Security of Freeway Traffic Systems: A Distributed Optimal Control Approach

by

Jack Daniel Reilly

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering – Civil and Environmental Engineering

in the

Graduate Division
of the
University of California, Berkeley

Committee in charge:
Professor Alexandre M. Bayen, Chair
Professor Roberto Horowitz
Professor Scott Moura

Fall 2014

Security of Freeway Traffic Systems: A Distributed Optimal Control Approach

Abstract

Security of Freeway Traffic Systems:
A Distributed Optimal Control Approach

by

Jack Daniel Reilly
Doctor of Philosophy in Engineering – Civil and Environmental Engineering

University of California, Berkeley

Professor Alexandre M. Bayen, Chair

This dissertation develops a general, scalable framework for controlling dynamical, networked systems based on mathematical optimization theory, with a strong focus on applications to freeway traffic management. The generality of the framework allows for controllers to consider high-level objectives applied to systems with complex, nonlinear dynamics.

A continuous freeway traffic model and its discretization was developed specifically for onramp metering control. The application serves as the motivating example behind the theory developed subsequently. To apply effective control on such systems, a discrete-adjoint-based *model-predictive-control* (MPC) approach for controlling networked systems of conservation laws is presented, with explicit derivations for ramp metering applications. Linear scalability of the method with respect to network size and time horizon is derived for the discrete adjoint computations. To enable a more asynchronous control architecture, the dissertation presents a distributed optimization algorithm for dynamical, networked systems. The algorithm allows for a physical network to be partitioned into subnetworks that optimize locally and communicate only with adjacent subnetworks to achieve a globally optimal performance.

Within the context of the *Connected Corridors* project associated with UC Berkeley PATH, the developed theory was implemented in a production-level traffic management and simulation environment. Numerical examples applied to the San Diego I15 freeway are presented alongside the theory to motivate the highly practical aspects of the work. Simulations demonstrate the superiority of the MPC approach over existing methods widely used in practice.

The optimization tools are applied to an investigation of security and vulnerabilities of traffic control systems. The potential impact of a compromise of freeway traffic metering lights is analyzed using MPC and multi-objective optimization tools. Several realizable scenarios that exploit traffic system vulnerability locations are constructed and simulated to illustrate the severity of compromises.

Investigations are made into optimal rerouting strategies while controlling only a subset of network flow. A novel behavioral model is developed to account for the interaction of controllable and uncontrollable agents sharing a single flow network, where latency is a function of total flow. Using static freeway traffic models and communication network models, a framework based on convex optimization techniques is presented for computing rerouting policies, with numerical examples given for both freeway and communication networks.

To Mom and Dad

Contents

List of Figures

List of Tables

Acknowledgments

I would first like to thank Professor Alexandre Bayen for inviting me into his lab five years ago and for working closely with me during the entirety of that time. From him I learned much about rigor, thoroughness, persuasiveness, leadership and professionalism: traits which I have found to be the most valuable assets gained during my PhD. I am forever grateful for his patience and guidance.

I would also like to thank Professor Roberto Horowitz, Dr. Gabriel Gomes, Dr. Ajith Muralidharan, and the other researchers in Professor Horowitz's lab. My work has benefited greatly from the different approaches to research and traffic in which I have participated. In particular, their work on linear formulations of freeway optimal control problems [50, 84] greatly improved the practical nature of the theory developed within this thesis. Additionally, Gabriel's suggested extensions of my work to new problems such as network sensitivity analysis and model calibration have illuminated a broader applicability of the underlying research.

I give my gratitude to Professor Scott Moura, Professor Alexander Skabardonis, Professor Eli Yablonovitch, and Professor Steven Glaser for their guidance during my qualifying examination. Their feedback was valuable in directing the remainder of my research.

One of the most productive and enjoyable periods during my PhD was spent at the INRIA research center in Nice, France during Fall 2012, working with Dr. Paola Goatin and her student Maria Laura Delle Monache. I gained much insight into traffic models in general and how to develop a combined ordinary differential equation and partial differential equation model of freeway traffic. Their knowledge and methodology (e.g. mandatory coffee breaks) have permanently affected me.

