

Discrete-time system optimal dynamic traffic assignment (SO-DTA) with partial control for horizontal queuing networks

Samitha Samaranayake, Jack Reilly, Walid Krichene, Maria Laura Delle Monache, Paola Goatin, Alexandre Bayen

► To cite this version:

Samitha Samaranayake, Jack Reilly, Walid Krichene, Maria Laura Delle Monache, Paola Goatin, et al.. Discrete-time system optimal dynamic traffic assignment (SO-DTA) with partial control for horizontal queuing networks. Transportation Science, INFORMS, In press. <hal-01095707>

HAL Id: hal-01095707

<https://hal.inria.fr/hal-01095707>

Submitted on 16 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Discrete-time system optimal dynamic traffic assignment (SO-DTA) with partial control for horizontal queuing networks

Samitha Samaranayake*, Jack Reilly, Walid Krichene,
Maria Laura Delle Monache, Paola Goatin and Alexandre Bayen

Abstract

We consider the *System Optimal Dynamic Traffic Assignment problem with Partial Control* (SO-DTA-PC) for general networks with horizontal queuing. The goal of which is to optimally control any subset of the networks agents to minimize the total congestion of all agents in the network. We adopt a flow dynamics model that is a Godunov discretization of the Lighthill-Williams-Richards (LWR) partial differential equation with a triangular flux function and a corresponding multi-commodity junction solver. Full Lagrangian paths are assumed to be known for the controllable agents, while we only assume knowledge of the aggregate turn ratios for the non-controllable (selfish) agents. We solve the resulting finite horizon non-linear optimal control problem using the discrete adjoint method.

1 Introduction

Dynamic traffic assignment (DTA) is the process of allocating time-varying *origin-destination* (OD) based traffic demand to a set of paths on a road network. This problem has been studied extensively over the last 35 years, since the seminal works of Merchant and Nemhauser (1978a,b). There are two types of traffic assignment that are generally considered, the *user equilibrium* or Wardrop equilibrium allocation (UE-DTA), in which users minimize individual travel-time in a selfish manner, and the *system optimal* allocation (SO-DTA) where a central authority picks the route for each user and seeks to minimize the aggregate total travel-time over all users. These principles were first presented in the work of Wardrop (1952) in the context of static traffic assignment and expanded on by Beckman et al. (1956). See Peeta and Ziliaskopoulos (2001) for a broad overview of dynamic traffic assignment. *User equilibrium* (UE) traffic assignment can lead to inefficient network utilization, highlighted by Braess' Paradox (Braess, 1968), where adding capacity to the network can actually result in longer travel times for all users. It has been shown that this paradox can occur in real road networks (Kelly, 1991) and that it is hard to design networks that are immune to it (Roughgarden, 2006). In fact, it can be shown that the *price of anarchy* (PoA) (Koutsoupias and Papadimitriou, 1999), the worst-case ratio of the system delay caused by the selfish behavior over the system optimal solution, may be arbitrarily large even in simple networks (Roughgarden and Tardos, 2004; Swamy, 2007). *System optimal* (SO) traffic assignment on the other hand leads to optimal utilization of the network resources, but is hard to achieve in practice since the overriding objective for individual drivers in a road network is to minimize their own travel-time. It is well known that setting a toll on each road segment corresponding to the marginal delay of the demand moves the user equilibrium towards a SO allocation (Roughgarden, 2002; Vickrey, 1969). However, imposing time-varying tolls on each road segment is impractical and tolling in general is difficult to implement in many settings due to both infrastructure and political considerations.

An alternative approach is to attempt to control a fraction of the drivers (via direct control or some incentive scheme) and assign routes via a central authority that tries to minimize system-wide total travel time. This has been studied in the context of Stackelberg routing games (Korilis et al., 1997; Roughgarden, 2001; Swamy, 2007), where the goal of the central controller is to assign routes to a fraction of the demand

*Corresponding author, e-mail: samitha@alum.mit.edu

using a strategy that minimizes the system wide cost, while anticipating selfish behavior from the demand that is not being controlled. Most of this work has been in the area of communication networks and assumes a non-decreasing latency function and vertical queues. However, these assumptions are generally not satisfied in road traffic networks, with horizontal queuing, because of congestion propagation and more complex latency functions, due to the physics of flows and driver behavior (Daganzo, 1994, 1995; Lighthill and Whitham, 1955; Richards, 1956). Therefore, the literature on partial control in traffic assignment is sparse and usually makes simplifying assumptions. For example, Aswani and Tomlin (2011) use vertical queues and non-decreasing latency functions, while Krichene et al. (2013) consider simple parallel networks. Ziliaskopoulos (2000) formulated the single destination SO-DTA problem (with full control) as a *Linear Program* (LP) under a LP relaxation that approximates the non-linear dynamics of the system. However, the SO-DTA problem with partial control can not be formulated as a convex problem, even in the case of a single destination, without violating the *first-in-first-out* (FIFO) condition (Carey, 1992), due to the multiple commodities (selfish and cooperative demand). Furthermore, solving the SO-DTA problem with an LP relaxation of the dynamics can lead to the holding of vehicles on links when the model allows for a larger flow. It has been argued that this holding can be achieved in practice via *variable speed limit* (VSL) signs (Ziliaskopoulos, 2000) and makes sense when the goal of the problem is to also solve for the optimal VSL values (Gomes and Horowitz, 2006), but is impractical to implement in most cases. Thus, there is a need for a more general solution that does not require VSL.

A further complication of DTA in practical settings is the unavailability of *origin-destination* (OD) data for the entire demand. Most DTA solutions assume that this data is available, although it can be challenging to obtain in practice. Therefore, we formulate the partial control problem in a manner that requires full OD information only for the demand that can be controlled by the central authority, and junction split ratios for the remaining demand (which are much easier to obtain via inductive loop detectors for example).

We formulate the *system optimal dynamic traffic assignment problem with partial control* (SO-DTA-PC), using a traffic dynamics model similar to the *Cell Transmission Model* (CTM) (Daganzo, 1994, 1995), which is a Godunov discretization of the *Lighthill-Williams-Richards* (LWR) *partial differential equation* (PDE) (Lighthill and Whitham, 1955; Richards, 1956) with a triangular flux function. The CTM is a horizontal queuing model and uses a latency function that gives a constant delay when the traffic density is below a certain threshold and progressively increases as the density increases beyond this threshold, and is well accepted in the transportation community as a good first order approximation of road traffic dynamics.

We propose solving the SO-DTA-PC problem with the non-convex traffic dynamics from (Delle Monache et al., 2013) and limited OD data with complete split ratios as a non-linear optimal control problem. This formulation generalizes to multiple sources and multiple destinations. The next challenge is in finding efficient descent methods for this non-convex optimal control problem. There is a vast literature on optimization techniques for non-convex control problems (see Bertsekas (1999) and the references therein) that can be utilized to solve this problem. While gradient based methods do not provide any guarantees of converging to the optimal solution in non-convex optimization problems, they can still be used to find local minima. One of the main computational challenges in this approach is the efficient computation of the gradient, since this computation must be repeated a large number of times. We show that the structure of our dynamical system allows for very efficient computation of the gradient via the discrete adjoint method (Bayen et al., 2006; Giles and Pierce, 1997, 2000; Jameson and Martinelli, 2000; Raffard, 2008). If the state vector is n dimensional and the control vector is m dimensional, direct computation of the gradient takes $O(n^2m)$ time. The adjoint method generally reduces the complexity to $O(n^2 + nm)$, but the structure of our system allows for further reduction of the complexity to $O(n + m)$ by exploiting the sparse nature of the forward system.

It should be noted that this work currently does not model the response of the selfish users (demand). It is clear that a change in the network state will result in a response from the selfish users as in a Stackelberg game. Finding the optimal control for a Stackelberg game is NP-Hard in the size of the network for the class of increasing latency functions even in the static case (Roughgarden, 2001) and it is common to use approximate strategies (Roughgarden, 2001; Swamy, 2007). We wish to extend this work in the future to model the selfish response.

The contributions of this article are as follows: 1) Formulating of the system optimal dynamic traffic

assignment with partial control problem as a multi-commodity finite horizon optimal control problem, 2) Defining the appropriate multi-commodity junction model for the network dynamics with reasonable data requirements, 3) Solving the gradient of the system with $O(n + m)$ time complexity for an n dimensional state space and m dimensional control vector using the discrete adjoint method, and 4) Experimental results for illustrating the benefits and applications of the technique.

The rest of the article is organized as follows. In Section 2, we present the traffic dynamics model with its assumptions, junction solver and boundary conditions. Following that, we introduce the notion of controllable and non-controllable flows in the network, in Section 3. In Section 4, we define the forward model of the system with explicit solutions to the junction problems. Section 5 formulates the optimization problem, derives the adjoint system and describes the gradient decent method. This is followed with numerical results in Section 6. We conclude with some final remarks and future research directions in Section 7.

2 Traffic model

The aggregate traffic dynamics are modeled using a macroscopic traffic flow model based on the *Lighthill-Williams-Richards* (LWR) PDE (Lighthill and Whitham, 1955; Richards, 1956). We use a multicommodity variant with buffers of the PDE model developed in Garavello and Piccoli (2006). This model imposes strong boundary conditions at the entrances to the network, so that no vehicles are dropped due to congestion propagating outside the bounds of the network, an important consideration in the optimal control setting. We then use a Godunov discretization (Godunov, 1959) of the network PDE model as explained in Reilly et al. (2013) to obtain an equivalent discrete model.

2.1 Network definitions

The road network is divided into cells, indexed by $i \in \mathcal{A}$. We add a buffer cell at the entrances of the network, to be able to impose the boundary demands in a strong sense. Each junction, indexed by $z \in \mathcal{J}$, connects a set of incoming links $\mathcal{J}_z^{\text{in}}$ to a set of outgoing link $\mathcal{J}_z^{\text{out}}$. The total flow in the network is decomposed into a set of $|\mathcal{C}|$ commodities that correspond to different types of flow.

Definition 1 (Supply and demand). *The supply of a cell i at time step k , denoted $\sigma_i(k)$, is maximal flow that can enter the cell, while the demand $\delta_i(k)$ is the maximal flow that can leave the cell. By assumption, buffers have no supply and the sinks have no demand.*

Definition 2 (Density). *The density on a link i at time step k , denoted by $\rho_i(k)$, is the total number of vehicles on the link during that time step divided by the length of the link L_i . The vehicles in the link could be from any of the $|\mathcal{C}|$ commodities in the network.*

Definition 3 (Single commodity density). *The density induced by a single commodity c on a link i at time step k , denoted by $\rho_{i,c}(k)$, is the total number of vehicles of commodity c on the link during that time step divided by the length of the link L_i , and satisfies*

$$\rho_i(k) = \sum_{c \in \mathcal{C}} \rho_{i,c}(k) \quad (1)$$

Definition 4 (Initial conditions). *The initial conditions of the network are the densities of each commodity at each link at time step $k = 0$ and are denoted $\rho_{i,c}(0)$.*

Definition 5 (Inflow and outflow). *The inflow (resp. outflow) from a cell i at time step k , denoted $f_i^{\text{in}}(k)$ (resp. $f_i^{\text{out}}(k)$), is the total flow leaving (resp. entering) the cell at time step k . By assumption, buffers have no inflow and sinks have no outflow.*

Definition 6 (Single commodity inflow and outflow). *The inflow (resp. outflow) from a cell i at time step k corresponding to commodity c , denoted $f_{i,c}^{\text{in}}(k)$ (resp. $f_{i,c}^{\text{out}}(k)$), is the total flow of commodity c leaving*

(resp. entering) the cell at time step k .

$$f_i^{\text{in}}(k) = \sum_{c \in \mathcal{C}} f_{i,c}^{\text{in}}(k) \quad (2)$$

$$f_i^{\text{out}}(k) = \sum_{c \in \mathcal{C}} f_{i,c}^{\text{out}}(k) \quad (3)$$

Definition 7 (State evolution). *The state of the network at time step k is given by the density $\rho_{i,c}(k)$ of each commodity c at each cell i . The density evolution is governed by the following dynamics, which simply correspond to mass conservation.*

$$\rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} (f_{i,c}^{\text{in}}(k-1) - f_{i,c}^{\text{out}}(k-1)) \quad \forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 1, T_f \rrbracket, \forall c \in \mathcal{C} \quad (4)$$

$$\rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} \cdot f_{i,c}^{\text{in}}(k-1) \quad \forall i \in \mathcal{S}, \forall k \in \llbracket 1, T_f \rrbracket, \forall c \in \mathcal{C} \quad (5)$$

with initial condition

$$\rho_{i,c}(0) = \rho_{i,c}^0 \quad \forall i \in \mathcal{A} \setminus \mathcal{S}, \forall c \in \mathcal{C} \quad (6)$$

$$\rho_{i,c}(0) = 0 \quad \forall i \in \mathcal{S}, \forall c \in \mathcal{C} \quad (7)$$

Assumption 1. The flux function defining the relationship between density and flow is given by the triangular fundamental diagram shown in figure 1. This is a first order approximation of the empirical relationship between flow and density Dervisoglu et al. (2009).

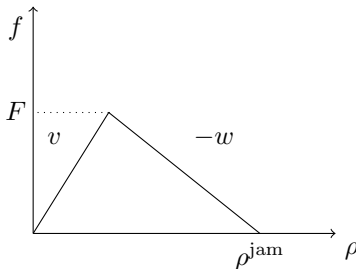


Figure 1: Triangular fundamental diagram.

Assumption 2. First-in first-out (FIFO) property. We assume that no vehicles entering a link at time step t will overtake vehicles that have already entered the link at some time step $t' < t$.

2.2 Discretization requirements

The network model is composed of links (each link representing a physical road) and nodes (each node representing a junction between some incoming and outgoing links). As with any numerical scheme, the accuracy of a macroscopic traffic model increases with the granularity of the network discretization. In addition, to ensure the convergence of the solution of the discretized model to the solution of the continuous LWR equation as the time step Δt and space discretization L goes to zero, the network must satisfy the *Courant-Friedrichs-Lewy* (CFL) conditions, which are standard requirements in numerical analysis (Godunov, 1959; Leveque, 2002). See Appendix B for a more detailed explanation of the CFL requirements.

2.3 Controllable and non-controllable flow

There are two types of flows that are transported in the network. Controllable flows that have origin destination requirements, but can be routed along any path in the network, and non-controllable flows

that have fixed paths. These flows are modeled by distributing the total flow of the network into multiple commodities, as explained below.

Assumption 3. Path decomposition of controllable flow. We assume that the controllable flows from each origin destination pair are restricted to a small pre-determined subset of paths in the network.

Definition 8 (Non-controllable commodity). *There is a single non-controllable commodity c_n that represents all non-controllable flow in the network.*

The paths of the flow corresponding to the non-controllable commodity are defined via the junction split ratios.

Definition 9 (Split ratio). *The split ratio of a commodity c at cell i and time step k among the outgoing cells $j \in \Gamma(i)$, denoted $\beta_{ij,c}(k)$, is the fraction of the commodity c flow out of cell i that is entering cell j at time step k , such that $\sum_{j \in \Gamma(i)} \beta_{ij,c}(k) = 1$.*

Definition 10 (Controllable commodities). *The controllable commodities $c_c \in \mathcal{CC}$ correspond to the controllable flow. There is a unique controllable commodity that corresponds to each path that the controllable flow can be routed along in the network. A controllable commodity is then equivalent to a tuple (origin, destination, path).*

Definition 11 (Conservation of flow). *The path of a controllable commodity is defined via the junction split ratios for this commodity.*

$$\beta_{ij,c}(k) = \begin{cases} 1 & \text{if the path of commodity } c \text{ includes cell } i \text{ and cell } j \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Definition 12 (Origin-destination demand). *The number of controllable vehicles that seek to travel from origin $o \in \mathcal{B}$ to destination $s \in \mathcal{S}$ at time step k is given by $D_{(o,s)}(k) \cdot \Delta t$. It is an exogenous input.*

Assumption 4. Data requirements. We assume that the origin destination information for all the controllable flow and the aggregate path information for all the non-controllable flow is known.

