



Transportation Science

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Discrete-Time System Optimal Dynamic Traffic Assignment (SO-DTA) with Partial Control for Physical Queuing Networks

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

To cite this article:

Samitha Samaranayake, Walid Krichene, Jack Reilly, Maria Laura Delle Monache, Paola Goatin, Alexandre Bayen (2018) Discrete-Time System Optimal Dynamic Traffic Assignment (SO-DTA) with Partial Control for Physical Queuing Networks. *Transportation Science* 52(4):982-1001. <https://doi.org/10.1287/trsc.2017.0800>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2018, INFORMS

Please scroll down for article—it is on subsequent pages




With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Discrete-Time System Optimal Dynamic Traffic Assignment (SO-DTA) with Partial Control for Physical Queuing Networks

Samitha Samaranayake,^a Walid Krichene,^b Jack Reilly,^c Maria Laura Delle Monache,^{d,e} Paola Goatin,^d Alexandre Bayen^{b,c}

^a School of Civil and Environmental Engineering, Cornell University, Ithaca, New York 14850; ^b Department of Electrical Engineering and Computer Sciences, University of California Berkeley, Berkeley, California 94702; ^c Department of Civil and Environmental Engineering, Institute of Transportation Studies, University of California Berkeley, Berkeley, California 94702; ^d Inria Sophia Antipolis Méditerranée, Université Côte d'Azur, Inria, CNRS, LJAD, 06902 Sophia Antipolis Cedex, France; ^e Université Grenoble Alpes, 38400 Saint-Martin-d'Hères, France

Contact: samitha@cornell.edu,  <http://orcid.org/0000-0002-5459-3898> (SS); walid@berkeley.edu (WK); jackdreilly@berkeley.edu (JR); ml.dellemonache@inria.fr (MLDM); paola.goatin@inria.fr (PG); bayen@berkeley.edu (AB)

Received: November 5, 2014

Revised: March 31, 2016; April 15, 2017

Accepted: June 26, 2017

Published Online in Articles in Advance:
June 4, 2018

<https://doi.org/10.1287/trsc.2017.0800>

Copyright: © 2018 INFORMS

Abstract. We consider the *System Optimal Dynamic Traffic Assignment (SO-DTA) problem with Partial Control* for general networks with physical queuing. Our goal 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 partial differential equation with a triangular flux function and a corresponding multicommodity junction solver. The partial control formulation generalizes the SO-DTA problem to consider cases where only a fraction of the total flow can be controlled, as may arise in the context of certain incentive schemes. This leads to a nonconvex multicommodity optimization problem. We define a multicommodity junction model that only requires full Lagrangian paths for the controllable agents, and aggregate turn ratios for the noncontrollable (selfish) agents. We show how the resulting finite horizon nonlinear optimal control problem can be efficiently solved using the discrete adjoint method, leading to gradient computations that are linear in the size of the state space and the controls.

Funding: 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.”

Supplemental Material: The online appendix is available at <https://doi.org/10.1287/trsc.2017.0800>.

Keywords: transportation: assignment models • network models • vehicle routing • networks/graphs: flow algorithms • multicommodity

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 extensively studied over the last 35 years, since the seminal works of Merchant and Nemhauser (1978a, b). There are two types of traffic assignments 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, McGuire, and Winsten (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 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). The UE-DTA problem, also commonly referred to as the *dynamic user equilibrium* (DUE) problem, is not addressed here, but we refer the reader to Friesz et al. (1993, 2013); Han, Friesz, and Yao (2013); Smith and Winsten (1995); and Wie, Tobin, and Carey (2002) for some solutions, strategies, and variants of the DUE problem. System optimal (SO) traffic

assignment, on the other hand, leads to optimal use of the network resources (Carey and Watling 2012; Qian, Shen, and Zhang 2012; Li, Waller, and Ziliaskopoulos 2003; Nie 2011), 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 an SO allocation (Vickrey 1969, Roughgarden 2002). 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 systemwide total travel time. This has been studied in the context of Stackelberg routing games (Korilis, Lazar, and Orda 1997; Roughgarden 2001; Swamy 2007), where the goal of the central controller is to assign routes to a fraction of the demand using a strategy that minimizes the systemwide 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 nondecreasing latency function and vertical queues. However, these assumptions are generally not satisfied in road traffic networks, with physical (or 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 strong simplifying assumptions. For example, Aswani and Tomlin (2011) use vertical queues and nondecreasing 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 nonlinear dynamics of the system. However, the SO-DTA problem with partial control cannot 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 also to 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 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 the boundary flows 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 physical 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; it is well accepted in the transportation community as a good first order approximation of road traffic dynamics for DTA (Lo and Szeto 2002; Ukkusuri, Han, and Doan 2012). We propose solving the SO-DTA-PC problem with the nonconvex traffic dynamics from Delle Monache et al. (2014) and limited OD data with complete split ratios as a non-linear optimal control problem. The multicommodity formulation that we propose can be directly used to solve SO-DTA problems with multiple origins and destinations.

The next challenge is finding efficient descent methods for this nonconvex optimal control problem. There is a vast literature on optimization techniques for nonconvex control problems (see Bertsekas 1999 and the references therein) that can be used to solve this problem. While gradient based methods do not guarantee converging to the optimal solution in nonconvex 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 many times. We show that the structure of our dynamical system allows for very efficient computation of the gradient via the discrete adjoint method (Bayen, Raffard, and Tomlin 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 further reduction of the complexity to $O(n + m)$ by exploiting the sparse nature of the forward system.

Note 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); 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) formulation of the SO-DTA-PC as a multicommodity finite horizon optimal control problem (our formulation allows for networks with multiple origins and destinations); (2) defining a multicommodity junction model for the network dynamics that reduces the burden of input data by using different input data models for controllable and noncontrollable commodities, (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 to illustrate 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, and introduce the notion of controllable and noncontrollable flows in the network. In Section 3, we define the forward model of the system with explicit solutions to the junction problems. Section 4 formulates the optimization problem, derives the adjoint system, and describes the gradient descent method. This is followed with numerical results in Section 5. We conclude with some final remarks and future research directions in Section 6.

2. Traffic Model

The aggregate traffic dynamics are modeled using a macroscopic traffic flow model based on the LWR PDE (Lighthill and Whitham 1955, Richards 1956). We use a multicommodity variant of the PDE model developed in Garavello and Piccoli (2006) with input buffers. The buffers are used to ensure that no vehicles are dropped from the system due to congestion propagating outside the bounds of the network, which is an important consideration in the optimal control setting, and is known in the PDE literature as imposing strong boundary conditions. We then use a Godunov discretization (Godunov 1959) of the network PDE model as explained in Reilly et al. (2015) to obtain an equivalent discrete model.

2.1. Network Definitions

The road network is divided into cells, indexed by $i \in \mathcal{A}$ where \mathcal{A} is the set of all cells. Note that a physical road link connecting two junctions may be divided into

multiple cells. We add a buffer (origin) cell, indexed by $i \in \mathcal{B}$ where $\mathcal{B} \subset \mathcal{A}$, at each entrance (origin) of the network to ensure that vehicles are not dropped from the system when congestion propagates to the entrance of the network (i.e., to impose strong boundary conditions). The sink cells, indexed by $i \in \mathcal{S} \subset \mathcal{A}$, correspond to the boundaries (destinations) where flows exit the network. Therefore, each origin o belongs to the set \mathcal{B} and each destination belongs to the set \mathcal{S} . Each junction, indexed by $z \in \mathcal{J}$, connects a set of incoming cells $\mathcal{J}_z^{\text{in}}$ to a set of outgoing cells $\mathcal{J}_z^{\text{out}}$. The total flow in the network is decomposed into a set of $|\mathcal{C}|$ commodities that correspond to different types of flow. Appendix A provides a full list of the notation that is used throughout the article with their corresponding descriptions.

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 cell i at time step k , denoted by $\rho_i(k)$, is the total number of vehicles on the cell during that time step divided by the length of the cell L_i . The vehicles in the cell 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 cell i at time step k , denoted by $\rho_{i,c}(k)$, is the total number of vehicles of commodity c on the cell during that time step divided by the length of the cell 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 cell at time step $k = 0$ and are denoted $\rho_{i,c}(0)$.

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

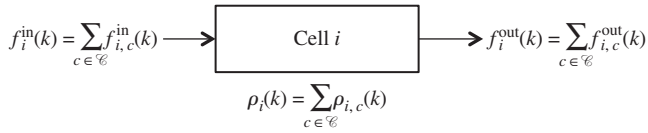
Definition 6 (Single Commodity Inflow and Outflow). The inflow (respectively, outflow) from a cell i at time step k corresponding to commodity c , denoted $f_{i,c}^{\text{in}}(k)$ (respectively, $f_{i,c}^{\text{out}}(k)$), is the total flow of commodity c leaving (respectively, entering) the cell at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

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

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

See Figure 1 for an illustration.

Figure 1. The Density (ρ), Inflow (f^{in}), and Outflow (f^{out}) of Cell i at Time Step k Includes Components Corresponding to Each Commodity That Flows Through Cell i



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 correspond to mass conservation. Time is indexed by $k \in [0, T_f = T - 1]$ corresponding to T time steps

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

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

with initial condition

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

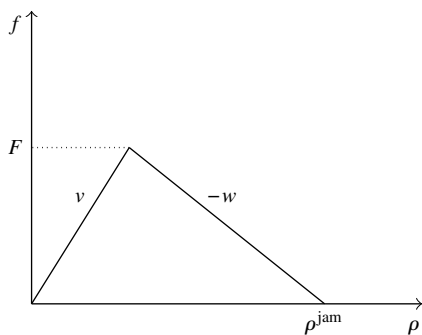
$$\rho_{i,c}(0) = 0, \quad \forall i \in \mathcal{S}, \forall c \in \mathcal{C}. \quad (7)$$

Prescribing traffic flows using a macroscopic traffic flow model based on the LWR PDE requires defining a flux function that describes the relationship between traffic density and flow (Daganzo 1994).

