots, M$. This is done to ensure that inputs with $p_i = 0$ are not ignored in the split ratio assignment.

6. Find the largest oriented demand-supply ratio:

$$\mu^+(k) = \max_j \max_i \frac{\sum_{c=1}^C \tilde{S}_{ij}^c(k)}{\tilde{p}_{ij}(k) R_j} \sum_{i \in U_j} \tilde{p}_{ij}(k).$$

7. Define the set of all output links, where the minimum of the oriented demand-supply ratio is achieved:

$$Y(k) = \arg \min_{j \in \tilde{V}(k)} \min_{i \in \tilde{U}_j(k)} \frac{\sum_{c=1}^C \tilde{S}_{ij}^c(k)}{\tilde{p}_{ij}(k) R_j} \sum_{i \in U_j} \tilde{p}_{ij}(k),$$

and from this set pick the output link j^- with the smallest output demand-supply ratio (when there are multiple minimizing output links, any of the minimizing output links may be chosen as j^-):

$$j^- = \arg \min_{j \in Y(k)} \frac{\sum_{i=1}^M \sum_{c=1}^C \tilde{S}_{ij}^c(k)}{R_j}.$$

8. Define the set of all input links, where the minimum of the oriented demand-supply ratio for the output link j^- is achieved:

$$W_{j^-}(k) = \arg \min_{i \in \tilde{U}_{j^-}(k)} \frac{\sum_{c=1}^C \tilde{S}_{ij^-}^c(k)}{\tilde{p}_{ij^-}(k) R_{j^-}} \sum_{i \in U_{j^-}} \tilde{p}_{ij^-}(k),$$

and from this set pick the input link i^- and commodity c^- with the smallest remaining unallocated demand:

$$\{i^-, c^-\} = \arg \min_{\substack{i \in W_{j^-}(k), \\ c : \overline{\beta}_{i^-}^c(k) > 0}} \overline{S}_i^c(k).$$

9. Define the smallest oriented demand-supply ratio:

$$\mu^-(k) = \frac{\sum_{c=1}^{C} \tilde{S}^c_{i^-j^-}(k)}{\tilde{p}_{i^-j^-}(k) R_{j^-}} \sum_{i \in U_{j^-}} \tilde{p}_{ij^-}(k).$$

   - If $\mu^-(k) = \mu^+(k)$, which means that the oriented demand is perfectly balanced among the output links, distribute the unassigned demand proportionally to the allocated supply:

$$\tilde{\beta}^{c^-}_{i^-j}(k+1) = \tilde{\beta}^{c^-}_{i^-j}(k) + \frac{\tilde{p}_{i^-j}(k) R_j}{\sum_{j' \in \tilde{V}^{c^-}_{i^-}(k)} \tilde{p}_{i^-j'}(k) R_{j'}} \overline{\beta}^{c^-}_{i^-}(k), \quad j \in \tilde{V}^{c^-}_{i^-}(k); \tag{4.4}$$

$$\tilde{\beta}^c_{ij}(k+1) = \tilde{\beta}^c_{ij}(k) \text{ for } \{i,j,c\} \neq \{i^-, j, c^-\}; \tag{4.5}$$

$$\overline{\beta}^{c^-}_{i^-}(k+1) = 0;$$

$$\overline{\beta}^{c}_{i}(k+1) = \overline{\beta}^{c}_{i}(k) \text{ for } \{i,c\} \neq \{i^-, c^-\}.$$

   - Else, assign:

$$\Delta\tilde{\beta}^{c^-}_{i^-j^-}(k) = \min\left\{ \overline{\beta}^{c^-}_{i^-}(k), \left(\frac{\mu^+(k)\tilde{p}_{i^-j^-}(k) R_{j^-}}{\overline{S}^{c^-}_{i^-}(k) \sum_{i \in U_{j^-}} \tilde{p}_{ij^-}(k)} - \frac{\sum_{c=1}^{C} \tilde{S}^c_{i^-j^-}(k)}{\overline{S}^{c^-}_{i^-}(k)} \right) \right\}; \tag{4.6}$$

$$\tilde{\beta}^{c^-}_{i^-j^-}(k+1) = \tilde{\beta}^{c^-}_{i^-j^-}(k) + \Delta\tilde{\beta}^{c^-}_{i^-j^-}(k); \tag{4.7}$$

$$\tilde{\beta}^c_{ij}(k+1) = \tilde{\beta}^c_{ij}(k) \text{ for } \{i,j,c\} \neq \{i^-, j^-, c^-\}; \tag{4.8}$$

$$\overline{\beta}^{c^-}_{i^-}(k+1) = \overline{\beta}^{c^-}_{i^-}(k) - \Delta\tilde{\beta}^{c^-}_{i^-j^-}(k);$$

$$\overline{\beta}^{c}_{i}(k+1) = \overline{\beta}^{c}_{i}(k) \text{ for } \{i,c\} \neq \{i^-, c^-\}.$$