While at first glance this might seem like a lot of information to gather, it is in fact reasonable to assume in road traffic networks. We assume that the controllable flows are vehicles that are cooperating with the traffic coordination system that is trying to route vehicles efficiently and therefore will share their origin destination information. The aggregate paths of the non-controllable flows can be obtained from historical traffic patterns. The caveat is that empirical split ratios also include the contribution of the controllable flows and therefore must be pre-processed to remove this contribution.

Remark 1. *The junction split ratios for the non-controllable commodities may be time-dependent, while the junction split ratios for the controllable commodities are not, since a controllable commodity corresponds to a single path.*

Definition 13 (Equivalence between controllable commodity and path). *Any controllable commodity $c \in \mathcal{CC}$ is a tuple (o, s, p) where $o \in \mathcal{B}$, $s \in \mathcal{S}$ and p is a path (i.e. a sequence of cells). We define the function Ω as follows:*

$$\begin{aligned} \Omega: \mathcal{CC} &\rightarrow \mathcal{S} \times \mathcal{B} \\ c &\mapsto (o, s). \end{aligned} \quad (9)$$

$\Omega^{-1}(\{o, s\})$ is then the set of commodities corresponding to the flows from source o to destination s .

Definition 14 (Controllable flow control). *A control u is an allocation of the controllable demand over the set $\Omega^{-1}(\{o, s\})$ for each time step. Formally u is defined as:*

$$\begin{aligned} u: \mathcal{CC} \times \llbracket 0, T-1 \rrbracket &\rightarrow [0, 1] \\ (c, k) &\mapsto \gamma_c(k) \end{aligned} \quad (10)$$

$\gamma_c(k)$ is called the demand allocation for commodity c at time step k . The number of vehicles with origin o and destination s that are allocated to commodity c at time step k is $D_{(o,s)}(k) \cdot \gamma_c(k) \cdot \Delta t$.

Definition 15 (Physically feasible control). *A physically feasible control u verifies the mass conservation of the controllable demand allocation:*

$$\sum_{c \in \Omega^{-1}(\{o,s\})} \gamma_c(k) = 1 \quad \forall k \in \llbracket 0, T \rrbracket, \forall (o, s) \in \mathcal{B} \times \mathcal{S} \quad (11)$$

We will denote with \mathcal{U} be the set of all physically feasible controls.

3 Forward system

3.1 Junction model

The junction model defines the dynamics of the flow between neighboring cells. We require that it satisfies the following properties.

Requirement 1 (Multicommodity first-in first-out (FIFO) condition). *For any outgoing link i , the distribution of its flow across the different commodities must be proportional to the ratio of vehicles of each commodity on the link. If $\rho_i(k) \neq 0$ we must have:*

$$f_{i,c}^{\text{out}}(k) = f_i^{\text{out}}(k) \frac{\rho_{i,c}(k)}{\rho_i(k)} \quad (12)$$

Requirement 2 (Consistency with split ratios). *Let $f_{ij,c}(k)$ be the flow of commodity c from cell i to j at time step k . The outflow must be consistent with the split ratios.*

$$f_{ij,c}(k) = f_{i,c}^{\text{out}}(k) \cdot \beta_{ij,c}(k) \quad (13)$$

Requirement 3 (Maximum flow constraint). *The outflow cannot exceed the demand and the inflow cannot exceed the supply:*

$$0 \leq f_i^{\text{in}}(k) \leq \sigma_i(k) \quad \forall k \in \llbracket 0, T \rrbracket \quad (14)$$

$$0 \leq f_i^{\text{out}}(k) \leq \delta_i(k) \quad \forall k \in \llbracket 0, T \rrbracket \quad (15)$$

We wish to define a multi-commodity junction flow solver that assigns flows across the network in a manner that is consistent with the above requirements. The multi-commodity junction solver we consider is based on the *source destination model* (SDM) in Garavello and Piccoli (2006).

Definition 16 (Priority vector). *In the case of a junction where there is more than one incoming cell and the aggregate demand of these cells is greater than the aggregate supply of the outgoing cells, the available supply needs to be distributed among the competing demands according to some priority vector as follows. The priority vector P_j for cell j defines the allocation of its supply over the incoming cells $i \in \Gamma^{-1}(j)$. The priority for a given incoming cell i is given by P_{ij} such that $\sum_{i \in \Gamma^{-1}(j)} P_{ij} = 1$.*

Definition 17 (Aggregate split ratio). *The aggregate split ratio $\beta_{ij}(k)$ over all commodities for a given path through a junction is defined as follows:*

$$\begin{aligned} \beta_{ij}(k) &= \sum_{c \in \mathcal{C}} \frac{\rho_{i,c}(k)}{\rho_i(k)} \beta_{ij,c}(k) \\ &= \frac{1}{\rho_i(k)} \sum_{c \in \mathcal{C}} \rho_{i,c}(k) \beta_{ij,c}(k) \end{aligned} \quad (16)$$

The aggregate split ratio is only defined for positive aggregate densities, i.e. $\rho_i(k) > 0$.

3.1.1 Diverge solver ($1 \times m$)

We consider a diverging junction z with one incoming link i and m outgoing links. There are $|\mathcal{C}|$ commodities that flow through the junction each with their own time-varying split ratio $\beta_{ij,c}(k)$. If $\rho_i(k) = 0$, then $\delta_i(k) = 0$ and $f_i^{\text{out}}(k)$ is zero. We only consider the case of $\rho_i(k) \neq 0$.

Given the split ratios and densities of the cells at a junction, we wish to maximize the flow across the junction subject to the maximum flow constraints.

$$\begin{aligned} & \mathbf{max} \ f_i^{\text{out}}(k) && (17) \\ & \mathbf{subject\ to} \\ & \quad 0 \leq f_j^{\text{in}}(k) \leq \sigma_j(k) \quad \forall j \in \mathcal{J}_z^{\text{out}} \\ & \quad 0 \leq f_i^{\text{out}}(k) \leq \delta_i(k) \end{aligned}$$

We replace $f_i^{\text{in}}(k)$ using the following relation:

$$\begin{aligned} f_j^{\text{in}}(k) &= \sum_{c \in \mathcal{C}} \beta_{ij,c}(k) \cdot f_{j,c}^{\text{out}}(k) && [\text{mass conservation}] \\ &= f_i^{\text{out}}(k) \sum_{c \in \mathcal{C}} \frac{\rho_{i,c}(k)}{\rho_i(k)} \cdot \beta_{ij,c}(k) && [\text{by the FIFO constraint}] \\ &= f_i^{\text{out}}(k) \cdot \beta_{ij}(k) && [\text{by definition of the aggregate split ratios}] \end{aligned} \quad (18)$$

Which gives us a trivial maximization problem that implies the following equality.

$$f_i^{\text{out}}(k) = \min \left(\left\{ \frac{\sigma_j(k)}{\beta_{ij}(k)}, \forall j \in \mathcal{J}_z^{\text{out}} \mid \beta_{ij}(k) > 0 \right\}, \delta_i(k) \right) \quad (19)$$

The total outflow $f_i^{\text{out}}(k)$ for each incoming link i is then divided among the commodities according to the FIFO law: $f_{i,c}^{\text{out}}(k) = \frac{\rho_{i,c}(k)}{\rho_i(k)} f_i^{\text{out}}(k)$. The commodity flows are split among the outgoing links according to the split ratios constraints: $f_{j,c}^{\text{in}}(k) = \beta_{ij,c}(k) f_{i,c}^{\text{out}}(k)$.

Existence and uniqueness of solution. A non-zero solution exists if none of the constraints of the optimization/feasibility problem imposes a zero flow. In other words, as long as the demand is non-zero and none of the outgoing links with positive demand ($\beta_{ij}(k) > 0$) have non-zero supply, a non-zero solution exists. Since the solution to the maximum junction flow is given by equation (19) and the outflows are uniquely determined by the split ratios, the solution is unique.

3.1.2 Merge solver ($n \times 1$)

We consider a merging junction z with n incoming links and one outgoing link j . A priority vector P_j (s.t. $\sum P_{ij} = 1$) prescribes the priorities at which the outgoing link accepts flows from the n incoming links when the junction is supply constrained.

If the problem is demand constrained (i.e. $\sum_{i \in \mathcal{J}_z^{\text{in}}} \delta_i(k) < \sigma_j(k)$), then the solution is given by:

$$f_i^{\text{out}}(k) = \delta_i(k) \quad \forall i \in \mathcal{J}_z^{\text{in}} \quad (20)$$

Otherwise the problem is supply constrained and the solution to the junction problem is given by solving the following quadratic optimization problem that finds the flow-maximizing solution with the smallest violation

of the priority vector, where the violation is measured using the L_2 distance:

$$\min_{t, \{f_i^{\text{out}}(k)\}_{i \in \mathcal{J}_z^{\text{in}}}} \sum_{i \in \mathcal{J}_z^{\text{in}}} (f_i^{\text{out}}(k) - t \cdot p_{ij})^2 \quad (21)$$

subject to

$$\begin{aligned} \sum_{i \in \mathcal{J}_z^{\text{in}}} f_i^{\text{out}}(k) &= \sigma_j(k) \\ 0 \leq f_i^{\text{out}}(k) &\leq \delta_i(k) \quad \forall i \in \mathcal{J}_z^{\text{in}} \end{aligned}$$

The total outflow $f_i^{\text{out}}(k)$ for each incoming link i is then divided among the commodities according to the FIFO law: $f_{i,c}^{\text{out}}(k) = \frac{\rho_{i,c}(k)}{\rho_i(k)} f_i^{\text{out}}(k)$. See figure 2a for a graphical illustration of the solution to a 2×1 junction.

The priorities are satisfied exactly when the intersection of the maximum flow isoline and the priority constraint is feasible. When this point is outside the feasible set, the flow-maximizing feasible point that is closest to the priority constraint (in euclidean distance) is chosen¹.

Existence and uniqueness of the solution. In the demand-constrained case, existence and uniqueness are trivial. In the supplied constrained case, the solution is the feasible point that lies on the boundary of the feasible supply set (a segment) and minimizes the euclidean distance to the priority vector (a line), where the feasible set is given by the supply constraint (an n dimensional hyperplane: $\sum_{i \in \mathcal{J}_z^{\text{in}}} f_i^{\text{out}}(k) \leq \sigma_j(k)$) and demand constraints (an n -dimensional hyperrectangle: $f_i^{\text{out}}(k) \leq \delta_i(k), \forall i \in \mathcal{J}_z^{\text{in}}$). A solution exists when the feasible set is non-empty, which is the case if the supply/demand constraints are greater than zero. This proves the existence of a solution in all non-degenerate (zero supply or demand) cases. The boundary of the supply constraint hyperplane intersects each coordinate axis at $x_i(k) = \sigma_j(k)$ and can not be parallel to the priority constraint P , which is a line that goes through the origin. Therefore, since the solution must lie on a segment that is not parallel to the priority constraint line P , the solution that minimizes the distance to the P must be unique. This concludes the proof. See figure 2a for an illustration of the (2×1) case.

3.1.3 Merge-diverge solver ($2 \times m$)

We consider a junction with 2 incoming links and m outgoing links².

Assumption 5. The priority vectors P_j for each outgoing link j are identical. This implies that the inflow priorities are allocated with respect to the total flow that enters the junction and that the priority does not depend on which outgoing link the vehicles will enter.

The priority vector P_j prescribes the ratios at which the m outgoing links allocate their available supply to the 2 incoming links. It satisfies $P_i = P_{i1} = P_{i2}$ and $\sum_{i \in \mathcal{J}_z^{\text{in}}} P_i = 1$.

Let $\mathcal{J}_z^{\text{in}}$ and $\mathcal{J}_z^{\text{out}}$ be the sets of incoming and outgoing links at the junction. If the problem is demand-constrained (i.e. $\sum_{i \in \mathcal{J}_z^{\text{in}}} \beta_{ij}(k) \delta_i(k) \leq \sigma_j(k), \forall j \in \mathcal{J}_z^{\text{out}}$), then the solution is given by:

$$f_i^{\text{out}}(k) = \delta_i(k) \quad \forall i \in \mathcal{J}_z^{\text{in}} \quad (22)$$

¹ The solution violates the priority rule only in the case where the demand for one or more of the incoming links is less than what its flow-maximizing allocation is, based on the priority vector. In other words, the priority rule is only violated when an incoming link doesn't have enough flow to satisfy its priority-based allocation. It is reasonable in the physical sense to maximize flow and only violate the priority when it is a lack of demand that causes the violation. The model is not denying any vehicles with priority the ability to pass through the junction. This is an important property to note, because it avoids having to solve a multi-objective optimization problem to come up with a physically meaningful set of flows through the junction.

² We limit our analysis to merge-diverge junctions of no more than two incoming links because our model does not prescribe a unique solution when the number of incoming links is greater than two. Thus, our model can only be used with $1 \times m, n \times 1$ and $2 \times m$ junctions.

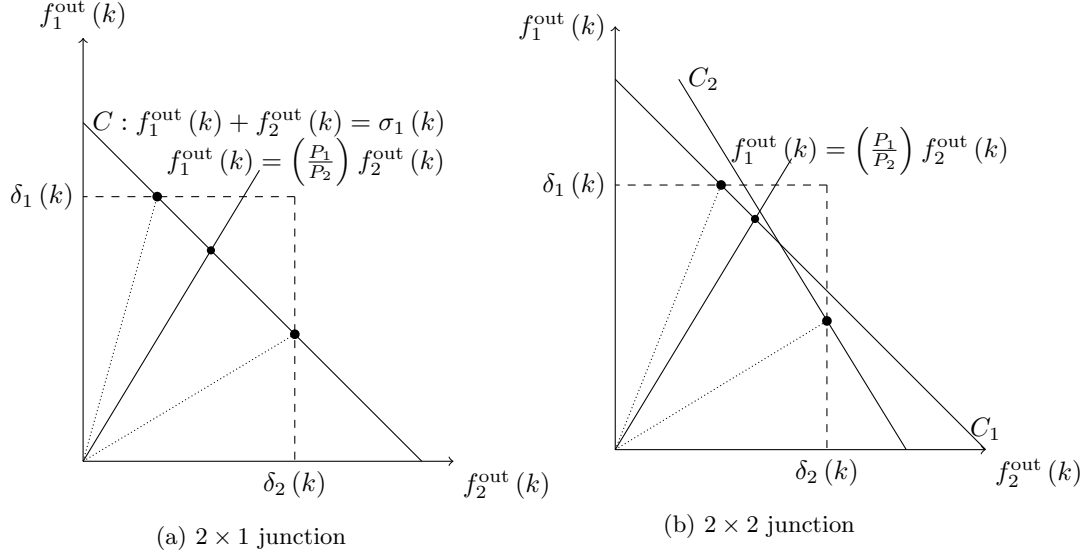


Figure 2: An illustration of the solutions to merging junctions. The dashed lines denote the demand constraints imposed by the density on the incoming links. The solid lines (C) denote the supply constraints imposed by the density in the outgoing links. The solid lines going through the origin denote the merge priority vector. If the priority vector intersects any supply constraint inside the feasible demand set, the solution will be the feasible intersection point. If the intersection is outside the feasible demand set, then the solution will be the nearest feasible point where the supply and the demand constraints intersect (marked with a dot).