Assumption 1. The flux function defining the relationship between density and flow for each cell is given by a triangular fundamental diagram as shown in Figure 2. The parameters F , v , w , and ρ^{jam} are cell specific constants denoting, respectively, the maximum flow rate through the cell, the maximum (or jam) density of the cell, the free-flow (or congestion free) speed of the cell, and the speed at which congestion propagates backwards (congestion wave speed).

This is a first order approximation of the empirical relationship between flow and density. See Dervisoglu et al. (2009) for more details.

Figure 2. Triangular Fundamental Diagram



Assumption 2. FIFO property. We assume that no vehicles entering a cell at time step k will overtake vehicles that have already entered the cell at some time step $k' < k$.

2.2. Discretization Requirements

The discretized network model is composed of cells (each link representing a physical road may be divided into multiple cells) and nodes (each node representing a junction between some incoming and outgoing cells). 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

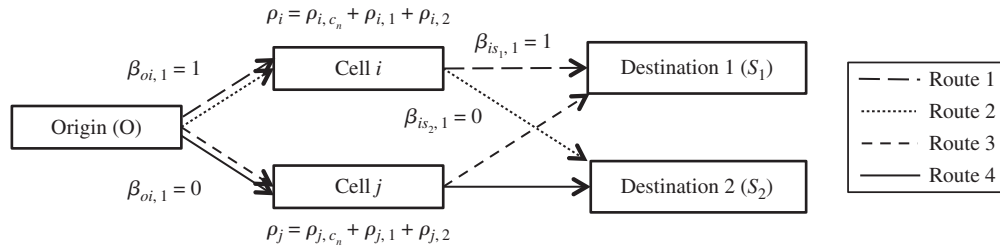
There are two types of flows that are transported in the network. Controllable flows that have OD requirements, but can be routed along any path in the network, and noncontrollable flows that have fixed paths. These flows are modeled by distributing the total flow of the network into multiple commodities, as explained below. See Figure 3 for a simple example network.

Assumption 3. Path decomposition of controllable flow. We assume that the controllable flows from each OD pair are restricted to a small predetermined subset of paths in the network.

Remark 1. The SO-DTA model presented here is designed primarily for applications where the vehicles need to be rerouted in real-time along a set of predetermined routes. Application with this requirement includes (i) real-time traffic rerouting due to an accident or other perturbation and (ii) redistributing recurring traffic on high capacity arterials and highways to avoid schools, small residential streets, and other traffic sensitive streets (a problem created by apps such as Waze). In this context, for most operational settings, it was determined that a small set of paths per OD pair was sufficient. See Appendix C for a discussion on how to find this set of paths and an alternative approach for when paths do not need to be predetermined.

Definition 8 (Non-Controllable Commodity). There is a single noncontrollable commodity c_n that represents all noncontrollable flow in the network. The paths of the flow corresponding to the noncontrollable 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

Figure 3. A Simple Network with a Single Origin and Two Destinations

Notes. The noncontrollable commodity is denoted by c_n and the compliant commodities by their corresponding route numbers. Routes 1 and 3 correspond to the OD pair (o, s_1) and routes 2 and 4 correspond to the OD pair (o, s_2) . Commodities 1, 2, and c_n flow through cell i and commodities 3, 4, and c_n flow through cell j . The routing for commodity 1 is illustrated using the split-ratios β .

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). There is no restriction on the number of OD pairs in the network.

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 OD information for all controllable flow and the aggregate path information for all noncontrollable 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 efficiently route vehicles and therefore will share their OD information. The aggregate paths of the noncontrollable 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 preprocessed to remove this contribution. This can be done, for example, by using the following process: (i) Forward simulate the system assuming a controllable flow fraction of zero; (ii) Assign the controllable flow to the shortest path corresponding to each

OD pair; (iii) Remove this flow from the network and recompute each junction split ratio to reflect the flow that has been removed.

Remark 2. 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 (Controllable Commodity and Origin-Destination Pair). 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:

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

The set $\Omega^{-1}(\{o, s\})$ contains all the commodities corresponding to the flows from origin o to destination s . The function Ω provides a mapping between commodities and their corresponding OD pairs.

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

$$u: \mathcal{CC} \times \llbracket 0, T_f \rrbracket \rightarrow [0, 1], \quad (c, k) \mapsto \gamma_c(k). \quad (10)$$

The demand allocation for commodity c at time step k is given by $\gamma_c(k)$. 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

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

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

3. Forward System

We now define the dynamics of the flows through the network in our model, which includes the boundary conditions, the rules governing the movements across junctions, and system level dynamics of the flow (i.e., the forward system).

3.1. Junction Model

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

Requirement 1 (Multicommodity First-in First-Out (FIFO) Condition). For any outgoing cell i , the distribution of its flow across the different commodities must be proportional to the ratio of vehicles of each commodity on the cell. 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

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

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

We define a multicommodity junction flow solver that assigns flows across the network in a manner that is consistent with the above requirements. The multicommodity 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 must 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., $\rho_i(k) > 0$.

See Appendix D for the specific junction model (satisfying Requirements 1–3) and solution methods that we use. Our model is related to the SDM in Garavello and Piccoli (2006) and the multicommodity CTM in Daganzo (1995).

3.2. Boundary Conditions

The boundary conditions at each origin cell 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 noncontrollable flow. Note that only the aggregate demand at each origin and junction split-ratios are required for the noncontrollable flow and that explicit OD information is not required. For the controllable flow, the explicit demand corresponding to every OD pair must be known. The nonzero terms are defined as

$$d_{i,c}(k) = \Delta t \cdot D_{i,c}(k), \quad \forall i \in \mathcal{B}, c = c_n \quad (\text{Non-controllable demand})$$

$$d_{i,c}(k) = \Delta t \cdot D_{\Omega(c)}(k) \cdot \gamma_c(k), \quad \forall i \in \mathcal{B}, \forall c \in \mathcal{C} \quad (\text{Controllable demand})$$

Because the inflow to the network is limited by the maximum flow capacity and density of the immediate downstream cell, all of the demand at a given time step might not make it into the network. An origin buffer is used to accumulate the flow that cannot enter the network to guarantee conservation of boundary flows. In the single commodity case, this model is sufficient. However, in the multicommodity 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 cell i at the ratio $d_{i,c}(k) / (\sum_{c' \in \mathcal{C}} d_{i,c'}(k))$ 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 $d_{i,c}(k)/d_i(k)$. 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 E 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 $v_i = 1/\Delta t$ 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 $x(u)$ denote 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) is 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 cell type or junction type. For notational simplicity, the equations given here are for the single buffer model. The explicit formulation is given below.

Mass conservation

$$H_{k,i,c}^1: \rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} (f_{i,c}^{\text{in}}(k-1) - f_{i,c}^{\text{out}}(k-1)),$$

$$\forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 1, T_f \rrbracket, \forall c \in \mathcal{C}, \quad (\text{H1a})$$

$$H_{k,i,c}^1: \rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} \cdot f_{i,c}^{\text{in}}(k-1),$$

$$\forall i \in \mathcal{S}, \forall k \in \llbracket 1, T_f \rrbracket, \forall c \in \mathcal{C}, \quad (\text{H1b})$$

with initial conditions

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

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

Boundary conditions

$$H_{k,i,c}^1: \rho_{i,c}(k)$$

$$= \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} (D_{\Omega(c)}(k) \cdot \gamma_c(k) - f_{i,c}^{\text{out}}(k-1)),$$

$$\forall i \in \mathcal{B}, \forall k \in \llbracket 1, T_f \rrbracket, \forall c \in \mathcal{C}, \quad (\text{H1c})$$

$$H_{k,i,c}^1: \rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} (D_{i,c}(k) - f_{i,c}^{\text{out}}(k-1)),$$

$$\forall i \in \mathcal{B}, \forall k \in \llbracket 1, T_f \rrbracket, c = c_n, \quad (\text{H1d})$$

with initial conditions

$$H_{0,i,c}^1: \rho_{i,c}(0) = \rho_{i,c}^0 + \frac{\Delta t}{L_i} \cdot D_{\Omega(c)}(0) \cdot \gamma_c(k),$$

$$\forall i \in \mathcal{B}, \forall c \in \mathcal{C}, \quad (\text{I2a})$$

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

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

$$H_{k,i}^2: \delta_i(k) = \min(F_i, v_i \rho_i(k)),$$

$$\forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket, \quad (\text{H2a})$$

$$H_{k,i}^2: \delta_i(k) = \min(F_i, \rho_i(k) L_i / (\Delta t)),$$

$$\forall i \in \mathcal{B}, \forall k \in \llbracket 0, T_f \rrbracket, \quad (\text{H2b})$$

$$H_{k,i}^3: \sigma_i(k) = \min(F_i, w_i (\rho_i^{\text{jam}} - \rho_i(k))),$$

$$\forall i \in \mathcal{A} \setminus (\mathcal{B} \cup \mathcal{S}), \forall k \in \llbracket 0, T_f \rrbracket, \quad (\text{H3a})$$

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

Junction solution

The derivation of the explicit solutions to the 1×2 , 2×1 , and 2×2 junctions listed below are given in Online Appendix F. 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 $\dot{i}_1 = i_2$ and $\dot{i}_2 = i_1$

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

Aggregate split ratios

$$H_{k,i,j,z}^4: \beta_{ij} = \frac{1}{\rho_i} \sum_{c \in \mathcal{C}} \rho_{i,c} \beta_{ij,c}, \quad \forall z \in \mathcal{J}, \forall i \in \mathcal{J}_z^{\text{in}},$$

$$\forall j \in \mathcal{J}_z^{\text{out}}, \forall k \in \llbracket 0, T_f \rrbracket. \quad (\text{H4})$$