10. Update sets $\tilde{U}_j(k)$ and $\tilde{V}(k)$:

$$\tilde{U}_j(k+1) = \tilde{U}_j(k) \setminus \left\{ i^- : \overline{\beta}^c_{i^-}(k+1) = 0, \ c = 1, \ldots, C \right\}, \quad j \in \tilde{V}(k);$$

$$\tilde{V}(k+1) = \tilde{V}(k) \setminus \left\{ j : \tilde{U}_j(k+1) = \emptyset \right\}.$$

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

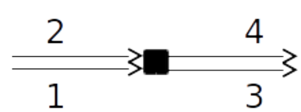

Figure 5: Merge-diverge intersection example.

Example. Consider the node presented in Figure 5. The node represents a junction between two parallel links, or a merge-diverge. Nodes such as these may be used to model locations where drivers can choose to shift between two parallel roads or lane groups; the canonical example is an access point for a managed lane facility adjacent to a freeway, where drivers may choose to enter or exit the managed lane. This is an example of a possibly data-poor situation we discussed earlier where split ratios can be hard to measure, but it makes sense to dynamically model drivers' split ratios in response to local conditions.

In this example, we will say that links 1 and 3 represent a freeway's general purpose (GP) lanes, and links 2 and 4 represent a high-occupancy vehicle (HOV) lane, open only to a select group of vehicles. The Low-Occupancy Vehicles (LOVs) and HOVs will be modeled as separate commodities, with notations of L for LOVs and H for HOVs.

Suppose we have input parameters of:

$$\begin{array}{lll}
\beta_{14}^L = 0 & \beta_{13}^L = 1 & \\
\beta_{14}^H = ? & \beta_{13}^H = ? & p_1 = 3/4 \\
\beta_{23}^H = ? & \beta_{24}^H = ? & p_2 = 1/4 \\
S_1^L = 500 & S_1^H = 100 & R_3 = 600 \\
S_2^H = 50 & & R_4 = 200
\end{array}$$

In words, the LOVs are not allowed to enter the HOV lane; HOVs are allowed, but individual vehicles may or may not choose to do so: both links can bring them to their eventual downstream destination so the choice of link to take will be made in response to immediate local congestion conditions.

Let us outline how the algorithm would assign split ratios for the HOVs.

$\underline{k = 0:}$ $\Delta\tilde{\beta}_{i^- j^-}^{c^-}(0) = \Delta\tilde{\beta}_{24}^H(0) = 1$

$$\tilde{\beta}_{24}^H(1) = 0 + 1 = 1$$

$$\overline{\beta}_2^H(1) = 1 - 0 = 0$$

$\underline{k = 1:}$ $\Delta\tilde{\beta}_{i^- j^-}^{c^-}(1) = \Delta\tilde{\beta}_{14}^H(1) = 1/3$

$$\tilde{\beta}_{14}^H(2) = 0 + 1/3 = 1/3$$

$$\overline{\beta}_2^H(2) = 1 - 1/3 = 2/3$$

$\underline{k = 2:}$ $\tilde{\beta}_{13}^H(3) = 0.64$

$$\tilde{\beta}_{23}^H(3) = 0.36$$

$$\overline{\beta}_1^H(3) = 0$$

$\underline{k = 3:}$

The algorithm terminates. The resulting split ratios are $\boldsymbol{\beta_{13}^H = 0.64}$, $\boldsymbol{\beta_{23}^H = 0.36}$, $\boldsymbol{\beta_{24}^H = 1}$, $\boldsymbol{\beta_{23}^H = 0}$.