Otherwise, the flows through the junction are given by the following optimization problem.

$$\begin{aligned}
 & \min_{t, \{f_i^{\text{out}}(k)\}} \sum_{i \in \mathcal{J}_z^{\text{in}}} (f_i^{\text{out}}(k) - t \cdot P_i)^2 & (23) \\
 & \text{subject to} \\
 & \sum_{i \in \mathcal{J}_z^{\text{in}}} \beta_{ij}(k) f_i^{\text{out}}(k) \leq \sigma_j(k) \quad \forall j \in \mathcal{J}_z^{\text{out}} \\
 & \max_j \left(\sum_{i \in \mathcal{J}_z^{\text{in}}} \beta_{ij}(k) f_i^{\text{out}}(k) - \sigma_j(k) \right) = 0 \quad \forall j \in \mathcal{J}_z^{\text{out}} \\
 & f_i^{\text{out}}(k) \leq \delta_i(k) \quad \forall i \in \mathcal{J}_z^{\text{in}}
 \end{aligned}$$

The total outflow $f_i^{\text{out}}(k)$ for each incoming link i is then divided among the commodities according to the FIFO law: $f_{i,c}^{\text{out}}(k) = \frac{\rho_{i,c}(k)}{\rho_i(k)} f_i^{\text{out}}(k)$. The commodity flows are split among the outgoing links according to the split ratio constraints: $f_{j,c}^{\text{in}}(k) = \sum_{i:(i,j) \in A} \beta_{ij,c}(k) f_{i,c}^{\text{out}}(k)$. See figure 2b for a graphical illustration of the solution to a 2×2 junction.

Existence and uniqueness of solution. In the demand constrained case, existence and uniqueness are trivial. In the supplied constrained case, the solution is the feasible point (with respect to the supply and demand constraints) that lies on the boundary of the feasible supply set (a union of segments) and minimizes the euclidean distance to the priority vector (a line). A solution exists when the feasible set is non-empty, which is the case if the supply/demand constraints are greater than zero. This proves the existence of a solution in all non-degenerate (zero supply or demand) cases. The feasible supply set is the intersection of

m two dimensional hyperplanes, which is a convex set. Therefore, the boundary of the feasible supply set (a union of segments) is also convex. Furthermore, the boundary of the feasible supply set intersects each coordinate axis at $x_i(k) = \min_j \frac{\sigma_j(k)}{\beta_{ij}(k)}$ and therefore can not be parallel to the priority constraint P , which is a line that goes through the origin. Therefore, since the solution must lie on a convex union of segments and none of these segments is parallel to the priority constraint line P , the solution that minimizes the distance to P must be unique. This concludes the proof.

3.2 Boundary conditions

The boundary conditions at each source link of the network dictate the flows that enter the network. Each boundary condition is given as a flow rate at the boundary.

Definition 18 (Boundary demand). *The number of vehicles of commodity c leaving cell $i \in \mathcal{B}$ at time step k is the boundary demand of commodity $d_{i,c}(k)$. Let c_n be the commodity corresponding to non-controllable flow. The non-zero terms are defined as:*

$$\begin{aligned} d_{i,c}(k) &= \Delta t \cdot D_{i,c}(k) & \forall i \in \mathcal{B}, c = c_n & \quad \text{(Non-controllable demand)} \\ d_{i,c}(k) &= \Delta t \cdot D_{\Omega(c)}(k) \cdot \gamma_c(k) & \forall i \in \mathcal{B}, \forall c \in \mathcal{CC} & \quad \text{(Controllable demand)} \end{aligned}$$

Since the inflow to the network is limited by the maximum flow capacity and density of the immediate downstream link, all of the demand at a given time step might not make it into the network. A source buffer is used to accumulate the flow that cannot enter the network to guarantee the conservation of boundary flows. In the single commodity case, this model is sufficient. However, in the multi commodity case, we also need to make sure that the flow through the boundary respects the multicommodity FIFO condition given in requirement 1.

As stated in Definition 18, $d_{i,c}(k)$ is the boundary demand per commodity on cell i at time step k . The FIFO condition dictates that the vehicles entering the boundary buffer at time k should enter link i at the ratio $\frac{d_{i,c}(k)}{\sum_{c \in \mathcal{C}} d_{i,c}(k)}$ for each commodity c .

The simplest solution is to have a single buffer l at the boundary, as in the single commodity case, and keep track of how many vehicles of each commodity are at the buffer. The flow into the boundary cell will be as follows:

$$f_{i,c}^{\text{in}}(k) = \frac{l_{i,c}(k)}{l_i(k)} f_i^{\text{in}}(k)$$

This equality satisfies the FIFO condition assuming that the vehicles in the buffer are uniformly distributed. However, in reality the buffer can accumulate vehicles arriving at the boundary at different time steps with different commodity ratios $\frac{d_{i,c}(k)}{d_i(k)}$. Thus, this model can violate the FIFO property across multiple time steps if some vehicles cannot leave the buffer in one time step. See Appendix C for a more detailed discussion on the FIFO condition and a multi buffer model that satisfies the FIFO condition.

As the length of a buffer can be seen as the density of a cell of length 1, we use the same notation $\rho_{i,c}(k)$ for a buffer i . The speed of this cell is then $v_i = \frac{1}{\Delta t}$ because of requirements 4 and 6.

3.3 System dynamics

For a given control u , we can determine the evolution of the network using the following equations that prescribe the system dynamics. Let $x(u)$ give the state of the network under these dynamics subject to the control u .

The system of equations governing the evolution of the network (implicit definition of x) are written formally in the form $H(x, u) = 0$, thus x is an implicit function of u . The discretized system dynamics can be described using six types of constraints, given by $H^h = 0, h \in \{1, \dots, 6\}$, listed below. These six constraints have different individual instantiations depending on the specific setting such as link type or junction type. The explicit formulation is given below.

Mass conservation

$$H_{k,i,c}^1 : \quad \rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} \left(f_{i,c}^{\text{in}}(k-1) - f_{i,c}^{\text{out}}(k-1) \right) \quad \forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 1, T_f \rrbracket, \forall c \in \mathcal{C} \quad (\text{H1a})$$

$$H_{k,i,c}^1 : \quad \rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} \cdot f_{i,c}^{\text{in}}(k-1) \quad \forall i \in \mathcal{S}, \forall k \in \llbracket 1, T_f \rrbracket, \forall c \in \mathcal{C} \quad (\text{H1b})$$

with initial conditions

$$H_{0,i,c}^1 : \quad \rho_{i,c}(0) = \rho_{i,c}^0 \quad \forall i \in \mathcal{A} \setminus \mathcal{S}, \forall c \in \mathcal{C} \quad (\text{I1a})$$

$$H_{0,i,c}^1 : \quad \rho_{i,c}(0) = 0 \quad \forall i \in \mathcal{S}, \forall c \in \mathcal{C} \quad (\text{I1b})$$

Boundary conditions³

$$H_{k,i,c}^1 : \quad \rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} (D_{\Omega(c)}(k) \cdot \gamma_c(k) - f_{i,c}^{\text{out}}(k-1)) \quad \forall i \in \mathcal{B}, \forall k \in \llbracket 1, T_f \rrbracket, \forall c \in \mathcal{C} \quad (\text{H1c})$$

$$H_{k,i,c}^1 : \quad \rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} (D_{i,c}(k) - f_{i,c}^{\text{out}}(k-1)) \quad \forall i \in \mathcal{B}, \forall k \in \llbracket 1, T_f \rrbracket, c = c_n \quad (\text{H1d})$$

with initial conditions

$$H_{0,i,c}^1 : \quad \rho_{i,c}(0) = \rho_{i,c}^0 + \frac{\Delta t}{L_i} \cdot D_{\Omega(c)}(0) \cdot \gamma_c(k) \quad \forall i \in \mathcal{B}, \forall c \in \mathcal{C} \quad (\text{I2a})$$

$$H_{0,i,c}^1 : \quad \rho_{i,c}(0) = \rho_{i,c}^0 + \frac{\Delta t}{L_i} \cdot D_{i,c}(0) \quad \forall i \in \mathcal{B}, c = c_n \quad (\text{I2b})$$

Flow propagation

Recall that $\rho_i(k) = \sum_{c=1}^C \rho_{i,c}(k)$ is the total density of cell i at time step k .

$$H_{k,i}^2 : \quad \delta_i(k) = \min(F_i, v_i \rho_i(k)) \quad \forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket \quad (\text{H2a})$$

$$H_{k,i}^2 : \quad \delta_i(k) = \min\left(F_i, \frac{\rho_i(k) L_i}{\Delta t}\right) \quad \forall i \in \mathcal{B}, \forall k \in \llbracket 0, T_f \rrbracket \quad (\text{H2b})$$

$$H_{k,i}^3 : \quad \sigma_i(k) = \min\left(F_i, w_i \left(\rho_i^{\text{jam}} - \rho_i(k)\right)\right) \quad \forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket \quad (\text{H3a})$$

$$H_{k,i}^3 : \quad \sigma_i(k) = F_i \quad \forall i \in \mathcal{S}, \forall k \in \llbracket 0, T_f \rrbracket \quad (\text{H3b})$$

Junction solution

The derivation of the explicit solutions to the 1×2 , 2×1 and 2×2 junctions listed below are given in appendix D. The general solutions for the junction model are given in section 3.1.

To simplify the notation, we use the following shorthand:

- We drop the time index k
- We abbreviate $i_1 = 1$ and $i_2 = 2$
- We use the following notation $\underline{i}_1 = i_2$ and $\underline{i}_2 = i_1$

When $\rho_i = 0$, there are no vehicles in the incoming link i and the outflow of this link is zero for all commodities. The following equations only apply for $\rho_i \neq 0$.

Aggregate split ratios:

$$H_{k,i,j,z}^4 : \quad \beta_{ij} = \frac{1}{\rho_i} \sum_{c \in \mathcal{C}} \rho_{i,c} \beta_{ij,c} \quad \forall z \in \mathcal{J}, \forall i \in \mathcal{J}_z^{\text{in}}, \forall j \in \mathcal{J}_z^{\text{out}}, \forall k \in \llbracket 0, T_f \rrbracket \quad (\text{H4})$$

³For notational simplicity, the equations given here are for the single buffer model.

Flow out of incoming links by commodity:

$$H_{k,i,c}^5 : f_{i,c}^{\text{out}} = \frac{\rho_{i,c}}{\rho_i} \min \left(\left\{ \frac{\sigma_j}{\beta_{ij}}, \forall j \in \mathcal{J}_z^{\text{out}} \mid \beta_{ij} > 0 \right\}, \delta_i \right) \quad \forall z \in \mathcal{J}_{1 \times n}, \forall i \in \mathcal{J}_z^{\text{in}} \quad (\text{H5a})$$

$$H_{k,i,c}^5 : f_{i,c}^{\text{out}} = \frac{\rho_{i,c}}{\rho_i} \begin{cases} \delta_i & \text{if } P_i (\min(\delta_1 + \delta_2, \sigma_1) - \delta_i) > \delta_i P_i \\ \min(\delta_1 + \delta_2, \sigma_1) - \delta_i & \text{if } P_i (\min(\delta_1 + \delta_2, \sigma_1) - \delta_i) > \delta_i P_i \\ P_i \min(\delta_1 + \delta_2, \sigma_1) & \text{otherwise} \end{cases} \quad \forall z \in \mathcal{J}_{2 \times 1}, \forall i \in \mathcal{J}_z^{\text{in}} \quad (\text{H5b})$$

$$H_{k,i,c}^5 : f_1^{\text{out}} = \begin{cases} \delta_1 & \text{if } \frac{P_1}{P_2} > \frac{\delta_1}{\min(\delta_2, \frac{\sigma_1 - \beta_{11}\delta_1}{\beta_{21}}, \frac{\sigma_2 - \beta_{12}\delta_1}{\beta_{22}})} \\ \min\left(\delta_1, \frac{\sigma_1 - \beta_{21}\delta_2}{\beta_{11}}, \frac{\sigma_2 - \beta_{22}\delta_2}{\beta_{12}}\right) & \text{if } \frac{P_1}{P_2} < \frac{\delta_1}{\min(\delta_2, \frac{\sigma_1 - \beta_{21}\delta_2}{\beta_{11}}, \frac{\sigma_2 - \beta_{22}\delta_2}{\beta_{12}})} \\ \min\left(\frac{P_1\sigma_1}{P_1\beta_{11} + P_2\beta_{21}}, \frac{P_1\sigma_2}{P_1\beta_{12} + P_2\beta_{22}}\right) & \text{otherwise} \end{cases} \quad \forall i \in \mathcal{J}_{2 \times 2}^{\text{in}} \quad (\text{27a})$$

f_2^{out} is obtained by symmetry. (27b)

$$f_{i,c}^{\text{out}} = \frac{\rho_{i,c}}{\rho_i} f_i^{\text{out}} \quad \forall i \in \{1, 2\}, \forall c \in C \quad (\text{H5c})$$

Flow into outgoing links by commodity:

$$H_{k,i,c}^6 : f_{i,c}^{\text{in}} = \sum_{x \in \mathcal{J}_z^{\text{in}}} \beta_{xi,c} f_{x,c}^{\text{out}} \quad \forall z \in \mathcal{J}, \forall i \in \mathcal{J}_z^{\text{out}}, \forall c \in C \quad (\text{H6})$$

As shown in figure 3, the dynamics equations have a topological ordering that allows for the following efficient forward simulation algorithm⁴.

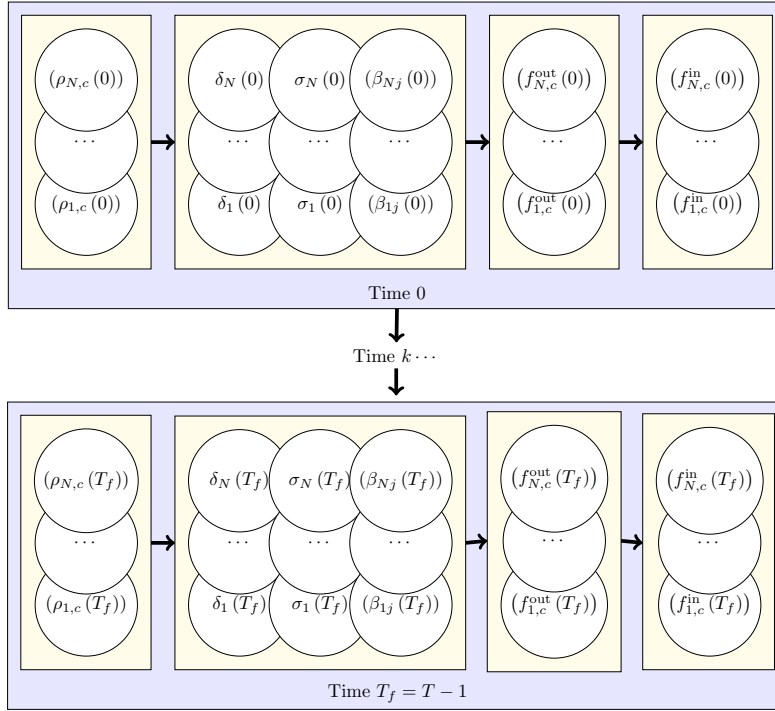


Figure 3: Dependency diagram of the variables in the system.

⁴For any given time step k , each of the three internal loops are trivially parallelizable problems.