Flow out of incoming cells by commodity

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

$$\forall z \in \mathcal{J}_{1 \times n}, \forall i \in \mathcal{J}_z^{\text{in}}, \quad (\text{H5a})$$

$$H_{k,i,c}^5: f_{i,c}^{\text{out}} = \frac{\rho_{i,c}}{\rho_i}$$

$$\cdot \begin{cases} \delta_i & \text{if } P_i (\min(\delta_1 + \delta_2, \sigma_1) - \delta_i) > \delta_i P_i, \\ \min(\delta_1 + \delta_2, \sigma_1) - \delta_i & \text{if } P_i (\min(\delta_1 + \delta_2, \sigma_1) - \delta_i) > \delta_i P_i, \\ P_i \min(\delta_1 + \delta_2, \sigma_1) & \text{otherwise,} \end{cases}$$

$$\forall z \in \mathcal{J}_{2 \times 1}, \forall i \in \mathcal{J}_z^{\text{in}}, \quad (\text{H5b})$$

$$H_{k,i,c}^5: f_{i,c}^{\text{out}} = \begin{cases} \delta_1 & \text{if } P_1/P_2 > \delta_1 / \min(\delta_2, (\sigma_1 - \beta_{11} \delta_1) / \beta_{21}, (\sigma_2 - \beta_{12} \delta_1) / \beta_{22}), \\ \min(\delta_1, (\sigma_1 - \beta_{21} \delta_2) / \beta_{11}, (\sigma_2 - \beta_{22} \delta_2) / \beta_{12}) & \\ \text{if } P_1/P_2 < \min(\delta_1, (\sigma_1 - \beta_{21} \delta_2) / \beta_{11}, (\sigma_2 - \beta_{22} \delta_2) / \beta_{12}) / \delta_2, & \\ \min(P_1 \sigma_1 / (P_1 \beta_{11} + P_2 \beta_{21}), P_1 \sigma_2 / (P_1 \beta_{12} + P_2 \beta_{22})) & \\ \text{otherwise,} & \end{cases}$$

$$\forall i \in \mathcal{J}_{2 \times 2}^{\text{in}},$$

f_2^{out} is obtained by symmetry,

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

Flow into outgoing cells by commodity

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

As shown in Figure 4, the dynamics equations has a topological ordering that allows for the following efficient forward simulation algorithm. For any given time step k , each of the three internal loops are trivially parallelizable problems.

4. Adjoint Based Optimization

Given the traffic model and the dynamics of the flow, we now describe the adjoint-based optimization framework for minimizing the total travel-time of all agents in the system.

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

$$J = \sum_{k=0}^{T-1} \sum_{i \in \mathcal{N} \setminus \mathcal{S}} \rho_i(k) \cdot L_i. \quad (17)$$

The SO-DTA-PC is a physically acceptable (see Definition 14) division of the controllable agents among the different commodities that minimizes the total travel-time (including the travel-time of the noncontrollable commodities). The solution is obtained by solving the following optimization problem:

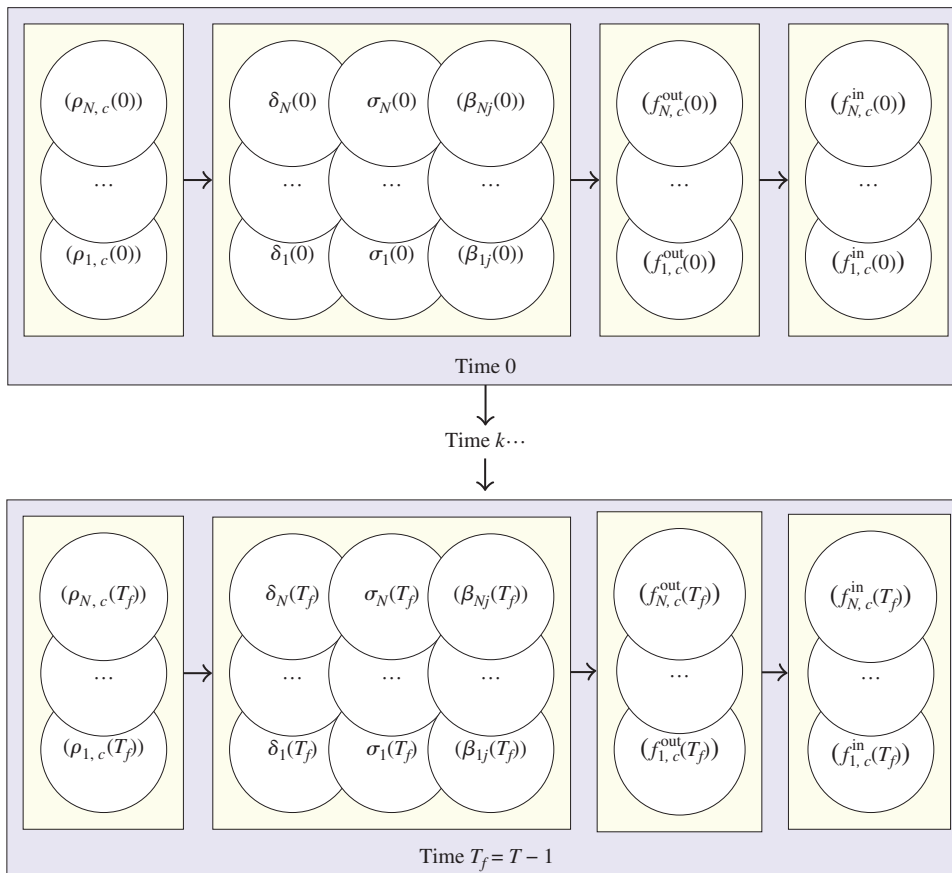
$$\begin{aligned} & \min_{u \in \mathcal{U}} J(x(u)) \\ & \text{subject to } \text{system dynamics,} \\ & \qquad \qquad \text{control constraints,} \end{aligned}$$

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

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

Note that this is a nonconvex optimization problem that might contain multiple local minima. Therefore,

Figure 4. (Color online) Dependency Diagram of the System Variables



the 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, Kahng, and Muddu 1994; Marti 2003). Furthermore, nonconvex 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 Section 4.2, to efficiently solve for the gradient of the system. The control constraints can be satisfied 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:

$$\begin{aligned} \min_{u \in \mathcal{U}} J(x, u) \\ \text{subject to } H(x, u) = 0, \end{aligned} \quad (18)$$

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

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

$$L(x, u, \lambda) = J(x, u) + \lambda^T H(x, u) \quad (19)$$

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

$$\begin{aligned} \nabla_u L(x, u, \lambda) &= \frac{\partial J}{\partial u} + \frac{\partial J}{\partial x} \frac{dx}{du} + \lambda^T \left(\frac{\partial H}{\partial u} + \frac{\partial H}{\partial x} \frac{dx}{du} \right) \\ &= \frac{\partial J}{\partial u} + \lambda^T \frac{\partial H}{\partial u} + \left(\frac{\partial J}{\partial x} + \lambda^T \frac{\partial H}{\partial x} \right) \frac{dx}{du}. \end{aligned} \quad (20)$$

In particular, if λ satisfies the adjoint equation

$$\frac{\partial J}{\partial x} + \lambda^T \frac{\partial H}{\partial x} = 0, \quad (21)$$

then the gradient is

$$\nabla_u L(x, u) = \frac{\partial J}{\partial u} + \lambda^T \frac{\partial H}{\partial u}. \quad (22)$$

The solution for λ exists and is unique if $\partial H/\partial x$ is not singular, which is the case in our forward system, as explained in Section 4.3.

4.3. Applying the Adjoint Method

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

All of the diagonal terms of $\partial H/\partial x$ are nonzero (since equal to 1 or -1 depending on the way we rewrite H_v). As seen in the dependency chain shown in Figure 4, the nonzero derivative terms of H_v depend only on variables that have a smaller index in x . This means that $\partial H/\partial x$ is lower triangular with nonzero terms on the diagonal and is thus nonsingular. Therefore, we can apply the adjoint method to compute the gradient of this system.

Reduced state space. The forward system dynamics described in Section 3.3 had many state variables. However, the only required state variables of the system are the partial densities $\rho_{i,c}(k)$. All of the other 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

$$\begin{aligned} x &= \begin{pmatrix} ((\rho_{i,c}(k))_{c \in \mathcal{C}})_{i \in \mathcal{A}} \\ ((f_{i,c}^{\text{out}}(k))_{c \in \mathcal{C}})_{i \in \mathcal{A}} \\ ((f_{i,c}^{\text{in}}(k))_{c \in \mathcal{C}})_{i \in \mathcal{A}} \end{pmatrix}_{k \in \llbracket 0, T_f \rrbracket}, \\ H &= \begin{pmatrix} ((H_{k,i,c}^1)_{c \in \mathcal{C}})_{i \in \mathcal{A}} \\ ((H_{k,i,c}^5)_{c \in \mathcal{C}})_{i \in \mathcal{A}} \\ ((H_{k,i,c}^6)_{c \in \mathcal{C}})_{i \in \mathcal{A}} \end{pmatrix}_{k \in \llbracket 0, T_f \rrbracket}. \end{aligned}$$

Computational complexity. Let n be the dimension of the state vector $x \in R^n$, m be the dimension of the control vector $u \in R^m$, 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 $O(n^2m)$ time

$$\nabla_u J(x, u) = \frac{\partial J}{\partial x} \cdot \nabla_u x + \frac{\partial J}{\partial u}. \quad (23)$$

Computing $\nabla_u J$ requires solving the system $H(x, u) = 0 \Rightarrow (\partial H/\partial x)(dx/du) + \partial H/\partial u = 0$, which is equivalent to solving m different $n \times n$ linear systems and takes $O(n^2m)$ time. The final step of multiplying $(\partial J/\partial x)(dx/du)$ and adding $\partial J/\partial u$ takes $O(nm)$ time, but is dominated by the time to compute 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 (21) only takes $O(n^2)$ time since it only requires solving one $n \times n$ linear system. Multiplying $\lambda^T(\partial H/\partial u)$ and adding $\partial J/\partial u$ 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 previously, $\partial H/\partial x$ is a lower triangular matrix and therefore we can compute the solution to Equation (21) using backwards substitution. We exploit the fact that the matrix $\partial H/\partial x$ 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 (21) can be solved in $O(n)$ time. If the maximum in-degree of the network is d_{in} , the maximum column cardinality is $2 + |\mathcal{C}| \cdot (1 + d_{in})$, as will be clear in Section 4.4 from Equation (27). Assuming that d_{in} is a small constant, the multiplication step in Equation (22) 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

$$\frac{\partial J}{\partial x} + \lambda^T \frac{\partial H}{\partial x} = 0, \quad (24)$$

$$\Rightarrow \frac{\partial J}{\partial x} + \sum_{x' \in x} \lambda_{x'} \frac{\partial H_{x'}}{\partial x} = 0, \quad (25)$$

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 forward system equations where x' appears and consider all non-null $\partial H_{x'}/\partial x$ terms. In particular we write the equations such that $\partial H_{x'}/\partial x' = -1$. Note that this can be done because the Godunov scheme provides an explicit expression for the forward system constraints.