5 Conclusion

This paper discussed several modeling issues that the authors have encountered in macroscopic simulation of large and/or high-dimensional road networks. Two issues, unrealistic spillback behavior and making sensible split ratios with a dearth of data, were traced back to issues with common node and route choice models. As discussed, the FIFO flow constraint shared by all node models in the style of Tampère et al. (2011) can cause overly aggressive spillback, and the statistical nature of the logit model means that it cannot be expected to perform well when the data are lacking, and "hand-tuning" or transferring data can be difficult as well.

We proposed a generalization of the FIFO flow rule through the use of mutual restriction intervals. These intervals allow for a range of relaxations of the FIFO rule, on a spectrum from no-FIFO to full-FIFO. We also presented an easy graphical method for calculating flows from supplies and demands. We also presented an integral-control-inspired split ratio assignment algorithm that can produce split ratios at a node with access only to the local demands and supplies. We have had success in applying this method in situations where no data to fit a logit model is available.

The above mentioned results are 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.

This research was funded by the California Department of Transportation.

A Notation

Symbol	Definition	Used in section(s)
k	Iteration index	3,4
i	Index of links entering a node	3,4
M	Number of links entering a node	3,4
j	Index of links exiting a node	3,4
N	Number of links exiting a node	3,4
F_l	Capacity of link l	3
$\tilde{S}_l^c(k)$	Adjusted demand for commodity c of link l as of iteration k	3
R_l	Supply of link l	3
p_i	Priority of input link i	3,4
$U(k)$	Set of input links whose flows have yet to be fully determined as of iteration k	3
$V(k)$	Set of output links whose flows have yet to be fully determined as of iteration k	3
$\tilde{p}_i(k)$	Adjusted priority of link i at iteration k	3,4
$\tilde{R}_j(k)$	Adjusted supply of link j at iteration k	3
$U_j(k)$	Set of input links contributing to link j whose flows are undetermined as of iteration k	3
$\tilde{p}_{ij}(k)$	Oriented priority from link i to j at iteration k	3,4
$a_j(k)$	Restriction term of link j at iteration k	3
$a_{j^*}(k)$	Smallest (most restrictive) restriction term at iteration k	3
$\alpha_j(k)$	Reduction factor of link j at iteration k	3
$\tilde{U}(k)$	Set of input links whose demand can be fully met by downstream links at iteration k	3
$\mathcal{B}$	Set of commodity movement triples (i, j, c) whose split ratios are known	4
$\overline{\mathcal{B}}$	Set of commodity movement triples (i, j, c) whose split ratios are unknown	4
$\overline{\beta}_i^c$	Unassigned portion of the split ratios of commodity c from link i	4
$\overline{S}_i^c$	Unassigned demand of commodity c from link i	4
$\tilde{V}(k)$	Set of output links to which unassigned demand may still be assigned as of iteration k	4
$\mu^+(k)$	Largest oriented demand-supply ratio at iteration k	4
$\mu^-(k)$	Smallest oriented demand-supply ratio at iteration k	4

References

M. Bliemer. Dynamic queueing and spillback in an analytical multiclass dynamic network loading model. *Transportation Research Record*, 2029:14–21, 2007.

M. Bliemer and P. Bovy. Impact of Route Choice Set on Route Choice Probabilities. *Transportation Research Record: Journal of the Transportation Research Board*, 2076:10–19, Dec. 2008. ISSN 0361-1981. doi: 10.3141/2076-02. URL http://trrjournalonline.trb.org/doi/10.3141/2076-02.

California Department of Transportation (Caltrans). Office of System, Freight, & Rail Planning, 2015. URL http://www.dot.ca.gov/hq/tpp/corridor-mobility/.

R. Corthout, G. Flötteröd, F. Viti, and C. M. Tampére. 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*, 28(4):269–287, 1994.

C. Daganzo. The cell transmission model, Part II: Network traffic. *Transportation Research, Part B*, 29(2): 79–93, 1995.

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*, 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.

M. Hadi, S. Shabanian, 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.

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.

P. McCullagh and J. Nedler. *Generalized linear models*. Chapman & Hall/CRC, 1989.

D. McFadden. Conditional Logit Analysis of Qualitative Choice Behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, New York, 1973.

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*, In press:–, 2015. ISSN 0968-090X. doi: http://dx.doi.org/10.1016/j.trc.2015.07.002.

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.

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, 2015. Submitted to IEEE Transactions on Intelligent Transportation Systems.