4 Adjoint based optimization

4.1 Problem formulation

For a given control u , we can determine the evolution of the network using the equations for the system dynamics. Let $x(u)$ be the state of the network under these dynamics subject to the control u .

The total travel-time $J(x(u))$ is defined as:

$$J = \sum_{k=0}^{T-1} \sum_{i \in \mathcal{A} \setminus \mathcal{S}} \rho_i(k) \cdot L_i \quad (28)$$

The system optimal dynamic traffic assignment with partial compliance (SO-DTA-PC) is a physically acceptable (see Definition 14) division of the compliant agents among the different commodities that minimizes the total travel-time (including the travel-time of the non-compliant commodities). The solution is obtained by solving the following optimization problem.

$$\begin{aligned} & \min_{u \in \mathcal{U}} J(x(u)) \\ & \text{subject to} \\ & \quad \text{system dynamics} \\ & \quad \text{control constraints} \end{aligned}$$

where the system dynamics are given in Section 3.3 and the control constraints are the following.

$$\begin{aligned} \gamma_c(k) &\geq 0 & \forall c \in \mathcal{CC}, k \in \llbracket 0, T_f \rrbracket \\ \sum_{c \in \Omega^{-1}\{(o,s)\}} \gamma_c(k) &= 1 & \forall k \in \llbracket 0, T_f \rrbracket \end{aligned}$$

Note that this is a non-convex optimization problem that might contain multiple local minima. Therefore, gradient methods will not guarantee global optimality. However, descent algorithms can still be used to obtain locally optimal solutions and can be improved by using multiple starting points (Boese et al., 1994; Marti, 2003). Furthermore, non-convex optimization techniques such as subgradient and interior point methods (Giles and Pierce, 1997; Wachter and Biegler, 2005) require the gradient of the system. We use the discrete adjoint method, which will be explained in the next section, to efficiently solve for the gradient of the system. The control constraints can be satisfied either using a projected gradient descent or a barrier function. In our implementation, we use the projected gradient descent approach.

4.2 Overview of the adjoint method

We consider the following general optimization problem:

$$\begin{aligned} & \min_{u \in \mathcal{U}} J(x, u) \\ & \text{subject to } H(x, u) = 0 \end{aligned} \quad (29)$$

where $x \in \mathcal{X}$ denotes the state variables and $u \in \mathcal{U}$ denotes the control variables.

The adjoint method (Duffy, 2009) is a technique to compute the gradient $\nabla_u J(x, u) = \frac{dJ}{du}$ of the objective function without fully computing $\nabla_u x = \frac{dx}{du}$. The gradient is then used to perform a gradient descent. We suppose that for any control u , $\frac{\partial H}{\partial x}(x, u)$ is not singular.

Under equality constraints $H(x, u) = 0$, the Lagrangian

$$L(x, u, \lambda) = J(x, u) + \lambda^T H(x, u) \quad (30)$$

coincides with the objective function for any feasible point $(x(u), u)$. The problem is then equivalent to computing the gradient of the Lagrangian:

$$\begin{aligned}\nabla_u L(x, u, \lambda) &= \frac{\partial J}{\partial u} + \frac{\partial J}{\partial x} \frac{dx}{du} + \lambda^T \left(\frac{\partial H}{\partial u} + \frac{\partial H}{\partial x} \frac{dx}{du} \right) \\ &= \frac{\partial J}{\partial u} + \lambda^T \frac{\partial H}{\partial u} + \left(\frac{\partial J}{\partial x} + \lambda^T \frac{\partial H}{\partial x} \right) \frac{dx}{du}\end{aligned}\quad (31)$$

In particular, if λ satisfies the adjoint equation:

$$\frac{\partial J}{\partial x} + \lambda^T \frac{\partial H}{\partial x} = 0 \quad (32)$$

then the gradient is,

$$\nabla_u L(x, u) = \frac{\partial J}{\partial u} + \lambda^T \frac{\partial H}{\partial u} \quad (33)$$

The solution for λ exists and is unique if $\frac{\partial H}{\partial x}$ is not singular, which is the case in our forward system, as explained in the following section.

4.3 Applying the adjoint method

To be able to use the adjoint method to compute the gradient, the partial derivatives of the forward system with respect to the state variables $\frac{\partial H}{\partial x}$ must not be singular. We can rewrite our system of equations in the form $H(x, u) = 0$ and verify this condition trivially.

All the diagonal terms of $\frac{\partial H}{\partial x}$ are non zero (since equal to 1 or -1 depending on the way we rewrite H_v). As seen in the dependency chain shown in Figure 3, the non zero derivative terms of H_v depend only on variables that have a smaller index in x . This means that $\frac{\partial H}{\partial x}$ is lower triangular with no zero terms on the diagonal and is thus non singular. Therefore, we can apply the adjoint method to compute the gradient of this system.

Reduced state space. The forward system dynamics that were described in section 3.3 had a large number of state variables. However, the only required state variables of the system are the partial densities $\rho_{i,c}(k)$. All the others variables were introduced to make the forward system easier to understand. We will now drop most of these auxiliary variables to simplify the computation of the adjoint system. We only use $\rho_{i,c}(k)$, $f_{i,c}^{\text{out}}(k)$ and $f_{i,c}^{\text{in}}(k)$ to describe the system and replace the other variables by their expressions as a function of the three state variables that we retain.

We define:

$$x = \left(\begin{array}{c} \left((\rho_{i,c}(k))_{c \in \mathcal{C}} \right)_{i \in \mathcal{A}} \\ \left((f_{i,c}^{\text{out}}(k))_{c \in \mathcal{C}} \right)_{i \in \mathcal{A}} \\ \left((f_{i,c}^{\text{in}}(k))_{c \in \mathcal{C}} \right)_{i \in \mathcal{A}} \end{array} \right)_{k \in \llbracket 0, T \rrbracket} \quad H = \left(\begin{array}{c} \left(\left((H_{k,i,c}^1)_{c \in \mathcal{C}} \right)_{i \in \mathcal{A}} \right) \\ \left(\left((H_{k,i,c}^5)_{c \in \mathcal{C}} \right)_{i \in \mathcal{A}} \right) \\ \left(\left((H_{k,i,c}^6)_{c \in \mathcal{C}} \right)_{i \in \mathcal{A}} \right) \end{array} \right)_{k \in \llbracket 0, T \rrbracket}$$

Computational complexity. Let n be the dimension of the state vector $x \in R^n$, m be the dimension of the control vector is $u \in R^m$ and $N_c = |\mathcal{C}|$ be the total number of commodities. From the above definition of the state vector, we can see that $n = |\mathcal{A}| \cdot T \cdot N_c$. The dimension of \mathcal{H} is also n as defined above.

Direct computation of the gradient $\nabla_u J(x, u)$ takes $O(n^2 m)$ time.

$$\nabla_u J(x, u) = \frac{\partial J}{\partial x} \cdot \nabla_u x + \frac{\partial J}{\partial u} \quad (34)$$

Computing $\nabla_u J$ requires solving the system $H(x, u) = 0 \Rightarrow \frac{\partial H}{\partial x} \frac{dx}{du} + \frac{\partial H}{\partial u} = 0$, which is equivalent to solving m different $n \times n$ linear systems and takes $O(n^2 m)$ time. The final step of multiplying $\frac{\partial J}{\partial x} \frac{dx}{du}$ and adding $\frac{\partial J}{\partial u}$ takes $O(nm)$ time, but is dominated by the time to compute $\frac{dx}{du}$.

The discrete adjoint method reduces this complexity to $O(n^2 + nm)$ by solving for λ in the adjoint system. Computing the adjoint variables $\lambda^T \in R^n$ using equation (32) only takes $O(n^2)$ time since it only requires solving one $n \times n$ linear system. Multiplying $\lambda^T \frac{\partial H}{\partial u}$ and adding $\frac{\partial J}{\partial u}$ takes $O(nm)$ time, so the total computation time is $O(n^2 + nm)$.

The structure of our system allows for further reduction of the complexity to $O(n + m|\mathcal{C}|)$. As shown in section 4.3, $\frac{\partial H}{\partial x}$ is a lower triangular matrix and therefore we can compute the solution to equation (32) using backwards substitution. We will exploit the fact that the matrix $\frac{\partial H}{\partial x}$ is extremely sparse. The maximum row cardinality is four because the forward system does not contain any constraints with more than four variables. Therefore, equation (32) can be solved in $O(n)$ time. If the maximum in-degree of the network is d_{in} , the maximum column cardinality is $2 + |\mathcal{C}|(1 + d_{in})$, as will be clear in the next section from equation (38). Assuming that d_{in} is a small constant, the multiplication step in equation (33) takes $O(m|\mathcal{C}|)$ time. This leads to a total computation time of $O(n + m|\mathcal{C}|)$.

4.4 Adjoint equations

The adjoint equations are given by the system:

$$\frac{\partial J}{\partial x} + \lambda^T \frac{\partial H}{\partial x} = 0 \quad (35)$$

$$\Rightarrow \frac{\partial J}{\partial x} + \sum_{x' \in x} \lambda_{x'} \frac{\partial H_{x'}}{\partial x} = 0 \quad (36)$$

where x is the state vector and H'_x is the forward system equation corresponding to the variable $x' \in x$. To write the adjoint system equation corresponding to x' , we have to look at all the forward system equations where x' appears and consider all the non-null $\frac{\partial H_{x'}}{\partial x}$ terms. In particular we write the equations such that $\frac{\partial H_{x'}}{\partial x'} = -1$. Note that this can be done because the Godunov scheme provides an explicit expression for the forward system constraints.

Computing $\frac{\partial J}{\partial x}$

$$\frac{\partial J}{\partial \rho_{i,c}(k)} = \begin{cases} L_i & \forall c \in \mathcal{C}, \forall i \in \mathcal{A} \setminus \mathcal{S}, \forall k \in \llbracket 0, T \rrbracket \\ 0 & \text{otherwise} \end{cases} \quad (37)$$

Computing $\lambda^T \frac{\partial H}{\partial x}$

$$\begin{aligned} \frac{\partial H}{\partial \rho_{i,c}(k)} : \quad \sum_{x' \in x} \lambda_{x'} \frac{\partial H_{x'}}{\partial \rho_{i,c}(k)} &= \lambda_{\rho_{i,c}(k)} \frac{\partial H_{\rho_{i,c}(k)}}{\partial \rho_{i,c}(k)} + \lambda_{\rho_{i,c}(k+1)} \frac{\partial H_{\rho_{i,c}(k+1)}}{\partial \rho_{i,c}(k)} + \\ &\sum_{c' \in \mathcal{C}} \left(\lambda_{f_{i,c'}^{\text{out}}(k)} \frac{\partial H_{f_{i,c'}^{\text{out}}(k)}}{\partial \rho_{i,c}(k)} + \lambda_{f_{x,c'}^{\text{out}}(k)} \sum_{x:(x,i) \in \mathcal{A}} \frac{\partial H_{f_{x,c'}^{\text{out}}(k)}}{\partial \rho_{i,c}(k)} \right) \end{aligned} \quad (38)$$

$$\frac{\partial H}{\partial f_{i,c}^{\text{out}}(k)} : \quad \sum_{x' \in x} \lambda_{x'} \frac{\partial H_{x'}}{\partial f_{i,c}^{\text{out}}(k)} = \lambda_{\rho_{i,c}(k+1)} \frac{\partial H_{\rho_{i,c}(k+1)}}{\partial f_{i,c}^{\text{out}}(k)} + \lambda_{f_{i,c}^{\text{out}}(k)} \frac{\partial H_{f_{i,c}^{\text{out}}(k)}}{\partial f_{i,c}^{\text{out}}(k)} + \sum_{j:j \in (i,j)} \lambda_{f_{j,c}^{\text{in}}(k)} \frac{\partial H_{f_{j,c}^{\text{in}}(k)}}{\partial f_{i,c}^{\text{out}}(k)} \quad (39)$$

$$\frac{\partial H}{\partial f_{i,c}^{\text{in}}(k)} : \quad \sum_{x' \in x} \lambda_{x'} \frac{\partial H_{x'}}{\partial f_{i,c}^{\text{in}}(k)} = \lambda_{\rho_{i,c}(k+1)} \frac{\partial H_{\rho_{i,c}(k+1)}}{\partial f_{i,c}^{\text{in}}(k)} + \lambda_{f_{i,c}^{\text{in}}(k)} \frac{\partial H_{f_{i,c}^{\text{in}}(k)}}{\partial f_{i,c}^{\text{in}}(k)} \quad (40)$$

The next section shows how to compute all the individual partial derivatives that are required in the above equations. Once they are computed, we can simply plug them into the above equations and solve the system via backwards substitution since $\frac{\partial H}{\partial x}$ is lower triangular.

4.5 Partial derivatives

Computing the gradient of the system via the adjoint method requires computing the partial derivatives $\frac{\partial J}{\partial u}$, $\frac{\partial J}{\partial x}$, $\frac{\partial H}{\partial u}$ and $\frac{\partial H}{\partial x}$. The first three of these can be computed trivially.

- Partial derivatives of the cost function with respect to the control ($\frac{\partial J}{\partial u}$) from equation (28)

$$\frac{\partial J}{\partial \gamma_c(k)} = 0 \quad (41)$$

- Partial derivatives of the cost function with respect to the state variables ($\frac{\partial J}{\partial x}$) from equation (28)

$$\frac{\partial J}{\partial \rho_{i,c}(k)} = \begin{cases} L_i & \text{if } c \in \mathcal{C}, i \in \mathcal{A} \setminus \mathcal{S}, k \in \llbracket 0, T \rrbracket \\ 0 & \text{otherwise} \end{cases} \quad (42)$$

- Partial derivatives of the constraints with respect to the state variables ($\frac{\partial H}{\partial u}$) from equation (H1c)

$$\frac{\partial \rho_{i,c}(k)}{\partial \gamma_c(k)} = \begin{cases} \frac{\Delta t}{L_i} \cdot D_{\Omega(c)}(k) & \text{if } c \in \mathcal{CC}, i \in \mathcal{B}, k \in \llbracket 0, T \rrbracket \\ 0 & \text{otherwise} \end{cases} \quad (43)$$

Finally, we need to compute the partial derivatives of the constraints with respect to the state variables ($\frac{\partial H}{\partial x}$), which is much more algebraically involved and therefore described in appendix E.

5 Numerical Results

To illustrate the effectiveness of our framework for computing the system optimal dynamic flow allocation with partial control, we have implemented the algorithm and tested it on both synthetic and practical traffic rerouting scenarios using experimental field data. Our implementation uses the discrete adjoint method to compute the gradient and uses a log barrier function with a projection step to keep the solution in the physically feasible control set as explained in section 4.1. We use the Rprop (Riedmiller and Braun, 1992) algorithm as our gradient descent technique. All the experiments were run on a 1.8 GHz Intel Core i5 dual-core processor with 8GB of RAM. The performance cost C of each scenario is measured using the total travel time of all the vehicles passing through the network.

$$C = \sum_{k=0}^{T-1} \sum_{i \in \mathcal{A} \setminus \mathcal{S}} \rho_i(k) \cdot L_i \cdot \Delta t$$

We present numerical results for two network scenarios.

1. A network adapted from the synthetic network used in Ziliaskopoulos (2000). The network is illustrated in figure 4.
2. A subsection of Interstate 210 with a parallel arterial route, as depicted in figure 6.