Computing $\partial J/\partial x$

$$\frac{\partial J}{\partial \rho_{i,c}(k)} = \begin{cases} L_i & \forall c \in \mathcal{C}, \forall i \in \mathcal{A} \setminus \mathcal{S}, \forall k \in \llbracket 0, T_f \rrbracket, \\ 0 & \text{otherwise.} \end{cases} \quad (26)$$

Computing $\lambda^T(\partial H/\partial x)$

$$\begin{aligned} \frac{\partial H}{\partial \rho_{i,c}(k)} &: \sum_{x' \in x} \lambda_{x'} \frac{\partial H_{x'}}{\partial \rho_{i,c}(k)} \\ &= \lambda_{\rho_{i,c}(k)} \frac{\partial H_{\rho_{i,c}(k)}}{\partial \rho_{i,c}(k)} + \lambda_{\rho_{i,c}(k+1)} \frac{\partial H_{\rho_{i,c}(k+1)}}{\partial \rho_{i,c}(k)} \\ &\quad + \sum_{c' \in \mathcal{C} \setminus \mathcal{C}} \left(\lambda_{f_{i,c'}^{\text{out}}(k)} \frac{\partial H_{f_{i,c'}^{\text{out}}(k)}}{\partial \rho_{i,c}(k)} \right. \\ &\quad \left. + \lambda_{f_{x,c'}^{\text{out}}(k)} \sum_{x:(x,i) \in \mathcal{A}} \frac{\partial H_{f_{x,c'}^{\text{out}}(k)}}{\partial \rho_{i,c}(k)} \right), \quad (27) \end{aligned}$$

$$\begin{aligned} \frac{\partial H}{\partial f_{i,c}^{\text{out}}(k)} &: \sum_{x' \in x} \lambda_{x'} \frac{\partial H_{x'}}{\partial f_{i,c}^{\text{out}}(k)} \\ &= \lambda_{\rho_{i,c}(k+1)} \frac{\partial H_{\rho_{i,c}(k+1)}}{\partial f_{i,c}^{\text{out}}(k)} + \lambda_{f_{i,c}^{\text{out}}(k)} \frac{\partial H_{f_{i,c}^{\text{out}}(k)}}{\partial f_{i,c}^{\text{out}}(k)} \\ &\quad + \sum_{j:j \in (i,j)} \lambda_{f_{j,c}^{\text{in}}(k)} \frac{\partial H_{f_{j,c}^{\text{in}}(k)}}{\partial f_{i,c}^{\text{out}}(k)}, \quad (28) \end{aligned}$$

$$\begin{aligned} \frac{\partial H}{\partial f_{i,c}^{\text{in}}(k)} &: \sum_{x' \in x} \lambda_{x'} \frac{\partial H_{x'}}{\partial f_{i,c}^{\text{in}}(k)} \\ &= \lambda_{\rho_{i,c}(k+1)} \frac{\partial H_{\rho_{i,c}(k+1)}}{\partial f_{i,c}^{\text{in}}(k)} + \lambda_{f_{i,c}^{\text{in}}(k)} \frac{\partial H_{f_{i,c}^{\text{in}}(k)}}{\partial f_{i,c}^{\text{in}}(k)}. \quad (29) \end{aligned}$$

Section 4.5 shows how to compute all of the individual partial derivatives that are required in the above equations. Once they are computed, we simply plug them into the above equations and solve the system via backwards substitution since $\partial H/\partial x$ is lower triangular.

4.5. Partial Derivatives

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

- Partial derivatives of the cost function with respect to the control variables ($\partial J/\partial u$) from Equation (17)

$$\frac{\partial J}{\partial \gamma_c(k)} = 0. \quad (30)$$

- Partial derivatives of the cost function with respect to the state variables ($\partial J/\partial x$) from Equation (17)

$$\frac{\partial J}{\partial \rho_{i,c}(k)} = \begin{cases} L_i & \text{if } c \in \mathcal{C}, i \in \mathcal{A} \setminus \mathcal{S}, k \in \llbracket 0, T_f \rrbracket, \\ 0 & \text{otherwise.} \end{cases} \quad (31)$$

- Partial derivatives of the constraints with respect to the control variables ($\partial H/\partial u$) from Equation (H1c)

$$\frac{\partial \rho_{i,c}(k)}{\partial \gamma_c(k)} = \begin{cases} \frac{\Delta t}{L_i} \cdot D_{\Omega(c)}(k) & \text{if } c \in \mathcal{C} \setminus \mathcal{C}, i \in \mathcal{B}, k \in \llbracket 0, T_f \rrbracket, \\ 0 & \text{otherwise.} \end{cases} \quad (32)$$

Finally, we need to compute the partial derivatives of the constraints with respect to the state variables $\partial H/\partial x$, which is much more algebraically involved and is described in Online Appendix G.

5. Numerical Results

To illustrate the effectiveness of our framework for computing the SO-DTA-PC, we have implemented the algorithm and tested it on synthetic and practical traffic rerouting scenarios using experimental field data. Our implementation uses the discrete adjoint method to compute the gradient and a projection step to keep

the solution in the physically feasible control set as explained in Section 4.1. We use the Rprop algorithm (Riedmiller and Braun 1992) as our gradient descent technique. All of the experiments were run on a 1.8 GHz Intel Core i5 dual-core processor with 8 GB RAM. The performance cost C of each scenario is measured using the total travel time of all of the vehicles passing through the network

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

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 5.
2. A subsection of Interstate 210 with a parallel arterial route, as depicted in Figure 7.

5.1. Synthetic Network

The synthetic network is a simple 10 cell network adapted from the example used in Ziliaskopoulos (2000). We have slightly modified the original network to increase the capacity of cells 8 and 9 such that they can accommodate flow from routes 2 and 3, and changed some of the other parameters to satisfy the CFL conditions in Section 2.2. The network 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 5 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 of the first three time steps is, respectively, 8, 16, and 8 vehicles. The network

Table 1. Optimal Allocation of Demand Across Routes

Time step	1	2	3
(a) Normal operation			
Route 1	0.75	0.5	0.5
Route 2	0	0	0
Route 3	0.25	0.5	0.5
(b) Incident local minimum			
Route 1	0.25	0.417	0.417
Route 2	0	0	0
Route 3	0.75	0.583	0.583
(c) Incident global minimum			
Route 1	0	0.25	0.25
Route 2	0.25	0	0
Route 3	0.75	0.75	0.75

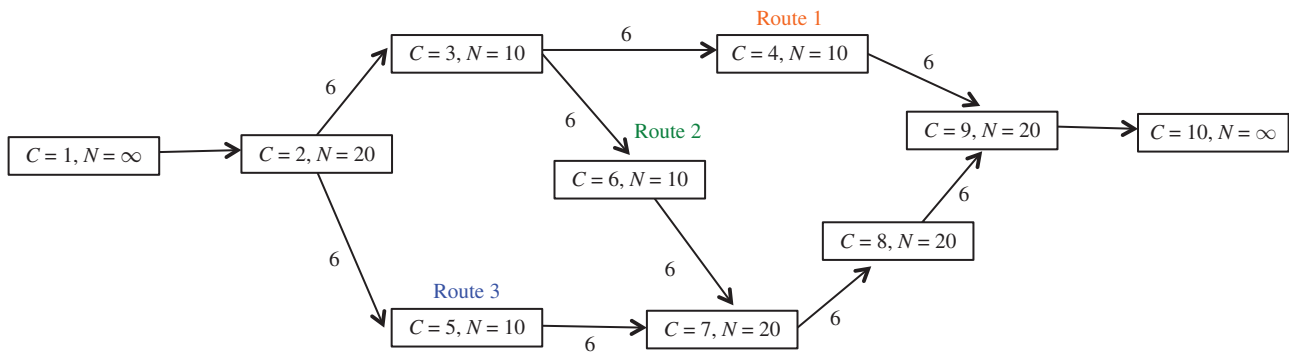
is simulated for 10 time steps, which gives enough time for all of the entering flow to exit the network.

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 controllable. Table 1(a) shows the optimal route allocation for the origin demands at each time step with nonzero 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.

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

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 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

Figure 5. (Color online) The Synthetic Network



Notes. 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.
 Source. Adapted from Ziliaskopoulos (2000).

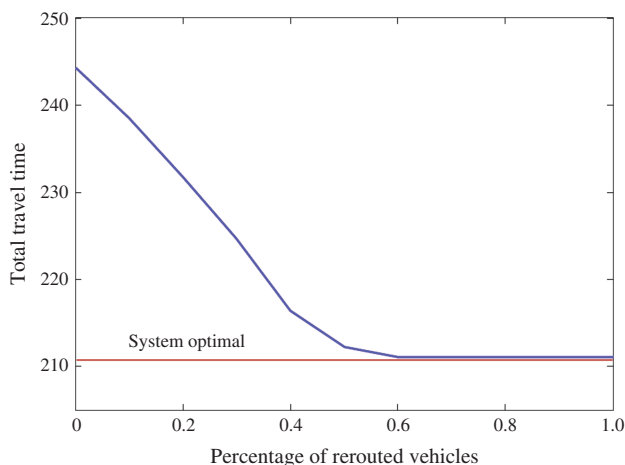
Table 2. Capacity Reduction Due to an Incident

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 1(b). It turns out that this solution is actually a local minimum in the system due to the FIFO condition for vehicles departing cell 3. Whenever there is some nonzero flow for route 1 and the capacity between cells 3 and 4 is zero, the flow of vehicles that take route 2 is also restricted to zero. This causes a nonconvexity 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, Kahng, and Muddu 1994; Marti 2003) and the efficient gradient computation obtained via the adjoint method can be combined with nonconvex optimization techniques such as interior point methods (Wachter and Biegler 2005).