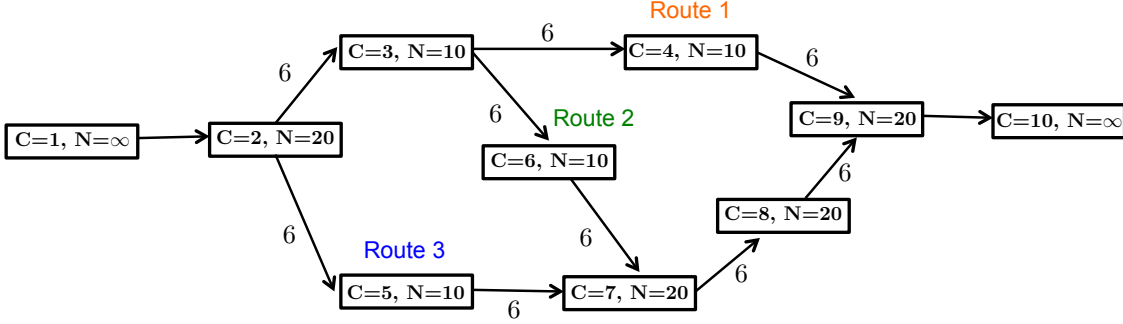


Figure 4: The synthetic network adapted from Ziliaskopoulos (2000). There are 10 cells marked $C = 1, \dots, 10$ and the jam density ρ^{jam} of each cell is denoted by the maximum number of vehicles (N), since the length of each cell is normalized to 1. The edge weights represent the max flow F between the neighboring cells.

5.1 Synthetic network

The synthetic network is a simple 10 cell network adapted from the example used in Ziliaskopoulos (2000)⁵. It contains three paths over which vehicles can be routed. The time discretization is set to one time unit and the length of each cell is also normalized to one unit. Therefore, the total capacity of each cell in terms of the number of vehicles N is equal to the jam density ρ^{jam} . Each cell in figure 4 is annotated with its cell capacity N , while the edge weights in the network prescribe the max flow F between the cells. The free flow speed v of each cell is also normalized to one and the congestion speed w is equal to the free flow speed. The demand at the origin at the first three time steps is respectively 8, 16 and 8 vehicles. The network is simulated for 10 time steps, which gives enough time for all the entering flow to exit the network.

First we use the discrete adjoint optimization framework to compute the system optimal flow allocation for the network assuming that all of the flow is compliant. Table 1(a) shows the optimal route allocation for the origin demands at each time step with non-zero demand. The total travel time cost (C) with the optimal flow allocation is 178 time units. The solution converges to within 0.5% of the optimal solution in three iterations.

Time step	1	2	3
Route 1	0.75	0.5	0.5
Route 2	0	0	0
Route 3	0.25	0.5	0.5

(a) normal operation

Time step	1	2	3
Route 1	0.25	0.417	.417
Route 2	0	0	0
Route 3	0.75	0.583	0.583

(b) incident local minimum

Time step	1	2	3
Route 1	0	0.25	0.25
Route 2	0.25	0	0
Route 3	0.75	0.75	0.75

(c) incident global minimum

Table 1: Optimal allocation of demand across routes

Capacity reduction due to incident. We now consider the case where the capacity between cell 3 and cell 4 is temporarily reduced due to some incident. The corresponding capacities for link (3, 4) are given in table 2.

Time step	1	2	3	4	5	6	7	8	9	10
$F_{(3,4)}$	6	6	0	0	3	3	6	6	6	6

Table 2: Capacity reduction due to incident

If the vehicles continue to be routed using the previous path allocation, the total travel cost (C) will now be 244 time units. The total cost increases by 37% because a large percentage of vehicles are routed along

⁵We have modified the network to increase the capacity of links 8 and 9 such that they can accommodate flow-from both route 2 and route 3, and changed some of the other parameters to satisfy the CFL conditions in section 2.2.

the path that is temporarily closed and then subjected to a reduced capacity. If we recompute the system optimal flow allocation, the total cost decreases to 211 time units and the corresponding flow allocation is given in table 1(b). This solution is actually a local minimum in the system due to the FIFO condition for vehicles departing cell 3. Whenever there is some non-zero flow for route 1 when the capacity is zero, the flow of vehicles that take route 2 is also restricted to zero. This causes a non-convexity that results in a discontinuity of the gradient at the point where the flow of vehicles on route 1 is zero. Gradient-based methods are not well suited to deal with such conditions because the information obtained from the gradient only provides local information. The global optimal solution occurs with the flow allocation given in table 1(c) and results in a total travel time cost of 207. As mentioned in Section 4.1, the effect of local minima can in general be mitigated using multi-start strategies (Boese et al., 1994; Marti, 2003) and the efficient gradient computation obtained via the adjoint method can be combined with non-convex optimization techniques such as interior point methods (Wachter and Biegler, 2005).

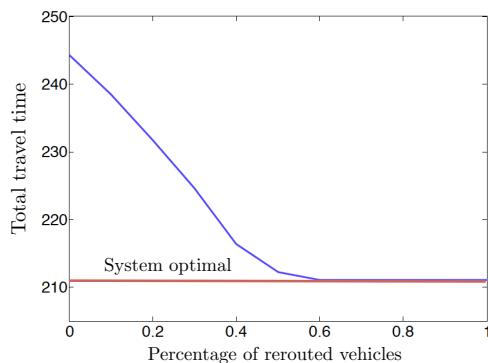


Figure 5: The change in total travel time vs percentage of vehicles that can be rerouted. All performance measures are with respect to the local minimum found by the optimizer.

Partial control. In many situations, it might not be possible to reroute all the vehicles in the system. Therefore, we also analyze the behavior of the system when only some fraction of the vehicles are rerouted. Figure 5 shows how the total travel time changes with the percentage of vehicles that can be rerouted. In this example, we see that the system optimal (local minimum) can be achieved by controlling only 60% of the vehicles.

5.2 Interstate 210 network

The experimental analysis was conducted on a 8 mile corridor of Interstate 210 in Arcadia, California with a parallel arterial route, as illustrated in figure 6. The network has 24 cells corresponding to satisfying the CFL condition for a time step of 30 seconds. The physical properties of the network such as the capacity were obtained using the Scenario Editor software developed as part of the Connected Corridors project, a collaboration between the University of California Berkeley and California Partners for Advanced Transportation Technology (PATH). Calibrated fundamental diagram parameters, split ratios, and boundary data were also obtained from other parallel research efforts at Connected Corridors. The data used for calibrating these parameters was obtained from the Freeway Performance Measurement System (PeMS) (Chen et al., 2001). We consider a prototypical one hour time horizon during the morning commute. The density profile of the freeway under the calibrated parameters and estimated boundary flows is shown in figure 7(a).

Capacity drop during morning commute. We analyze the behavior of the freeway corridor in the event of a capacity drop caused by some incident. We assume that the capacity drop occurs at the fifth freeway road segment 10 minutes into the simulation and that it lasts for 20 minutes, as illustrated in figure 7 (b).

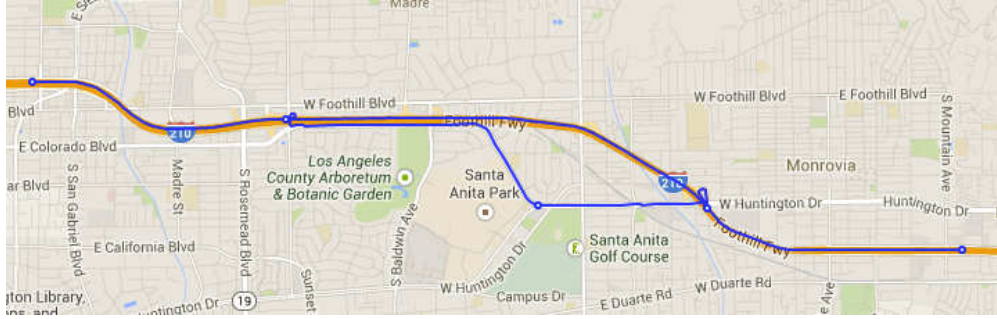


Figure 6: The Interstate 210 sub-network

The freeway capacity at segment five will be assumed to be reduced by half during this period, corresponding to a closure of two lanes (out of four) at the location of the incident. Figure 7 shows the density profile corresponding to; (a) normal operation with capacity drop, (b) a capacity drop due to a two lane closure during the incident with no traffic diversion, (c) the same capacity drop with traffic being diverted to the parallel arterial, and (d) the change in the density profile due to the traffic diversion. As the figure shows, rerouting the excess flow to the parallel arterial eliminates the bottleneck during the incident and improves the throughput of the freeway corridor. In this example, the parallel arterial is assumed to prioritize vehicles being routed from the freeway and the full arterial capacity is used for this purpose. However, in certain situations, municipalities may want to allocate some capacity of the parallel arterial for local traffic. In this case, the optimizer can be limited to only use a certain fraction of the capacity of the parallel arterial. Figure 8 shows the density evolution when the arterial capacity allocated for rerouting freeway traffic is limited to 40% and 50% in comparison to full arterial utilization. The arterial capacity allocation can be controlled via the traffic signal controls along the arterial.

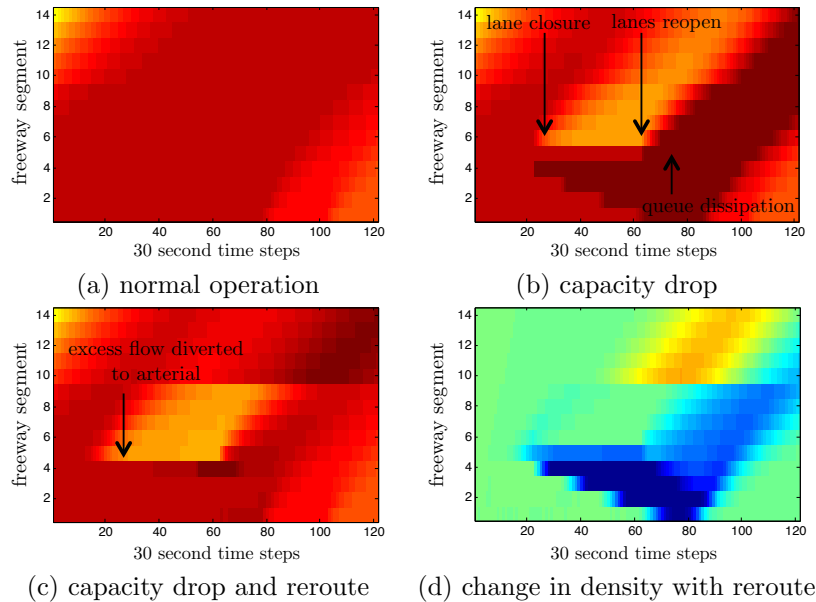


Figure 7: The density evolution along the 14 freeway road links with; (a) no incident, (b) a two lane capacity drop from minutes 10-30 at link 5, and (c) flow being rerouted to the parallel arterial due to the capacity drop, and (d) the density difference between the incident profiles with and without rerouting.

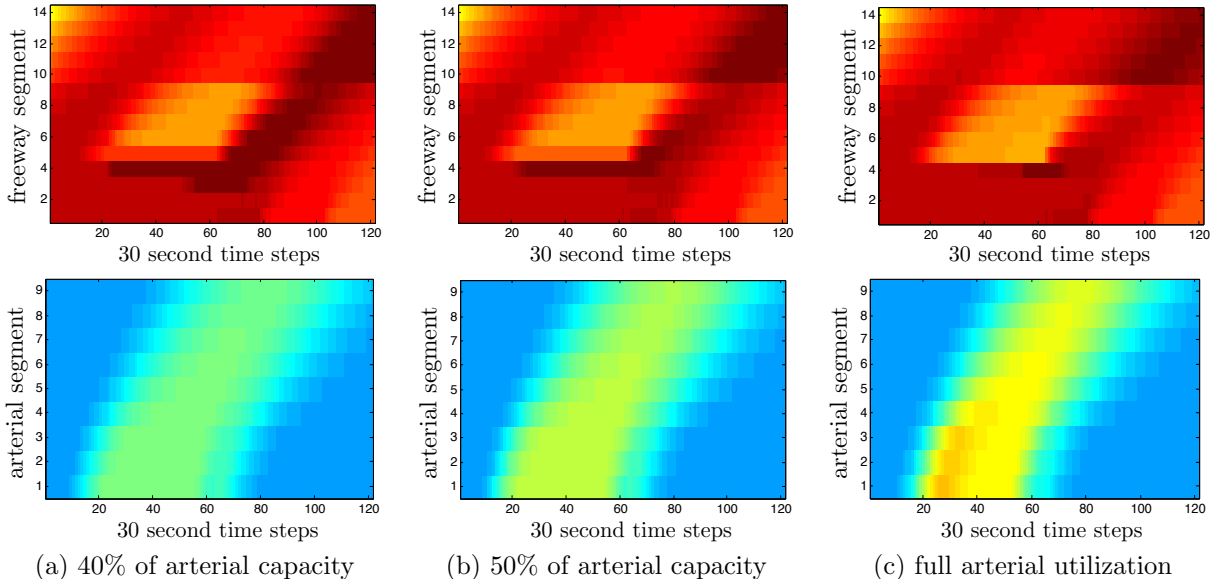


Figure 8: A comparison of the density evolution for different rerouting capacities on the parallel arterial route; (a) only 40% of the arterial capacity can be utilized for rerouting freeway vehicles, (b) only 50% of the arterial capacity can be utilized for rerouting freeway vehicles, and (c) The entire arterial capacity is utilized for rerouting freeway vehicles (i.e. the parallel arterial temporarily closed for other traffic).

Adjoint method vs finite differences. To demonstrate the increased efficiency of computing the solution via the discrete adjoint method, we also implemented the gradient computation using a simple finite differences method⁶ by perturbing each variable and measuring the response of the system. This approach, which approximates the gradient at a given point, has a runtime complexity of $O(nm)$ where n is the dimension of the state vector and m is the dimension of the control vector. Recall that the size of the state vector is $n = |\mathcal{A}| \cdot T \cdot |\mathcal{C}|$ and that the size of the control vector is $m = T \cdot |\mathcal{C}|$, where $|\mathcal{C}|$ is the total number of commodities (feasible paths) in the problem. Therefore, $O(nm) = O(|\mathcal{A}| \cdot T^2 \cdot |\mathcal{C}|^2)$ and the finite difference method has a computation time that is quadratic in the number of time steps. In comparison, the adjoint method has a time complexity of $O(n + m|\mathcal{C}|) = O(|\mathcal{A}| \cdot T \cdot |\mathcal{C}| + T \cdot |\mathcal{C}|^2)$, which is linear in the number of time steps. The complexity of both methods is quadratic in the total number of feasible paths, but this is assumed to be a small number in practical routing problems, since vehicles that travel between a fix origin-destination pair will only typically have a small number of reasonable paths. Figure 9 shows the time taken for one gradient computation as a function of the network number of time steps in the problem for the I-210 network. The simulations are run by changing the time discretization of the problem to control the total number of time steps. Reducing the time discretization also increases the number of cells due to the maximum cell length imposed by the CFL condition. The results show that the finite differences approach quickly becomes computationally intractable as the number of time steps in the problem increases and highlights the value of the discrete adjoint method for solving large problems in a tractable manner.

6 Conclusion

This article has presented a model and optimization framework for solving the *System Optimal Dynamic Traffic Assignment problem with Partial Control* (SO-DTA-PC) for general networks with horizontal queuing dynamics. The model only requires full origin-destination (OD) information for the fraction of the agents that are controllable, with aggregate split ratios being sufficient for the non-controllable (selfish) agents. We

⁶See (Morton and Mayers, 2005) for a detailed analysis of finite difference methods.

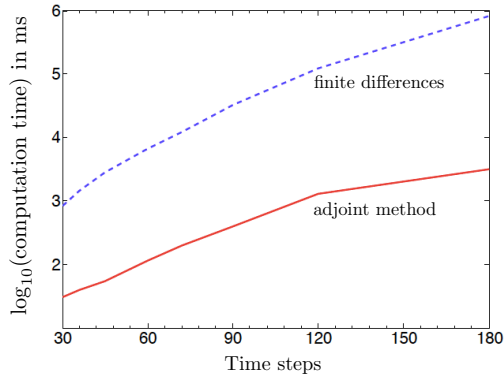


Figure 9: The base 10 logarithm (\log_{10}) of the total computation time for solving the I-210 network vs the number of time steps in the problem. The total time horizon is fixed, so a larger number of time steps implies a smaller time discretization. This also results in a larger number of cells (smaller in length) due to the CFL condition.

have utilized a Godunov discretization of the Lighthill-Williams-Richards (LWR) partial differential equation with a triangular flux function and a corresponding multi-commodity junction solver for the dynamics of the system, but the proposed framework can also be used with other dynamics models. Furthermore, we show that the sparsity pattern of the forward system allows us to compute the gradient of the system with linear computational complexity and memory using the discrete adjoint method. Finally, we apply this framework to find the optimal vehicles rerouting strategy in response to a capacity loss in the network, and show the congestions reductions that can be achieved. Numerical results are presented for a test network and Interstate 210 in Southern California. The system is implemented in the *Connected Corridors* system at UC Berkeley, a partnership between the California Department of Transportation (CalTrans) and California Partners for advanced transportation technology (PATH) for *Integrated Corridor Management* (ICM).

7 Acknowledgements

The authors have been supported by the California Department of Transportation under the Connected Corridors program, NSF CAREER grant CNS-0845076 under the project ‘Lagrangian Sensing in Large Scale Cyber-Physical Infrastructure Systems’, the European Research Council under the European Union’s Seventh Framework Program (FP/2007-2013) / ERC Grant Agreement 257661, the INRIA associated team ‘Optimal REroute Strategies for Traffic managEment’ and the France-Berkeley Fund under the project ‘Optimal Traffic Flow Management with GPS Enabled Smartphones’.

Appendices

A Notation

Constants

$\Delta t, T$	Time discretization and number of time steps ⁷
v_i	Free flow speed on cell i
w_i	Congestion wave speed on cell i
L_i	Length of cell i
ρ_i^{jam}	Jam density on cell i
F_i	Max flow capacity of cell i
P_{ij}	Merge priority factor from cell i to cell j

Sets

$\mathcal{J}_z^{\text{in}}$	Incoming links to junction z
$\mathcal{J}_z^{\text{out}}$	Outgoing links to junction z
\mathcal{A}	The cells (including buffers and sinks)
\mathcal{B}	The buffer cells
\mathcal{S}	The sink cells
OD	The set of origin-destination (OD) pairs
$\Gamma^{-1}(i)$	The predecessors of cell i
$\Gamma(i)$	The successors of cell i
\mathcal{C}	The commodities
\mathcal{CC}	The controllable commodities

Inputs

$\rho_{i,c}(0)$	Initial density of commodity c on cell i
$\beta_{ij,c}(k)$	Split ratio for commodity c on cell i to cell j , time step k
$D_{(o,s)}(k)$	Demand rate of controllable agents going from $o \in \mathcal{B}$ to $s \in \mathcal{S}$, time step k
$D_{i,c}(k)$	Demand rate of non-controllable agents of commodity $c \in \mathcal{NC}$ on cell i , time step k

Variables

$f_i^{\text{in}}(k)$	Total flow into cell i , time step k
$f_i^{\text{out}}(k)$	Total flow out of cell i , time step k
$f_{i,c}^{\text{in}}(k)$	Flow of commodity c into cell i , time step k
$f_{i,c}^{\text{out}}(k)$	Flow of commodity c out of cell i , time step k
$\rho_i(k)$	Density on cell i , time step k
$\rho_{i,c}(k)$	Density contribution of commodity c on cell i , time step k
$\sigma_i(k)$	Supply on cell i , time step k
$\delta_i(k)$	Demand on cell i , time step k
$d_i(k)$	Boundary demand on cell i , time step k
$\gamma_c(k)$	Demand allocation for commodity c , time step k

⁷Time from $t = 0$ and the final time step $T = T - 1$.

B CFL conditions

Requirement 4 (CFL condition 1). $\forall i \in \mathcal{A} \setminus \mathcal{S}, v_i \leq \frac{L_i}{\Delta t}$

For numerical stability, the vehicles in a given cell should only be able to travel forward at most one cell in a single time step. Requirement 4 ensures that this condition is satisfied by imposing an upper bound on the velocity.

Requirement 5 (CFL condition 2). $\forall i \in \mathcal{A} \setminus \mathcal{B}, w_i \leq \frac{L_i}{\Delta t}$

Each cell can not have a density greater than ρ_i^{jam} . Requirement 5 ensures that this condition is satisfied. While the requirement that $v_i \leq \frac{L_i}{\Delta t}$ comes from the positive density, this one comes from the fact that the density has to be smaller than ρ_i^{jam} . Indeed, in the case of a cell with no outflow (because of an extreme congestion of the next cell), the inflow can be limited by the supply. If there is enough demand we have at the next time the density which has to be smaller than ρ_i^{jam} :

$$\rho_i(k) + \frac{\Delta t}{L_i} w_i (\rho_i^{\text{jam}} - \rho_i(k)) < \rho_i^{\text{jam}}$$

$$\rho_i(k) \left(1 - w_i \frac{\Delta t}{L_i}\right) < \rho_i^{\text{jam}} \left(1 - w_i \frac{\Delta t}{L_i}\right)$$

And because $0 \leq \rho_i(k) \leq \rho_i^{\text{jam}}$ we also have $1 - w_i \frac{\Delta t}{L_i} \geq 0$ which is the requirement.

Requirement 6 (Finite time density discharge). $\forall i \in \mathcal{A}, v_i \geq \frac{L_i}{\Delta t}$

This condition guarantees that the density of a given cell discharges in a finite amount of time when there is no incoming flow. Requirement 6 ensures that this condition is satisfied. If $v_i < \frac{L_i}{\Delta t}$, we can have exponential decrease of the density in some cells when they should be emptied in only 1 steps. Taking the case of a cell without inflow, we have $\rho_i(k+1) = \rho_i(k) - \frac{\Delta t}{L_i} \rho_i(k) v_i$ which gives $\rho_i(k+t) = \rho_i(k) \left(1 - \frac{\Delta t v_i}{L_i}\right)^t$. This is not an acceptable physical solution and thus should be excluded.

Remark 2. *Satisfying requirements 4 and 6 implies imposing $v_i = \frac{L_i}{\Delta t}$. Given that the velocity v is an exogenous parameter, this imposes a strict requirement on the space discretization of the road segments. However, any given road segment might not be strictly into cells of exact length $v_i \cdot \Delta t$, and in most cases a cell of length $L \in (0, v_i \Delta t)$ will remain. There are multiple solutions to this issue:*

- *Approximate the length of each road segment to be a multiple of $v_i \cdot \Delta t$. The relative rounding error decreases as the road gets longer and the discretization Δt gets smaller.*
- *Change the dynamics of the last cell in each road segment to have a special case that allows for vehicles to be fully discharged when the supply of the downstream link allows it. This makes the dynamics equations and optimization problem more complicated.*
- *Accept this model limitation and have a small amount of density stuck in the network. This is not so bad in practice, since the number of vehicles stuck in a link decreases exponentially with time.*

C FIFO condition at the origin

It is important to note that the FIFO condition is violated in its strict sense even within the network. The flow propagation model assumes that all the flow within a cell is uniformly distributed according to the individual commodity ratios regardless of when the vehicles arrived at the cell.

Example 1. Consider the following simple example. There are two commodities a, b in cell i with 10 vehicles of each commodity at time k . At time $k + 1$, 10 vehicles exit the cell (5 of a and 5 of b by the FIFO rule) and 10 new vehicles (3 of a and 7 of b) enter the cell. The new ratio of vehicles at i is 8 a to 12 b . At time $k + 2$, once again 10 vehicles exit the network. According to the cell level FIFO rule, the 10 vehicles will consist of 4 a 's and 6 b 's. However, the first 10 cars of those currently in cell i came at the ratio of 1:1 and truly satisfying the FIFO rule would require the 10 exiting vehicles to consist of 5 a 's and 5 b 's.

Remark 3 (Cell-level multicommodity FIFO condition). *The strict multicommodity FIFO condition is not satisfied in most traffic flow models, but this is considered to be acceptable by the traffic modeling community. The multicommodity FIFO condition is therefore in practice at best limited to the cell level.*

While this argument is satisfactory at the interior of the network, due to the bounded number of vehicles in a given cell (due to the jam density), the FIFO violation can be significant at the boundaries of the network. The number of vehicles in a source buffer at the boundary of the network can be arbitrarily large, and thus, there is no bound on how badly the multi-commodity FIFO constraint can be violated.

A simple extension that limits the mixing of vehicles entering at different time steps is to have a series of source buffers at the boundary, where the multicommodity FIFO constraint is enforced when vehicles move between the buffers. In this model, the commodity ratios are maintained separately for each buffer. Any vehicles that enter the network at a given time step are added to the last active (non-empty buffer) buffer. This restricts the violation of the FIFO condition across multiple steps to the capacity of a single buffer.

The capacity of each buffer is chosen such that the buffer can satisfy the maximum supply of the first cell of the network boundary that the buffer serves. This prevents artificial delays at the source. The buffer capacity is set to $\Delta t \cdot F_b$ where b is the boundary cell.

The limitation of this model is that we need to maintain $\frac{l^{\max}}{\Delta t \cdot F_b}$ buffers per source, where l^{\max} is the maximum queue length at the boundary and $\Delta t \cdot F_b$ is the capacity of each buffer. Let $l_{i,c}^b(k)$ be the number of vehicles of commodity c in the b^{th} buffer for source node i at time step k and $l_i^b(k)$ its sum over all the commodities.

The buffers are updated as follows:

- First, given $f_i^{\text{in}}(k)$, move flow out of the initial buffer

$$f_{i,c}^{\text{in}}(k) = \frac{l_{i,c}^1(k)}{l_i^1(k)} f_i^{\text{in}}(k) \quad \forall c \in \mathcal{C} \quad (44)$$

$$l_{i,c}^1(k+1) = l_{i,c}^1(k) - f_{i,c}^{\text{in}}(k) \quad \forall c \in \mathcal{C} \quad (45)$$

- Let B be the number of buffers in use. Iterate through the buffers and push flow upstream using the following algorithm.

Algorithm 1 Update buffers

for $b = 1$ to $B - 1$ **do**
 $\Delta l^b = \min(L - l_i^b(k+1), l_i^{b+1}(k+1))$ \triangleright Maximum flow that can enter buffer b
 $l_{i,c}^b(k+1) = l_{i,c}^b(k+1) + \frac{l_{i,c}^{b+1}(k+1)}{l_i^{b+1}(k+1)} \Delta l^b \quad \forall c \in C$ \triangleright Per-commodity flow entering buffer b
 $l_{i,c}^{b+1}(k+1) = l_{i,c}^{b+1}(k+1) - \frac{l_{i,c}^{b+1}(k+1)}{l_i^{b+1}(k+1)} \Delta l^b \quad \forall c \in C$ \triangleright Per-commodity flow leaving buffer $b + 1$
end for
 $\Delta d = d_i(k)$ \triangleright Total demand at time step $k + 1$
 $b = B$
 \triangleright Allocate the demand to last buffer and create new buffers if needed
while $\Delta d > 0$ **do**
 $\Delta l^b = \min(L - l_i^b(k+1), \Delta d)$ \triangleright Maximum flow that can enter buffer b
 $l_{i,c}^b(k+1) = l_{i,c}^b(k+1) + \frac{d_{i,c}(k+1)}{d_i(k+1)} \Delta l^b \quad \forall c \in C$ \triangleright Per-commodity flow entering buffer b
 $b = b + 1$
end while

The only demand that is exposed to the system dynamics and the optimization problem is the demand that is captured in the first buffer.

D Explicit solutions for 1×2 , 2×1 and 2×2 junctions

In this section we present the explicit solutions used for the dynamics of the optimization problem and in the software implementation. We restrict our attention to 1×2 , 2×1 , and 2×2 junctions, since most road networks can be modeled using only these junction types.

(1×2) diverge junction

The explicit solution was already derived in section 3.1.1 for the general $(1 \times n)$ case.

(2×1) merge

Let i_1 be one incoming link, i_2 be the other incoming link and j be the outgoing link. As shown in Figure 10a, we distinguish 3 cases based on where the priority vector intersects the demand constraints. The cases are $\frac{P_1}{P_2} > \left(\frac{P_1}{P_2}\right)_{max}$, $\left(\frac{P_1}{P_2}\right)_{min} \leq \frac{P_1}{P_2} \leq \left(\frac{P_1}{P_2}\right)_{max}$ and $\frac{P_1}{P_2} < \left(\frac{P_1}{P_2}\right)_{min}$, where $\left(\frac{P_1}{P_2}\right)_{max} = \frac{\delta_{i_1}}{f_j^{in} - \delta_{i_1}}$ and $\left(\frac{P_1}{P_2}\right)_{min} = \frac{f_j^{in} - \delta_{i_2}}{\delta_{i_2}}$.

In all cases, the flow into the outgoing link is the minimum of the total supply and demand values.

$$f_j^{in}(k) = \min(\delta_{i_1}(k) + \delta_{i_2}(k), \sigma_j(k)) \quad (46)$$

The flow out of incoming links depends on the priority vector and is given by the solution to the optimization problem in equation (21).

$$f_{i_1}^{out}(k) = \begin{cases} \delta_{i_1}(k) & \text{if } P_{i_1} (f_j^{in}(k) - \delta_{i_1}(k)) > \delta_{i_1}(k) P_{i_2} \\ f_j^{in}(k) - \delta_{i_2}(k) & \text{if } P_{i_2} (f_j^{in}(k) - \delta_{i_2}(k)) > \delta_{i_2}(k) P_{i_1} \\ P_{i_1} f_j^{in}(k) & \text{otherwise} \end{cases}$$
$$f_{i_2}^{out}(k) = f_j^{in}(k) - f_{i_1}^{out}(k) \quad (47)$$

The flow out by commodity can then be computed as follows.

$$f_{i,c}^{out}(k) = \frac{\rho_{i,c}(k)}{\rho_i(k)} f_i^{out}(k) \quad \forall i \in \{i_1, i_2\} \quad (48)$$

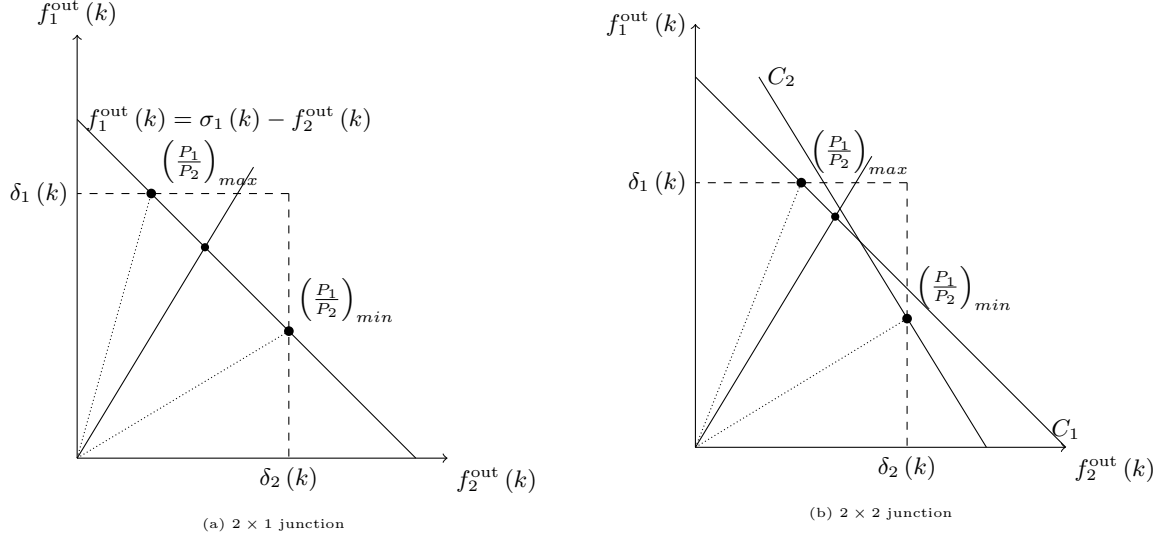


Figure 10: Graph displaying the constraints for different junctions.