Partial control. In many situations, it might not be possible to reroute all of the vehicles in the system. Therefore, we also analyze the behavior of the system when only some fraction of the vehicles are rerouted. We consider the example above with a capacity reduction due to an incident, but in this case, assume that only a certain fraction of the flow can be controlled. The remaining (noncontrollable) flow is expected to continue to follow the flow distribution from before the incident (i.e., the distribution given by Table 1(a)). Figure 6 shows how the total travel time changes with

Figure 6. (Color online) The Change in Total Travel Time vs. the Percentage of Vehicles That Can Be Rerouted



Note. All performance measures are with respect to the local minimum found by the optimizer.

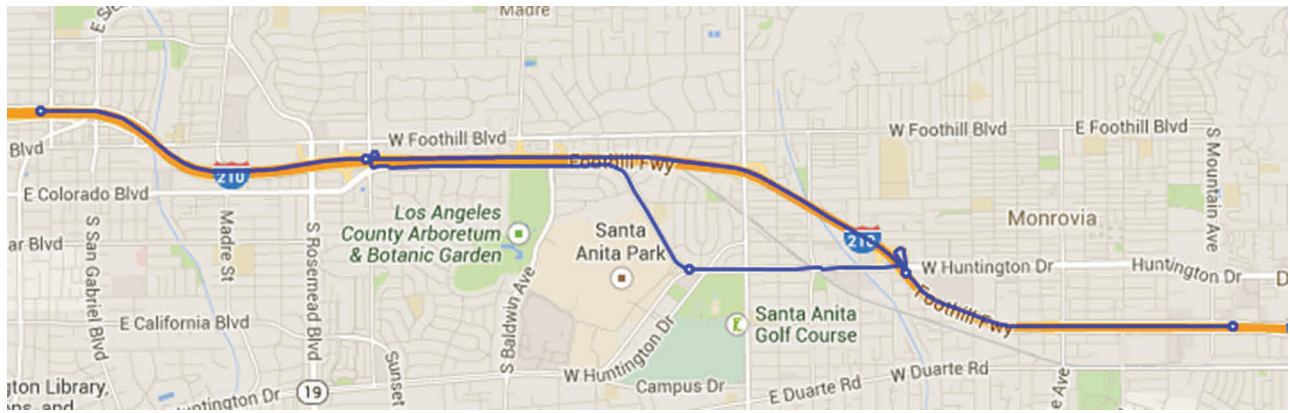
the percentage of vehicles that can be rerouted. In this example, we see that the system optimal (local minimum) can be achieved by controlling only 60% of the vehicles. However, in practice, it is possible that the noncontrollable passengers might alter their route choice distribution after the capacity drop, particularly if they are aware of the control strategy being used on the controllable travelers. In this work, we ignore the potential endogenous route change response by the noncontrollable users as explained in Section 1. Accounting for the route choice response by the noncontrollable users to the control strategy leads to a Stackelberg game and is a computationally hard problem to solve, as also alluded to in Section 1. This is an extremely interesting area of future research and an open problem.

5.2. Interstate 210 Sub-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 7. 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 the 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 were 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 8(a).

Capacity drop during the 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 8(b). The freeway capacity at segment five is 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 8 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 Figure 8 shows, rerouting the excess flow to the parallel arterial eliminates the bottleneck during the incident and improves the throughput of the freeway corridor.

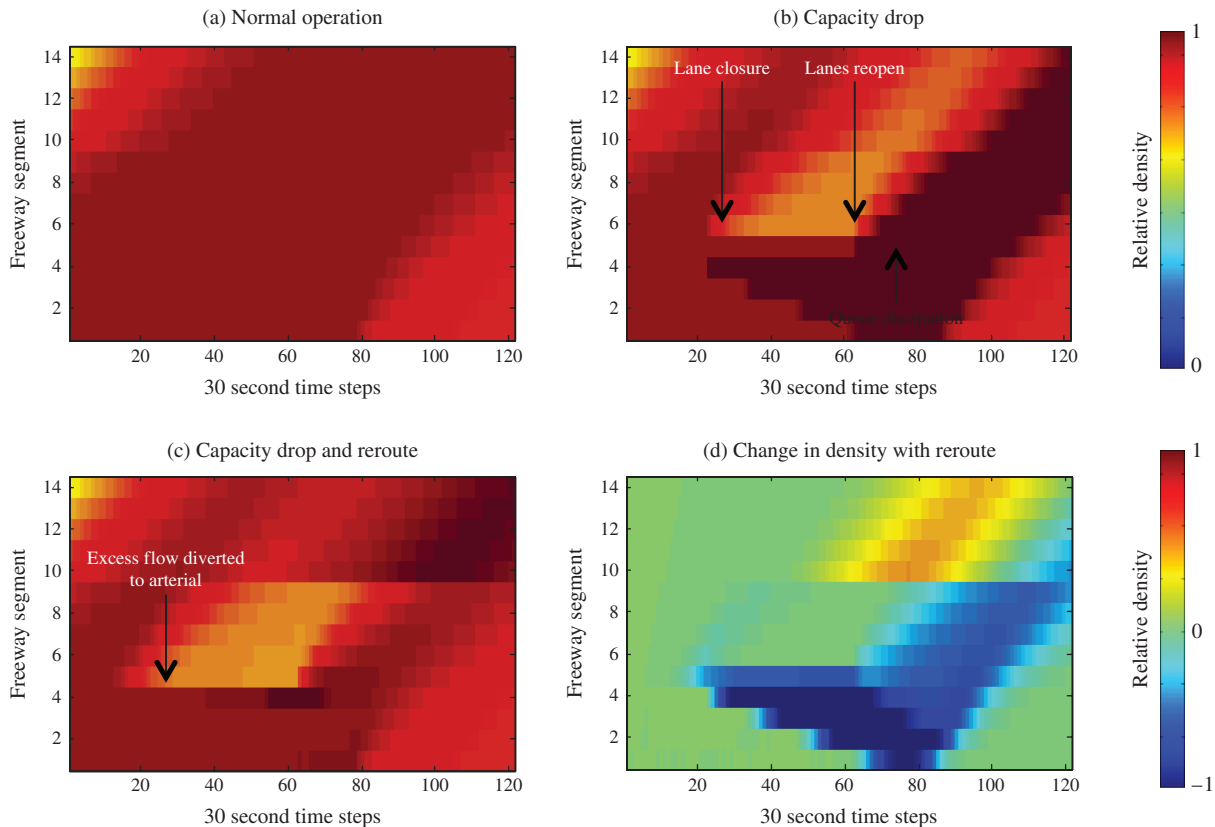
Figure 7. (Color online) The Interstate 210 Sub-Network



Partial capacity utilization. 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

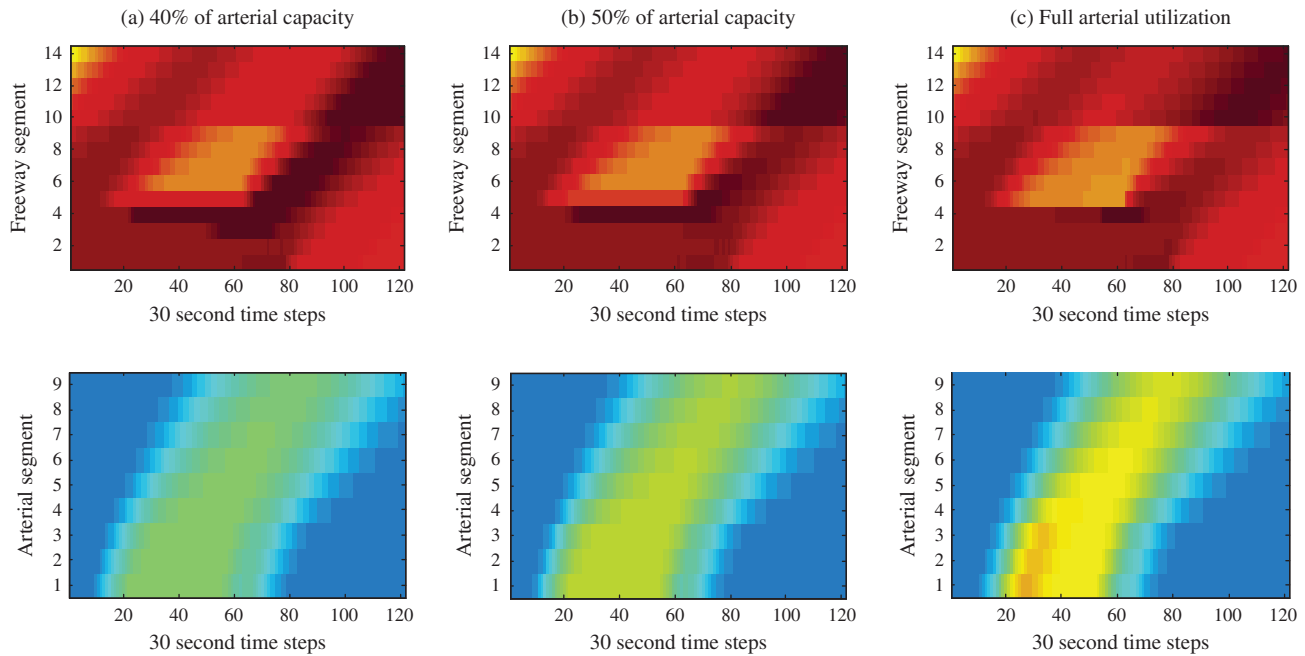
of the capacity of the parallel arterial to reroute flows. Figure 9 shows the density evolution when the arterial capacity allocated for rerouting freeway traffic is limited to 40% and 50% compared to full arterial utilization. The amount of local traffic that enters the parallel arterial can be controlled via the traffic signal controls along the arterial.