(2×2) merge and diverge

Let i_1, i_2 be the incoming links and j_1, j_2 be the outgoing links. To simplify the notation, we use the following shorthand:

- drop the time index k
- $\delta_1 = \delta_{i_1}, \delta_2 = \delta_{i_2}$
- $\sigma_1 = \sigma_{j_1}, \sigma_2 = \sigma_{j_2}$
- $P_1 = P_{i_1}, P_2 = P_{i_2}$

The aggregate split ratios are computed as follows.

$$\beta_{ij}(k) = \frac{1}{\rho_i(k)} \sum_{c=1}^C \rho_{i,c}(k) \beta_{ij,c}(k) \quad \forall (i, j) \in \{1, 2\} \times \{1, 2\} \quad (49)$$

As shown in Figure 10b, we once again distinguish 3 cases based on where the priority vector intersects the demand constraints. The cases are $\frac{P_1}{P_2} > \left(\frac{P_1}{P_2}\right)_{max}$, $\left(\frac{P_1}{P_2}\right)_{min} \leq \frac{P_1}{P_2} \leq \left(\frac{P_1}{P_2}\right)_{max}$ and $\frac{P_1}{P_2} < \left(\frac{P_1}{P_2}\right)_{min}$.

The values of $\left(\frac{P_1}{P_2}\right)_{max}$ (and $\left(\frac{P_1}{P_2}\right)_{min}$) can be obtained by plugging substituting δ_1 (and resp. δ_2) for f_1^{out} (and resp. f_1^{out}) in the equations C_j for the supply constraints.

$$C_j : \beta_{1j} f_1^{\text{out}} + \beta_{2j} f_2^{\text{out}} = \sigma_j \quad (50)$$

Thus, we obtain:

$$\left(\frac{P_1}{P_2}\right)_{max} = \frac{\delta_1}{\min\left(\delta_2, \frac{\sigma_1 - \beta_{11}\delta_1}{\beta_{21}}, \frac{\sigma_2 - \beta_{12}\delta_1}{\beta_{22}}\right)} \quad (51)$$

$$\left(\frac{P_1}{P_2}\right)_{min} = \frac{\min\left(\delta_1, \frac{\sigma_1 - \beta_{21}\delta_2}{\beta_{11}}, \frac{\sigma_2 - \beta_{22}\delta_2}{\beta_{12}}\right)}{\delta_2} \quad (52)$$

We can then compute the flow out of incoming links using the optimization problem in equation (23). Since the problem is symmetric in i_1 and i_2 , we just solve it for $i_1 = 1$.

- If $\frac{P_1}{P_2} > \left(\frac{P_1}{P_2}\right)_{max}$, the solution is the intersection of $f_1^{out} = \delta_1$ and the most constraining supply constraint (C_1 in figure 10b), and results in the trivial solution of $f_1^{out} = \delta_1$.
- If $\frac{P_1}{P_2} < \left(\frac{P_1}{P_2}\right)_{min}$, the solution is the intersection of $f_2^{out} = \delta_2$ and the most constraining supply constraint (C_2 in figure 10b), and therefore we obtain:

$$f_1^{out} = \min\left(\delta_1, \frac{\sigma_1 - \beta_{21}\delta_2}{\beta_{11}}, \frac{\sigma_2 - \beta_{22}\delta_2}{\beta_{12}}\right)$$
- If $\left(\frac{P_1}{P_2}\right)_{min} \leq \frac{P_1}{P_2} \leq \left(\frac{P_1}{P_2}\right)_{max}$, the solution lies at the point where the priority vector intersects the most constraining supply constraint (C_2 in figure 10b)), and therefore we obtain:

$$f_1^{out} = \min\left(\frac{P_1\sigma_1}{P_1\beta_{11} + P_2\beta_{21}}, \frac{P_1\sigma_2}{P_1\beta_{12} + P_2\beta_{22}}\right)$$

This gives us the explicit following explicit solution for f_1^{out} at a 2×2 junction.

$$f_1^{out} = \begin{cases} \delta_1 & \text{if } \frac{P_1}{P_2} > \frac{\delta_1}{\min\left(\delta_2, \frac{\sigma_1 - \beta_{11}\delta_1}{\beta_{21}}, \frac{\sigma_2 - \beta_{12}\delta_1}{\beta_{22}}\right)} \\ \min\left(\delta_1, \frac{\sigma_1 - \beta_{21}\delta_2}{\beta_{11}}, \frac{\sigma_2 - \beta_{22}\delta_2}{\beta_{12}}\right) & \text{if } \frac{P_1}{P_2} < \frac{\min\left(\delta_1, \frac{\sigma_1 - \beta_{21}\delta_2}{\beta_{11}}, \frac{\sigma_2 - \beta_{22}\delta_2}{\beta_{12}}\right)}{\delta_2} \\ \min\left(\frac{P_1\sigma_1}{P_1\beta_{11} + P_2\beta_{21}}, \frac{P_1\sigma_2}{P_1\beta_{12} + P_2\beta_{22}}\right) & \text{otherwise} \end{cases} \quad (53)$$

f_2^{out} is obtained by symmetry.

Finally, we can now compute the flow out of incoming links by commodity.

$$f_{i,c}^{out} = \frac{\rho_{i,k}}{\rho_i} f_i^{out} \quad \forall i \in \{1, 2\}, \forall c \in C \quad (54)$$

E Computing the partial derivatives of the constraints with respect to the state variables $\left(\frac{\partial}{\partial x}\right)$

We first iterate through the three classes of variables density, outflow and inflow. All unlisted derivatives evaluate to zero.

Commodity density $\rho_{i,c}(k)$ from equations (H1a, H1b, H1c, H1d))

$$\frac{\partial \rho_{i,c}(k)}{\partial \rho_{i,c}(k-1)} = 1, \quad \forall c \in C, \forall i \in \mathcal{A} \setminus \mathcal{B}, \quad \forall k \in \llbracket 1, T \rrbracket \quad (55)$$

$$\frac{\partial \rho_{i,c}(k)}{\partial f_{i,c}^{in}(k-1)} = \frac{\Delta t}{L_i}, \quad \forall c \in C, \forall i \in \mathcal{A} \setminus \mathcal{B}, \quad \forall k \in \llbracket 1, T \rrbracket \quad (56)$$

$$\frac{\partial \rho_{i,c}(k)}{\partial f_{i,c}^{out}(k-1)} = -\frac{\Delta t}{L_i}, \quad \forall c \in C, \forall i \in \mathcal{A} \setminus \mathcal{S}, \quad \forall k \in \llbracket 1, T \rrbracket \quad (57)$$

Flow in $f_{i,c}^{in}$ from equation (H6)

$$\frac{\partial f_{j,c}^{in}}{\partial f_{i,c}^{out}} = \beta_{ij,c} \quad \forall i \in \mathcal{J}_z^{in}, \forall j \in \mathcal{J}_z^{out}, \forall z \in \mathcal{J} \quad \forall k \in \llbracket 1, T-1 \rrbracket \quad (58)$$

Computing the partial derivatives of the flow out $f_{i,c}^{out}$ is requires a much more involved process. We begin by computing the following intermediate partial derivatives:

- Computing $\frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \delta_i(k)}{\rho_i(k)} \right)$:

From equations (H2a, H2b),

$$\delta_i(k) = \begin{cases} \min(F_i, v_i \rho_i(k)) & \forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket \\ \min\left(F_i, \frac{\rho_i(k) L_i}{\Delta t}\right) & \forall i \in \mathcal{B}, \forall k \in \llbracket 0, T_f \rrbracket \end{cases} \quad (59)$$

which gives the following equations:

$$\frac{\rho_{i,c}(k)}{\rho_i(k)} \delta_i(k) = \begin{cases} \min\left(\frac{\rho_{i,c}(k)}{\rho_i(k)} F_i, \rho_{i,c}(k) v_i\right) & \forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket \\ \min\left(\frac{\rho_{i,c}(k)}{\rho_i(k)} F_i, \frac{\rho_{i,c}(k) L_i}{\Delta t}\right) & \forall i \in \mathcal{B}, \forall k \in \llbracket 0, T_f \rrbracket \end{cases} \quad (60)$$

Using equation (1),

$\forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket$

$$\frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k)}{\rho_i(k)} \delta_i(k) \right) = \begin{cases} \frac{(\rho_i(k) - \rho_{i,c}(k))}{\rho_i(k)^2} F_i & \text{if } F_i < v_i \rho_i(k) \\ v_i & \text{otherwise} \end{cases} \quad (61)$$

$\forall i \in \mathcal{B}, \forall k \in \llbracket 0, T_f \rrbracket$

$$\frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k)}{\rho_i(k)} \delta_i(k) \right) = \begin{cases} \frac{(\rho_i(k) - \rho_{i,c}(k))}{\rho_i(k)^2} F_i & \text{if } F_i < \frac{L_i}{\Delta t} \rho_i(k) \\ \frac{L_i}{\Delta t} & \text{otherwise} \end{cases} \quad (62)$$

Remark 4. If $\rho_i(k) = 0$, then $F_i > \frac{L_i}{\Delta t} \rho_i(k)$ or $F_i > v_i \rho_i(k)$, so the derivatives are well defined.

- Computing $\frac{\partial}{\partial \rho_{i,c'}(k)} \left(\frac{\rho_{i,c}(k) \delta_i(k)}{\rho_i(k)} \right)$: (Note the two different commodities c and c' .)

$\forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket$

$$\frac{\partial}{\partial \rho_{i,c'}(k)} \left(\frac{\rho_{i,c}(k)}{\rho_i(k)} \delta_i(k) \right) = \begin{cases} \frac{-\rho_{i,c}(k)}{\rho_i(k)^2} F_i & \text{if } F_i < v_i \rho_i(k) \\ 0 & \text{otherwise} \end{cases} \quad (63)$$

$\forall i \in \mathcal{B}, \forall k \in \llbracket 0, T_f \rrbracket$

$$\frac{\partial}{\partial \rho_{i,c'}(k)} \left(\frac{\rho_{i,c}(k)}{\rho_i(k)} \delta_i(k) \right) = \begin{cases} \frac{-\rho_{i,c}(k)}{\rho_i(k)^2} F_i & \text{if } F_i < \frac{L_i}{\Delta t} \rho_i(k) \\ 0 & \text{otherwise} \end{cases} \quad (64)$$

Remark 5. If $\rho_i(k) = 0$, then $F_i > \frac{L_i}{\Delta t} \rho_i(k)$ or $F_i > v_i \rho_i(k)$, so the derivatives are well defined.

- Computing $\frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right)$ and $\frac{\partial}{\partial \rho_{i,c'}(k)} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right)$ when $i \neq j$:

Note that $\sigma_j(k)$ does not contain $\rho_i(k)$ terms.

$$\frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k)}{\rho_i(k)} \sigma_j(k) \right) = \frac{(\rho_i(k) - \rho_{i,c}(k))}{\rho_i(k)^2} \sigma_j(k) \quad (65)$$

$$\frac{\partial}{\partial \rho_{i,c'}(k)} \left(\frac{\rho_{i,c}(k)}{\rho_i(k)} \sigma_j(k) \right) = \frac{-\rho_{i,c}(k)}{\rho_i(k)^2} \sigma_j(k) \quad (66)$$

Remark 6. *This partial derivative is only needed in cases where the junction is strictly supply constrained and $\rho_i(k) > 0$, so we can ignore the fact that the derivative is undefined at $\rho_i(k) = 0$.*

- Computing $\frac{\partial}{\partial \rho_{j,c'}} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right)$:

From equations (H3a, H3b),

$$\sigma_j(k) = \begin{cases} \min \left(F_i, w_j \left(\rho_j^{\text{jam}} - \rho_j(k) \right) \right) & \forall j \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket \\ F_j & \forall j \in \mathcal{S}, \forall k \in \llbracket 0, T_f \rrbracket \end{cases} \quad (67)$$

$$\frac{\rho_{i,c}(k)}{\rho_i(k)} \sigma_j(k) = \begin{cases} \min \left(\frac{\rho_{i,c}(k)}{\rho_i(k)} F_j, \frac{\rho_{i,c}(k)}{\rho_i(k)} w_j \left(\rho_j^{\text{jam}} - \rho_j(k) \right) \right) & \forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket \\ \frac{\rho_{i,c}(k)}{\rho_i(k)} F_j & \forall i \in \mathcal{S}, \forall k \in \llbracket 0, T_f \rrbracket \end{cases} \quad (68)$$

$\forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket$

$$\frac{\partial}{\partial \rho_{j,c'}} \left(\frac{\rho_{i,c}(k)}{\rho_i(k)} \sigma_j(k) \right) = \begin{cases} 0 & \text{if } F_j < w_j \left(\rho_j^{\text{jam}} - \rho_j(k) \right) \\ -\frac{\rho_{i,c}(k)}{\rho_i(k)} w_j & \text{otherwise} \end{cases} \quad (69)$$

$\forall i \in \mathcal{S}, \forall k \in \llbracket 0, T_f \rrbracket$

$$\frac{\partial}{\partial \rho_{j,c'}} \left(\frac{\rho_{i,c}(k)}{\rho_i(k)} \sigma_j(k) \right) = 0 \quad (70)$$

Remark 7. *This partial derivative is only needed in cases where the junction is strictly supply constrained and $\rho_i(k) > 0$, so we can ignore the fact that the derivative is undefined at $\rho_i(k) = 0$.*

- Computing $\frac{\partial}{\partial \rho_{i,c}} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k) \beta_{ij}(k)} \right)$:

From equation (H4),

$$\beta_{ij}(k) = \frac{1}{\rho_i(k)} \sum_{c' \in \mathcal{C}} \rho_{i,c'}(k) \beta_{ij,c'}(k) \quad \forall k \in \llbracket 0, T_f \rrbracket \quad (71)$$

Let $\kappa_{ij}(k) = \sum_{c' \in \mathcal{C}} \rho_{i,c'}(k) \beta_{ij,c'}(k)$

$$\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k) \beta_{ij}(k)} = \frac{\rho_{i,c}(k) \sigma_j(k)}{\kappa_{ij}(k)} \quad (72)$$

$$\frac{\partial}{\partial \rho_{i,c}} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k) \beta_{ij}(k)} \right) = \frac{\kappa_{ij}(k) \sigma_j(k) - \rho_{i,c}(k) \sigma_j(k) \beta_{ij,c}(k)}{\kappa_{ij}(k)^2} \quad (73)$$

Remark 8. *This partial derivative is only needed in cases where the junction is strictly supply constrained and $\rho_i(k) > 0$, so we can ignore the fact that the derivative is undefined at $\rho_i(k) = 0$.*

- Computing $\frac{\partial}{\partial \rho_{i,c'}} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k) \beta_{ij}(k)} \right)$:

$$\frac{\partial}{\partial \rho_{i,c'}} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k) \beta_{ij}(k)} \right) = \frac{-\rho_{i,c}(k) \sigma_j(k) \beta_{ij,c'}(k)}{\kappa_{ij}(k)^2} \quad (74)$$

Remark 9. The $\rho_i(k) = 0$ condition is just as in the previous case.