Figure 8. (Color online) The Density Evolution Along the 14 Freeway Cells With (a) No Incident, (b) A Two-Lane Capacity Drop from Minutes 10–30 at Cell 5, (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



Note. Panels (a)–(c) correspond to the 0 to 1 relative density color map and panel (d) corresponds to the -1 to 1 relative density color map.

Figure 9. (Color online) A Comparison of the Density Evolution for Different Rerouting Capacities on the Parallel Arterial Route: (a) Only 40% of the Arterial Capacity Can Be Used for Rerouting Freeway Vehicles, (b) Only 50% of the Arterial Capacity Can Be Used for Rerouting Freeway Vehicles, and (c) The Entire Arterial Capacity Is Used for Rerouting Freeway Vehicles (i.e., the Parallel Arterial Temporarily Closed for Other Traffic)



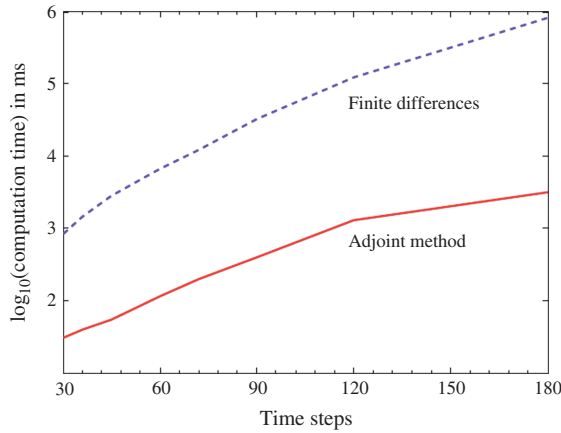
Note. All panels correspond to the 0 to 1 relative density color map shown in Figure 8.

Partial control. As in the case of the synthetic network, the percentage of controllable vehicles can also be regulated in this example. Practically speaking, such a situation may arise due to only a fraction of the vehicles on the roadway being equipped with the technology to be rerouted by a central controlling authority or due to the incentives (resources) required to get drivers to reroute from their original route. The heatmaps that arise from full control and partial control scenarios will look similar to the results in Figure 9, since both restrictions (on arterial capacity and the fraction of controllable vehicles) essentially limit the number of vehicles that can be rerouted to the parallel arterial.

Adjoint method—practical benefits. 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 method by perturbing each variable and measuring the response of the system. See Morton and Mayers (2005) for a detailed analysis of finite difference methods. 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. By 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 fixed OD pair will only typically have a small number of reasonable paths. Figure 10 shows the time taken for one gradient computation as a function of the 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. Therefore, one of the major practical contributions of this framework is the ability to efficiently compute the system gradient in multicommodity SO-DTA problems (including the cases of partial and full control). The adjoint-based solution technique provides a computationally tractable method for real-time route control in multicommodity traffic networks.

Figure 10. (Color online) The Base 10 Logarithm (\log_{10}) of the Gradient Computation Time for Solving the I-210 Network vs. the Number of Time Steps in the Problem



Notes. 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.

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 physical 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 noncontrollable (selfish) agents. We have used a Godunov discretization of the Lighthill–Williams–Richards (LWR) partial differential equation with a triangular flux function (similar to the Cell Transmission Model) and a corresponding multicommodity junction solver for the dynamics of the system, but the proposed framework can also be used with other dynamics models. 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 vehicle rerouting strategy in response to a capacity loss in a network, and show the congestion reductions that can be achieved. Numerical results are presented for a test network and Interstate 210 in Southern California. The major limitation of this work is the assumption that noncontrollable vehicles have a fixed path choice. Relaxing this assumption, as discussed in the article, requires posing the problem as a Stakelberg game and substantially increases the computational complexity of the problem. Finding efficient approximation techniques to solve this more general problem is an active area of future research and an open problem.

Appendix A. Notation

Constants

$\Delta t, \Delta x$ Time and space discretization;

k, T, T_f Time index, number of time steps, and final time step ($k \in [0, T_f = T - 1]$);
 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 cells to junction z ;
 $\mathcal{J}_z^{\text{out}}$ Outgoing cells 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 noncontrollable agents of commodity $c = c_n$ 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 .

Appendix B. CFL Conditions

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

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). $\forall i \in \mathcal{A} \setminus \mathcal{B}, w_i \leq L_i / \Delta t$.

Each cell cannot have a density greater than ρ_i^{jam} . Requirement 5 ensures that this condition is satisfied. While the requirement that $v_i \leq L_i / \Delta t$ 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. When the inflow is supply constrained, the resulting density at the next time step must always be smaller than ρ_i^{jam}

$$\begin{aligned} \rho_i(k) + \frac{\Delta t}{L_i} w_i (\rho_i^{\text{jam}} - \rho_i(k)) &< \rho_i^{\text{jam}}; \\ \rho_i(k) \left(1 - w_i \frac{\Delta t}{L_i}\right) &< \rho_i^{\text{jam}} \left(1 - w_i \frac{\Delta t}{L_i}\right), \end{aligned}$$

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

Requirement 6 (Finite Time Density Discharge). $\forall i \in \mathcal{A}$, $v_i \geq L_i/\Delta t$.

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 $v_i < L_i/\Delta t$, we can have an exponential decrease of the density in some cells when they should be emptied in only one step. Taking the case of a cell without inflow, we have $\rho_i(k+1) = \rho_i(k) - (\Delta t/L_i)\rho_i(k)v_i$, which gives $\rho_i(k+t) = \rho_i(k)(1 - \Delta t v_i/L_i)^t$. This is not an acceptable physical solution and thus should be excluded.

Remark 3. Satisfying requirements 4 and 6 implies imposing $v_i = L_i/\Delta t$. 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 cell 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 bad in practice, since the number of vehicles stuck in a cell decreases exponentially with time.

Appendix C. Routing Over Predetermined Paths

The model presented in this work assumes that each controllable OD demand is restricted to a small subset of predetermined paths in the network. As stated previously (see Assumption 3), this assumption is driven by the fact that the primary applications of this work required real-time control and the ability to prespecify paths.

In the corresponding operational context, the set of routes is selected by a traffic engineer or planner based on their expertise and specific considerations such as avoiding sensitive areas (schools, residential streets, etc.). However, if this process needs to be automated, the following algorithmic approaches can be considered.

One approach is to use an alternative paths algorithm. These algorithms, which are used by companies such as Google/TomTom/Apple to present alternate routes in navigation systems, are more sophisticated than simpler k -shortest

paths algorithms that typically provide many very similar routes with slight deviations. As described in Bast et al. (2016, p. 26), alternate paths algorithms aim to find paths that are “short, smooth, and significantly different (from the shortest path and other alternatives).” See Section 3.2 in Bast et al. (2016) for a short discussion on the topic and links to other articles for further details.

Another heuristic approach is to solve the static UE problem for the maximum corresponding flows and pick the paths with nonzero flows. This approach might however ignore some of the paths that appear in an SO solution, so it may be appropriate to consider increasing the maximum flow to activate more candidate paths.

Furthermore, even in the context of solving the SO-DTA problem for all possible paths, most road networks will only induce vehicle flows (for a given OD pair) on a small subset of paths (under an appropriate network aggregation). Small deviations from these aggregated paths on low capacity roads may not significantly compromise the quality of the solution. Therefore, limiting the flows to a small set of paths might not result in a major loss of efficiency. If the network is aggregated, a secondary optimization problem can be solved to distribute the flows from the aggregated path to its components in the original network.

Finally, if it is necessary to solve the problem in full generality, an alternative approach (that does not limit the number of paths that are used) is to change the control variables from the path choice at each origin to destination specific split ratios at each junction. This can still be achieved within the framework described in this article. In a junction split ration based formulation, the number of control variables scale as $O(D \times J)$ with D being the number of destinations and J is the number of junctions, as opposed to $O(P \times K)$ with P being the number of OD pairs and K being the number of paths per OD pair in the current formulation. Which formulation has more control variables will depend on the specifics of the network being considered. In terms of the adjoint formulation, the resulting partial derivatives will be more complex as the control variables will now explicitly appear in each junction problem. While this makes their analytical derivation more tedious, it will not shorten the time taken to compute each gradient. It is likely however that a junction based formulation might shorten the convergence time of the gradient descent. Implementing and comparing a junction based formulation is a natural extension of this work.

Appendix D. Junction Solvers

In this section we describe the junction solvers for the diverge, merger, and diverge/merge junctions that we consider in our model. The underlying optimization problem and an informal proof of existence and uniqueness of the solution is given. Explicit solutions to the junction problems are given in Online Appendix F.

D.1. Diverge Solver ($1 \times m$)

We consider a diverging junction z with one incoming cell i and m outgoing cells. 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 maximize the flow across the junction subject to the maximum flow constraints

$$\begin{aligned} & \max f_i^{\text{out}}(k) \\ & \text{subject to } 0 \leq f_j^{\text{in}}(k) \leq \sigma_j(k), \quad \forall j \in \mathcal{F}_z^{\text{out}}, \\ & \quad 0 \leq f_i^{\text{out}}(k) \leq \delta_i(k). \end{aligned} \quad (\text{D.1})$$

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

$$\begin{aligned} f_j^{\text{in}}(k) &= \sum_{c \in \mathcal{C}} \beta_{ij,c}(k) \cdot f_{j,c}^{\text{out}}(k) \quad [\text{mass conservation}] \\ &= f_i^{\text{out}}(k) \sum_{c \in \mathcal{C}} \frac{\rho_{i,c}(k)}{\rho_i(k)} \cdot \beta_{ij,c}(k) \quad [\text{by the FIFO constraint}] \\ &= f_i^{\text{out}}(k) \cdot \beta_{ij}(k) \quad [\text{by definition of the aggregate} \\ & \quad \text{split ratios}] \end{aligned} \quad (\text{D.2})$$

This gives us a trivial maximization problem that implies the following equality:

$$f_i^{\text{out}}(k) = \min \left(\left\{ \frac{\sigma_j(k)}{\beta_{ij}(k)}, \forall j \in \mathcal{F}_z^{\text{out}} \mid \beta_{ij}(k) > 0 \right\}, \delta_i(k) \right). \quad (\text{D.3})$$

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

Existence and uniqueness of the solution. A nonzero 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 nonzero and none of the outgoing cells with positive demand ($\beta_{ij}(k) > 0$) have nonzero supply, a nonzero solution exists. Because the solution to the maximum junction flow is given by Equation (D.3) and the outflows are uniquely determined by the split ratios, the solution is unique.

D.2. Merge Solver ($n \times 1$)

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

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

$$f_i^{\text{out}}(k) = \delta_i(k), \quad \forall i \in \mathcal{F}_z^{\text{in}}. \quad (\text{D.4})$$

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:

$$\begin{aligned} & \min_{t, \{f_i^{\text{out}}(k)\}_{i \in \mathcal{F}_z^{\text{in}}}} \sum_{i \in \mathcal{F}_z^{\text{in}}} (f_i^{\text{out}}(k) - t \cdot p_{ij})^2 \\ & \text{subject to } \sum_{i \in \mathcal{F}_z^{\text{in}}} f_i^{\text{out}}(k) = \sigma_j(k), \\ & \quad 0 \leq f_i^{\text{out}}(k) \leq \delta_i(k), \quad \forall i \in \mathcal{F}_z^{\text{in}}. \end{aligned} \quad (\text{D.5})$$

The total outflow $f_i^{\text{out}}(k)$ for each incoming cell i is then divided among the commodities according to the FIFO law: $f_{i,c}^{\text{out}}(k) = (\rho_{i,c}(k)/\rho_i(k))f_i^{\text{out}}(k)$. See Figure D.1 for an 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 chosen. The solution violates the priority rule only in the case where the demand for one or more of the incoming cells is less than its flow-maximizing allocation based on the priority vector. In other words, the priority rule is only violated when an incoming cell does not 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 multiobjective optimization problem to come up with a physically meaningful set of flows through the junction.

Existence and uniqueness of the solution. In the demand-constrained case, existence and uniqueness are trivial. In the supply-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: $\sum_{i \in \mathcal{F}_z^{\text{in}}} f_i^{\text{out}}(k) \leq \sigma_j(k)$) and demand constraints (an n -dimensional hyperrectangle: $f_i^{\text{out}}(k) \leq \delta_i(k), \forall i \in \mathcal{F}_z^{\text{in}}$). 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 nondegenerate (zero supply or demand) cases. The boundary of the supply constraint hyperplane intersects each coordinate axis at $x_i(k) = \sigma_i(k)$ and cannot 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.

D.3. Merge-Diverge Solver ($2 \times m$)

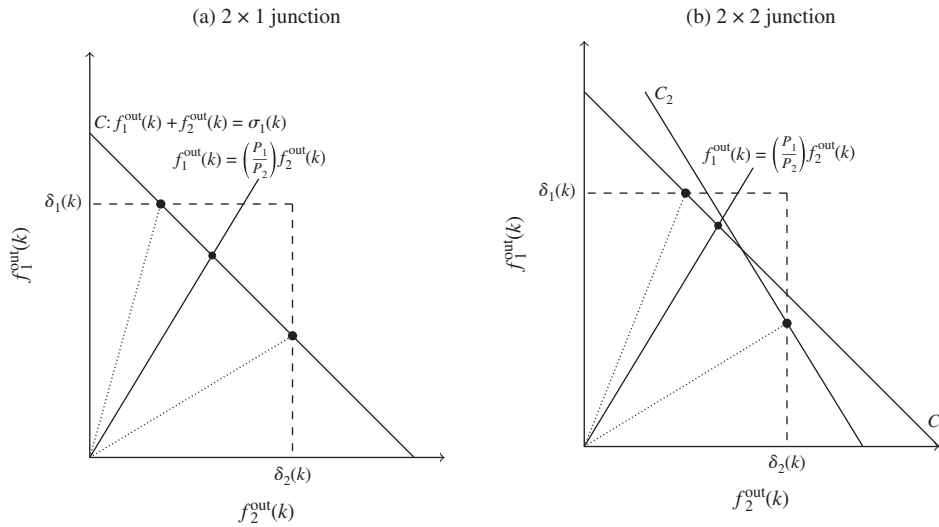
We consider a junction with 2 incoming cells and m outgoing cells. Our analysis is limited to merge-diverge junctions of no more than two incoming cells because our model does not prescribe a unique solution when the number of incoming cells is greater than two. Thus, our model can only be used with $1 \times m$, $n \times 1$, and $2 \times m$ junctions. To our knowledge, finding an explicit junction solver for general junctions that converges to the continuous solution in the limit when the discretization goes to zero is still an open problem. See Garavello and Piccoli (2009) and Monache, Goatin, and Piccoli (2016) for some recent results.

Assumption 5. *The priority vectors P_j for each outgoing cell 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 cell the vehicles will enter.*

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

Let $\mathcal{F}_z^{\text{in}}$ and $\mathcal{F}_z^{\text{out}}$ be the sets of incoming and outgoing cells at the junction. If the problem is demand-constrained

Figure D.1. An Illustration of the Solutions to Merging Junctions



Notes. A similar illustration appears in Daganzo (1995). The dashed lines denote the demand constraints imposed by the density on the incoming cells. The solid lines (C) denote the supply constraints imposed by the density in the outgoing cells. 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).

(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

$$f_i^{\text{out}}(k) = \delta_i(k), \quad \forall i \in \mathcal{J}_z^{\text{in}}. \quad (\text{D.6})$$

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

$$\begin{aligned} \min_{t, \{f_i^{\text{out}}(k)\}_{i \in \mathcal{J}_z^{\text{in}}}} \quad & \sum_{i \in \mathcal{J}_z^{\text{in}}} (f_i^{\text{out}}(k) - t \cdot P_i)^2 \\ \text{subject to} \quad & \sum_{i \in \mathcal{J}_z^{\text{in}}} \beta_{ij}(k) f_i^{\text{out}}(k) \leq \sigma_j(k), \quad \forall j \in \mathcal{J}_z^{\text{out}}, \\ & \max_j \left(\sum_{i \in \mathcal{J}_z^{\text{in}}} \beta_{ij}(k) f_i^{\text{out}}(k) - \sigma_j(k) \right) = 0, \quad \forall j \in \mathcal{J}_z^{\text{out}}, \\ & f_i^{\text{out}}(k) \leq \delta_i(k), \quad \forall i \in \mathcal{J}_z^{\text{in}}. \end{aligned} \quad (\text{D.7})$$

The total outflow $f_i^{\text{out}}(k)$ for each incoming cell i is then divided among the commodities according to the FIFO law: $f_{i,c}^{\text{out}}(k) = (\rho_{i,c}(k) / \rho_i(k)) f_i^{\text{out}}(k)$. The commodity flows are split among the outgoing cells 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 D.1 for an illustration of the solution to a 2×2 junction.

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 (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 nonempty, which is the case if the supply/demand constraints are greater than zero. This proves the existence of a solution in all nondegenerate (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 $x_i(k) = \min_j(\sigma_j(k) / \beta_{ij}(k))$ and therefore cannot 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.

Appendix E. FIFO Condition at the Origin

Note that the FIFO condition is violated in its strict sense even within the network. The flow propagation model assumes that all of 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 4 (Cell-Level Multicommodity FIFO Condition). The strict multicommodity FIFO condition is not satisfied in most traffic flow models, but this is considered 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 an origin buffer at the boundary of the network can be arbitrarily large, and thus, there is no bound on how badly the multicommodity 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 origin 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. 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 origin. 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 $l^{\max}/(\Delta t \cdot F_b)$ buffers per origin, 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 origin node i at time step k and $l_i^b(k)$ its sum over all of the commodities.

The buffers are updated as follows:

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

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

$$l_{i,c}^1(k+1) = l_{i,c}^1(k) - f_{i,c}^{\text{in}}(k), \quad \forall c \in \mathcal{C}. \quad (\text{E.2})$$

- 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**

$\Delta l^b = \min(L - l_i^b(k+1), l_i^{b+1}(k+1))$
 ▶ Maximum flow that can enter buffer b

$$l_{i,c}^b(k+1) = l_{i,c}^b(k+1) + \frac{l_{i,c}^{b+1}(k+1)}{l_i^{b+1}(k+1)} \Delta l^b, \quad \forall c \in \mathcal{C}$$

▶ Per-commodity flow entering buffer b

$$l_{i,c}^{b+1}(k+1) = l_{i,c}^{b+1}(k+1) - \frac{l_{i,c}^{b+1}(k+1)}{l_i^{b+1}(k+1)} \Delta l^b, \quad \forall c \in \mathcal{C}$$

▶ Per-commodity flow leaving buffer $b + 1$

end for

$\Delta d = d_i(k)$ ▶ Total demand at time step $k + 1$
 $b = B$

▶ Allocate the demand to last buffer and create new buffers if needed

while $\Delta d > 0$ **do**

$\Delta l^b = \min(L - l_i^b(k+1), \Delta d)$
 ▶ Maximum flow that can enter buffer b

$$l_{i,c}^b(k+1) = l_{i,c}^b(k+1) + \frac{d_{i,c}(k+1)}{d_i(k+1)} \Delta l^b, \quad \forall c \in \mathcal{C}$$

▶ 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. A mathematically rigorous spillback model in the continuous time setting is presented in Han, Piccoli, and Friesz (2016).