- Computing $\frac{\partial}{\partial \rho_{j,c'}} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k) \beta_{ij}(k)} \right)$:

$$\frac{\partial}{\partial \rho_{j,c'}} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k) \beta_{ij}(k)} \right) = \frac{1}{\beta_{ij}(k)} \frac{\partial}{\partial \rho_{i,c'}} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right) \quad (75)$$

which can then be simplified using equations (69, 70).

Now we can proceed to computing the partial derivatives of $f_{i,c}^{\text{out}}$.

Definition 19 (Demand-constrained junction). A junction is demand-constrained if the flow through the junction is limited by the incoming flow of cell i . We denote this condition by $DC(i)$.

Definition 20 (Supply-constrained junction). A junction is supply-constrained if the flow through the junction is limited by the outgoing flow into some outgoing cell j . We denote this condition by $SC(j)$.

Solution for 1×2 junctions

The solutions to all the partial derivatives that appear in the expressions below have already been solved explicitly.

From equation (H5a),

$$f_{i,c}^{\text{out}} = \frac{\rho_{i,c}}{\rho_i} \min \left(\left\{ \frac{\sigma_j(k)}{\beta_{ij}(k)}, \forall j \in \mathcal{J}_z^{\text{out}} \mid \beta_{ij}(k) > 0 \right\}, \delta_i(k) \right) \quad \forall z \in \mathcal{J}_{1 \times n}, \forall i \in \mathcal{J}_z^{\text{in}} \quad (76)$$

$$\frac{\partial f_{i,c}^{\text{out}}}{\partial \rho_{i,c}(k)} = \begin{cases} \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \delta_i(k)}{\rho_i(k)} \right) & \text{if } DC(i) \\ \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \sigma_i(k)}{\rho_i(k) \beta_{ij}(k)} \right) & \text{if } SC(j) \end{cases} \quad (77)$$

$$\frac{\partial f_{i,c'}^{\text{out}}}{\partial \rho_{i,c}(k)} = \begin{cases} \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c'}(k) \delta_i(k)}{\rho_i(k)} \right) & \text{if } DC(i) \\ \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c'}(k) \sigma_i(k)}{\rho_i(k) \beta_{ij}(k)} \right) & \text{if } SC(j) \end{cases} \quad (78)$$

$$\frac{\partial f_{i,c}^{\text{out}}}{\partial \rho_{j,c'}(k)} = \begin{cases} 0 & \text{if } DC(i) \\ \frac{\partial}{\partial \rho_{j,c'}(k)} \left(\frac{\rho_{i,c}(k) \sigma_i(k)}{\rho_i(k) \beta_{ij}(k)} \right) & \text{if } SC(j) \end{cases} \quad (79)$$

Remark 10. It is important to note that these derivatives are undefined if the junction is both supply and demand-constrained. However, this can only occur if the density of the cell is exactly equal to the value at which the demand and supply constraints meet. This is extremely unlikely in practice with floating point numerical operations. In the rare event that it does occur, we assume that the junction is supply-constrained.

Solution for 2×1 junctions

The solutions to all the partial derivatives that appear in the expressions below have already been solved explicitly.

From equation (H5b)

$$f_{i,c}^{\text{out}} = \frac{\rho_{i,c}}{\rho_i} \begin{cases} \delta_i & \text{if } P_i(\min(\delta_i + \delta_{\underline{i}}, \sigma_j) - \delta_i) > \delta_i P_{\underline{i}} \\ \min(\delta_i + \delta_{\underline{i}}, \sigma_j) - \delta_{\underline{i}} & \text{if } P_{\underline{i}}(\min(\delta_i + \delta_{\underline{i}}, \sigma_j) - \delta_{\underline{i}}) > \delta_{\underline{i}} P_i \\ P_i \min(\delta_i + \delta_{\underline{i}}, \sigma_j) & \text{otherwise} \end{cases} \quad \forall z \in \mathcal{J}_{2 \times 1}, \forall i \in \mathcal{J}_z^{\text{in}} \quad (80)$$

case 1: $P_i (\min (\delta_i + \delta_{\underline{i}}, \sigma_j) - \delta_i) > \delta_i P_{\underline{i}}$

$$\frac{\partial f_{i,c}^{\text{out}}}{\partial \rho_{i,c}(k)} = \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \delta_i(k)}{\rho_i(k)} \right) \quad (81)$$

$$\frac{\partial f_{i,c'}^{\text{out}}}{\partial \rho_{i,c}(k)} = \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c'}(k) \delta_i(k)}{\rho_i(k)} \right) \quad (82)$$

$$\frac{\partial f_{i,c'}^{\text{out}}}{\partial \rho_{j,c}(k)} = 0 \quad (83)$$

case 2: $P_{\underline{i}} (\min (\delta_i + \delta_{\underline{i}}, \sigma_j) - \delta_{\underline{i}}) > \delta_{\underline{i}} P_i$

$$\frac{\partial f_{i,c}^{\text{out}}}{\partial \rho_{i,c}(k)} = \begin{cases} \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \delta_i(k)}{\rho_i(k)} \right) & \text{if } \delta_i + \delta_{\underline{i}} < \sigma_j \\ \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right) - \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \delta_{\underline{i}}(k)}{\rho_i(k)} \right) & \text{otherwise} \end{cases} \quad (84)$$

$$\frac{\partial f_{i,c}^{\text{out}}}{\partial \rho_{i,c'}(k)} = \begin{cases} \frac{\partial}{\partial \rho_{i,c'}(k)} \left(\frac{\rho_{i,c}(k) \delta_i(k)}{\rho_i(k)} \right) & \text{if } \delta_i + \delta_{\underline{i}} < \sigma_j \\ \frac{\partial}{\partial \rho_{i,c'}(k)} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right) - \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \delta_{\underline{i}}(k)}{\rho_i(k)} \right) & \text{otherwise} \end{cases} \quad (85)$$

$$\frac{\partial f_{i,c}^{\text{out}}}{\partial \rho_{j,c'}(k)} = \begin{cases} 0 & \text{if } \delta_i + \delta_{\underline{i}} < \sigma_j \\ \frac{\partial}{\partial \rho_{j,c'}(k)} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right) & \text{otherwise} \end{cases} \quad (86)$$

case 3: otherwise

$$\frac{\partial f_{i,c}^{\text{out}}}{\partial \rho_{i,c}(k)} = \begin{cases} P_i \left(\frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \delta_i(k)}{\rho_i(k)} \right) + \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \delta_{\underline{i}}(k)}{\rho_i(k)} \right) \right) & \text{if } \delta_i + \delta_{\underline{i}} < \sigma_j \\ P_i \left(\frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right) \right) & \text{otherwise} \end{cases} \quad (87)$$

$$\frac{\partial f_{i,c}^{\text{out}}}{\partial \rho_{i,c'}(k)} = \begin{cases} P_i \left(\frac{\partial}{\partial \rho_{i,c'}(k)} \left(\frac{\rho_{i,c}(k) \delta_i(k)}{\rho_i(k)} \right) + \frac{\partial}{\partial \rho_{i,c}(k)} \left(\frac{\rho_{i,c}(k) \delta_{\underline{i}}(k)}{\rho_i(k)} \right) \right) & \text{if } \delta_i + \delta_{\underline{i}} < \sigma_j \\ P_i \left(\frac{\partial}{\partial \rho_{i,c'}(k)} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right) \right) & \text{otherwise} \end{cases} \quad (88)$$

$$\frac{\partial f_{i,c}^{\text{out}}}{\partial \rho_{j,c'}(k)} = \begin{cases} 0 & \text{if } \delta_i + \delta_{\underline{i}} < \sigma_j \\ P_i \left(\frac{\partial}{\partial \rho_{j,c'}(k)} \left(\frac{\rho_{i,c}(k) \sigma_j(k)}{\rho_i(k)} \right) \right) & \text{otherwise} \end{cases} \quad (89)$$

Solution for 2×2 junctions

The solution for the 2×2 junctions can be obtained using a similar set of computations, but is omitted here for readability and due to length constraints.

This concludes the computation of all the partial derivatives required for computing the gradient of the system using the discrete adjoint method.

Remark 11. *If we do not have closed form solutions for the junctions, it may not be possible to compute the explicit partial derivatives of the outgoing flow with respect to the partial densities of the incoming and outgoing links of the junction. However, for any junction that cannot be solved explicitly, it is still possible to compute $\frac{\partial f_{i,c}^{\text{out}}(k)}{\partial \rho_{i',c'}(k)}$ for all $i, i' \in \mathcal{J}_z^{\text{in}} \cup \mathcal{J}_z^{\text{out}}$ and $c, c' \in \mathcal{C}$ with a finite differences method using $|\mathcal{J}_z^{\text{in}}| \cdot |\mathcal{J}_z^{\text{out}}| \cdot |\mathcal{C}|$ local simulations of just the junction dynamics (not the entire system). We can then continue to use the adjoint method while numerically differentiating these junctions that do not admit an explicit solution. This local finite differences method can still allow a very efficient computation of the gradient for heterogeneous networks with some junctions or sub-networks that contain complex dynamics.*

References

- Aswani, A. and Tomlin, C. (2011). Game-theoretic routing of GPS-assisted vehicles for energy efficiency. In *American Control Conference (ACC), 2011*, pages 3375–3380. IEEE.
- Bayen, A. M., Raffard, R. L., and Tomlin, C. J. (2006). Adjoint-based control of a new Eulerian network model of air traffic flow. *Control Systems Technology, IEEE Transactions on*, 14(5):804–818.
- Beckman, M., McGuire, C. B., and Winsten, C. B. (1956). *Studies in the Economics of Transportation*. Yale University Press, New Haven.
- Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific.
- Boese, K. D., Kahng, A. B., and Muddu, S. (1994). A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters*, 16(2):101 – 113.
- Braess, D. (1968). Über ein Paradoxon aus der Verkehrsplanung. *Mathematical Methods of Operations Research*, 12(1):258–268.
- Carey, M. (1992). Nonconvexity of the dynamic traffic assignment problem. *Transportation Research Part B: Methodological*, 26(2):127–133.
- Chen, C., Petty, K., Skabardonis, A., Varaiya, P., and Jia, Z. (2001). Freeway performance measurement system: mining loop detector data. *Transportation Research Record: Journal of the Transportation Research Board*, 1748(1):96–102.
- Daganzo, C. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287.
- Daganzo, C. (1995). The cell transmission model, part II: Network traffic. *Transportation Research Part B*, 29(2):79–93.
- Delle Monache, M. L., Reilly, J., Samaranayake, S., Krichene, W., Goatin, P., and Bayen, A. (2013). A PDE-ODE Model for a Junction with Ramp Buffer. *SIAM Journal on Applied Mathematics (in review)*.
- Dervisoglu, G., Gomes, G., Kwon, J., Horowitz, R., and Varaiya, P. (2009). Automatic calibration of the fundamental diagram and empirical observations on capacity. In *Transportation Research Board 88th Annual Meeting*, number 09-3159.
- Duffy, A. (2009). An Introduction to Gradient Computation by the Discrete Adjoint Method. Technical report, Florida State University.
- Garavello, M. and Piccoli, B. (2006). *Traffic flow on networks*. American institute of mathematical sciences Springfield,, USA.
- Giles, M. B. M. and Pierce, N. A. N. (2000). An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65(3):393–415.
- Giles, M. B. M. B. and Pierce, N. A. N. (1997). Adjoint equations in CFD : duality , boundary conditions and solution behaviour. *AIAA paper*, 97(1850):182–198.
- Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306.
- Gomes, G. and Horowitz, R. (2006). Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C: Emerging Technologies*, 14(4):244–262.
- Jameson, A. and Martinelli, L. (2000). *Aerodynamic shape optimization techniques based on control theory*. Springer.

- Kelly, F. P. (1991). Network routing. *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, 337(1647):343–367.
- Korilis, Y. A., Lazar, A. A., and Orda, A. (1997). Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Transactions on Networking (TON)*, 5(1):161–173.
- Koutsoupias, E. and Papadimitriou, C. (1999). Worst-case equilibria. In *Proceedings of the 16th annual conference on Theoretical aspects of computer science*, pages 404–413. Springer-Verlag.
- Krichene, W., Reilly, J., Amin, S., and Bayen, A. M. (2013). Stackelberg Routing on Parallel Networks with Horizontal Queues. *IEEE Transactions on Automatic Control (in review)*.
- Leveque, R. (2002). *Finite volume methods for hyperbolic problems*. Cambridge University Press, Cambridge, UK.
- Lighthill, M. J. and Whitham, G. B. (1955). On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 229(1178):317–345.
- Marti, R. (2003). Multi-start methods. In Glover, F. and Kochenberger, G. A., editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research and Management Science*, pages 355–368. Springer US.
- Merchant, D. K. and Nemhauser, G. L. (1978a). A Model and an Algorithm for the Dynamic Traffic Assignment Problems. *Transportation science*, 12(3):183–199.
- Merchant, D. K. and Nemhauser, G. L. (1978b). Optimality conditions for a dynamic traffic assignment model. *Transportation Science*, 12(3):183–199.
- Morton, K. W. and Mayers, D. F. (2005). *Numerical solution of partial differential equations: an introduction*. Cambridge university press.
- Peeta, S. and Ziliaskopoulos, A. (2001). Foundations of Dynamic Traffic Assignment : The Past , the Present and the Future. *Networks and Spatial Economics*, pages 233–265.
- Raffard, R. (2008). An adjoint-based parameter identification algorithm applied to planar cell polarity signaling. *IEEE Transactions on Automatic Control*, (January):109–121.
- Reilly, J., Monache, M. L. D., Samaranayake, S., Krichene, W., Gaotin, P., and Bayen, A. (2013). An efficient method for coordinated ramp metering using the discrete adjoint method. *Journal of Optimization Theory and Applications*, *in review*.
- Richards, P. I. (1956). Shock waves on the highway. *Operations research*, 4(1):42–51.
- Riedmiller, M. and Braun, H. (1992). Rprop-a fast adaptive learning algorithm. In *Proceedings of the International Symposium on Computer and Information Science VII*, *Universitat*. Citeseer.
- Roughgarden, T. (2001). Stackelberg scheduling strategies. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 104–113. ACM.
- Roughgarden, T. (2002). The Price of Anarchy is Independent of the Network. *Computer*, (May):1–24.
- Roughgarden, T. (2006). On the severity of Braess’s paradox: designing networks for selfish users is hard. *Journal of Computer and System Sciences*, 72(5):922–953.
- Roughgarden, T. and Tardos, É. (2004). Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 47(2):389–403.

- Swamy, C. (2007). The effectiveness of Stackelberg strategies and tolls for network congestion games. In *Symposium on Discrete Algorithms: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, volume 7, pages 1133–1142.
- Vickrey, W. S. (1969). Congestion theory and transport investment. *The American Economic Review*, 59(2):251–260.
- Wachter, A. and Biegler, L. T. (2005). *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*.
- Wardrop, J. G. (1952). Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1:325–378.
- Ziliaskopoulos, A. K. (2000). A linear programming model for the single destination system optimum dynamic traffic assignment problem. *Transportation science*, 34(1):37–49.