References

- Aswani A, Tomlin C (2011) Game-theoretic routing of GPS-assisted vehicles for energy efficiency. *Amer. Control Conf. (ACC)*, 3375–3380.
- Bast H, Delling D, Goldberg A, Müller-Hannemann M, Pajor T, Sanders P, Wagner D, Werneck RF (2016) Route planning in transportation networks. Kliemann L, Sanders P, eds. *Algorithm Engineering*, Lecture Notes Comput. Sci., Vol. 9220 (Springer International Publishing, Cham, Switzerland), 19–80.
- Bayen AM, Raffard RL, Tomlin CJ (2006) Adjoint-based control of a new Eulerian network model of air traffic flow. *IEEE Trans. Control Systems Tech.* 14(5):804–818.
- Beckman M, McGuire CB, Winsten CB (1956) *Studies in the Economics of Transportation* (Yale University Press, New Haven, CT).
- Bertsekas DP (1999) *Nonlinear Programming* (Athena Scientific, Belmont, MA).
- Boese KD, Kahng AB, Muddu S (1994) A new adaptive multi-start technique for combinatorial global optimizations. *Oper. Res. Lett.* 16(2):101–113.
- Braess D (1968) Über ein Paradoxon aus der Verkehrsplanung. *Math. Methods Oper. Res.* 12(1):258–268.
- Carey M (1992) Nonconvexity of the dynamic traffic assignment problem. *Transportation Res. Part B: Methodological* 26(2):127–133.
- Carey M, Watling D (2012) Dynamic traffic assignment approximating the kinematic wave model: System optimum, marginal costs, externalities and tolls. *Transportation Res. Part B: Methodological* 46(5):634–648.
- Chen C, Petty K, Skabardonis A, Varaiya P, Jia Z (2001) Freeway performance measurement system: Mining loop detector data. *Transportation Res. Record: J. Transportation Res. Board* 1748(1):96–102.
- Daganzo C (1994) The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Res. Part B: Methodological* 28(4):269–287.
- Daganzo C (1995) The cell transmission model, part II: Network traffic. *Transportation Res. Part B: Methodological* 29(2):79–93.
- Delle Monache ML, Reilly J, Samaranayake S, Krichene W, Goatin P, Bayen AM (2014) A pde-ode model for a junction with ramp buffer. *SIAM J. Appl. Math.* 74(1):22–39.
- Dervisoglu G, Gomes G, Kwon J, Horowitz R, Varaiya P (2009) Automatic calibration of the fundamental diagram and empirical observations on capacity. *Transportation Res. Board 88th Annual Meeting, Washington, DC*.
- Duffy A (2009) An introduction to gradient computation by the discrete adjoint method. Technical report, Florida State University, Tallahassee.
- Friesz TL, Bernstein D, Smith TE, Tobin RL, Wie B (1993) A variational inequality formulation of the dynamic network user equilibrium problem. *Oper. Res.* 41(1):179–191.
- Friesz TL, Han K, Neto PA, Meimand A, Yao T (2013) Dynamic user equilibrium based on a hydrodynamic model. *Transportation Res. Part B: Methodological* 47:102–126.
- Garavello M, Piccoli B (2006) *Traffic Flow on Networks* (American Institute of Mathematical Sciences, Springfield, MO).
- Garavello M, Piccoli B (2009) Conservation laws on complex networks. *Annales de l'Institut Henri Poincaré (C) Non Linear Analysis*, 26(5):1925–1951.
- Giles MBM, Pierce NAN (2000) An introduction to the adjoint approach to design. *Flow, Turbulence Combustion* 65(3):393–415.
- Giles MBM, Pierce NAN (1997) Adjoint equations in CFD: Duality, boundary conditions and solution behaviour. AIAA Paper 97-1850.
- Godunov SK (1959) A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik* 89(3):271–306.

- Gomes G, Horowitz R (2006) Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Res. Part C: Emerging Tech.* 14(4):244–262.
- Han K, Friesz TL, Yao T (2013) Existence of simultaneous route and departure choice dynamic user equilibrium. *Transportation Res. Part B: Methodological* 53:17–30.
- Han K, Piccoli B, Friesz TL (2016) Continuity of the path delay operator for dynamic network loading with spillback. *Transportation Res. Part B: Methodological* 92:211–233.
- Jameson A, Martinelli L (2000) Aerodynamic shape optimization techniques based on control theory. Capasso V, Engl HW, Periaux J, eds. *Computational Mathematics Driven by Industrial Problems*, Lecture Notes Math., Vol. 1739 (Springer, Berlin Heidelberg), 151–221.
- Kelly FP (1991) Network routing. *Philos. Trans. Roy. Soc. London. Ser. A: Phys. Engng. Sci.* 337(1647):343–367.
- Korilis YA, Lazar AA, Orda A (1997) Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Trans. Networking* 5(1):161–173.
- Koutsoupias E, Papadimitriou C (1999) Worst-case equilibria. Meinel C, Tison S, eds. *Proc. 16th Annual Conf. Theoret. Aspects Comput. Sci.* (Springer, Berlin Heidelberg), 404–413.
- Krichene W, Reilly J, Amin S, Bayen AM (2013) Stackelberg routing on parallel networks with horizontal queues. *IEEE Trans. Automatic Control* 59(3):714–727.
- Leveque R (2002) *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, Cambridge, UK).
- Li Y, Waller ST, Ziliaskopoulos T (2003) A decomposition scheme for system optimal dynamic traffic assignment models. *Networks Spatial Econom.* 3(4):441–455.
- Lighthill MJ, Whitham GB (1955) On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proc. Roy. Soc. London. Ser. A. Math. Phys. Sci.* 229(1178):317–345.
- Lo HK, Szeto WY (2002) A cell-based variational inequality formulation of the dynamic user optimal assignment problem. *Transportation Res. Part B: Methodological* 36(5):421–443.
- Marti R (2003) Multi-start methods. Glover F, Kochenberger GA, eds. *Handbook of Metaheuristics*, Internat. Series Oper. Res. Management Sci., Vol. 57 (Springer, Boston), 355–368.
- Merchant DK, Nemhauser GL (1978a) A model and an algorithm for the dynamic traffic assignment problems. *Transportation Sci.* 12(3):183–199.
- Merchant DK, Nemhauser GL (1978b) Optimality conditions for a dynamic traffic assignment model. *Transportation Sci.* 12(3):183–199.
- Monache MLD, Goatin P, Piccoli B (2016) Priority-based Riemann solver for traffic flow on networks. arXiv preprint arXiv:1606.07418.
- Morton KW, Mayers DF (2005) *Numerical Solution of Partial Differential Equations: An Introduction* (Cambridge University Press, New York).
- Nie YM (2011) A cell-based Merchant–Nemhauser model for the system optimum dynamic traffic assignment problem. *Transportation Res. Part B: Methodological* 45(2):329–342.
- Peeta S, Ziliaskopoulos A (2001) Foundations of dynamic traffic assignment: The past, the present and the future. *Networks Spatial Econom.* 1(3–4):233–265.
- Qian ZS, Shen W, Zhang H (2012) System-optimal dynamic traffic assignment with and without queue spillback: Its path-based formulation and solution via approximate path marginal cost. *Transportation Res. Part B: Methodological* 46(7):874–893.
- Raffard R (2008) An adjoint-based parameter identification algorithm applied to planar cell polarity signaling. *IEEE Trans. Automatic Control* 53:109–121.
- Reilly J, Samaranayake S, Delle Monache M, Krichene W, Goatin P, Bayen A (2015) Adjoint-based optimization on a network of discretized scalar conservation laws with applications to coordinated ramp metering. *J. Optim. Theory Appl.* 167(2):733–760.
- Richards PI (1956) Shock waves on the highway. *Oper. Res.* 4(1):42–51.
- Riedmiller M, Braun H (1992) Rprop—a fast adaptive learning algorithm. *Proc. Internat. Sympos. Comput. Inform. Sci. VII, Universitat.*
- Roughgarden T (2001) Stackelberg scheduling strategies. *Proc. Thirty-Third Annual ACM Sympos. Theory Comput.* (ACM, New York) 104–113.
- Roughgarden T (2002) The price of anarchy is independent of the network topology. *Proc. Thirty-Fourth Annual ACM Sympos. Theory Comput.* (ACM, New York), 428–437.
- Roughgarden T (2006) On the severity of Braess’s paradox: Designing networks for selfish users is hard. *J. Comput. System Sci.* 72(5):922–953.
- Roughgarden T, Tardos É (2004) Bounding the inefficiency of equilibria in nonatomic congestion games. *Games Econom. Behav.* 47(2):389–403.
- Smith M, Wisten M (1995) A continuous day-to-day traffic assignment model and the existence of a continuous dynamic user equilibrium. *Ann. Oper. Res.* 60(1):59–79.
- Swamy C (2007) The effectiveness of Stackelberg strategies and tolls for network congestion games. *Proc. Eighteenth Annual ACM-SIAM Sympos. Discrete Algorithms*, Vol. 7 (SIAM, Philadelphia), 1133–1142.
- Ukkusuri SV, Han L, Doan K (2012) Dynamic user equilibrium with a path based cell transmission model for general traffic networks. *Transportation Res. Part B: Methodological* 46(10):1657–1684.
- Vickrey WS (1969) Congestion theory and transport investment. *Amer. Econom. Rev.* 59(2):251–260.
- Wachter A, Biegler LT (2005) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Programming* 106(1):25–57.
- Wardrop JG (1952) Some theoretical aspects of road traffic research. *Proc. Institution Civil Engineers* 1:325–378.
- Wie B-W, Tobin RL, Carey M (2002) The existence, uniqueness and computation of an arc-based dynamic network user equilibrium formulation. *Transportation Res. Part B: Methodological* 36(10):897–918.
- Ziliaskopoulos AK (2000) A linear programming model for the single destination system optimum dynamic traffic assignment problem. *Transportation Sci.* 34(1):37–49.