I would also like to thank Raphael Marinier and Mihai Stroe of Google Road Traffic for their mentorship during my internship in Zurich, Switzerland in Summer 2013. Raphael's expertise in data analysis and focus on concrete results are skills I admire and attempt to emulate. I very much enjoyed the plethora of interesting traffic-related problems they have and their open-mindedness towards their solutions.

I thank again Professor Steven Glaser and thank Professor Raja Sengupta for welcoming me into the Civil Systems program mere days before the start of my Masters. Their novel approach to civil engineering is what reinvigorated my interest in the area, and I still regard my rash decision to switch into the Systems program as the best decision I have made.

I have had the pleasure to collaborate with many different students during my time in

Berkeley. Specifically, I was lucky enough to work closely with Samitha Samaranayake over the last five years on many different projects and in many different venues (Sutardja Dai, La Val's, Nice, Tahoe, Gilman...). Additionally, I benefited greatly from the intelligence and philosophical musings of Sébastien Martin while working on freeway traffic security.

A distinguishing feature of my PhD was the ability to implement my research within production systems. I would like to thank Branko Kerkez and Mario Magliocco for their infinite patience with a greenhorn as I worked on the iShake project. I would also like to thank Joe Butler and many others, including Dimitris Triantafyllos, at CCIT and PATH for their dedication to supporting research in a professional environment and professionalism in a research environment.

To my dearest friends, Devon, Jack, Matty, Zack, Heidi, Timothy, Katelyn, thank you for embodying what I find good in this world.

Finally and foremost, I give my thanks and love to my mother Joanne, father Jim, and my brothers and best friends Jimmy and Christopher.

Chapter 1

Introduction

1.1 Motivation

Modern infrastructure, particularly systems associated with public use such as freeway networks and water supplies, exist within a world with decreasing physical space and resources. For instance, freeways often cannot add additional lanes to accommodate increased demands. Thus, one must rely upon better management systems and control algorithms in order to maximize performance within the limitations of the existing system. These “smart” systems make use of sensing instrumentation to estimate real-time conditions and modeling of the underlying physical dynamics to predict and plan for future states.

The focus of this thesis is the development of methods for the advanced modeling and control of such networked dynamical systems. The goal is to provide operational managers with scalable, flexible, and robust algorithms that can leverage the well-instrumented and highly-connected control infrastructure present on modern systems.

The remainder of this section discusses the *Connected Corridors* project (Section 1.2) on *integrated corridor management* (ICM), which served as the context in which this work was conducted, followed by an overview of *model predictive control* (MPC) methods and their suitability in solving some of the main objectives put forth in the *Connected Corridors* project (Section 1.3). The section concludes with a summary of the original contributions presented in this thesis (Section 1.4).

1.2 Connected Corridors

Connected-Corridors is a project funded by the California Department of Transportation with the goal of creating the next generation of traffic management tools [2, 78]. While most current systems consider the freeway networks as independent from the city-street *arterial* road networks, *Connected Corridors* is tasked with creating an integrated approach to traffic management (referred to as ICM) which accounts for their dual performance. The project has demonstrated innovative control and estimation approaches to ICM on macroscopic and microscopic simulation environments (presented in Section 3.2.3), with the ultimate plan of transferring the knowledge to a physical test-site within California located near the I210 freeway.

The proposed ICM system possesses the following capabilities.

- **Estimation**: Operators have access to a real-time estimation of the traffic state along the major freeways and adjacent arterials [128, 59, 31].
- **Simulation**: Well-calibrated and efficient traffic models allow operators to simulate many different future traffic conditions [82, 28, 58].
- **Control**: Traffic signal and message sign plans are computed online for a variety of specialized objectives and serve as an optimized decision support tool [104, 100].

Control schemes implemented by the system include coordinated traffic light metering plans on freeway onramps (commonly referred as ramp metering, see Section 2.2) and traffic flow diversion around incidents via changeable message signs [113].

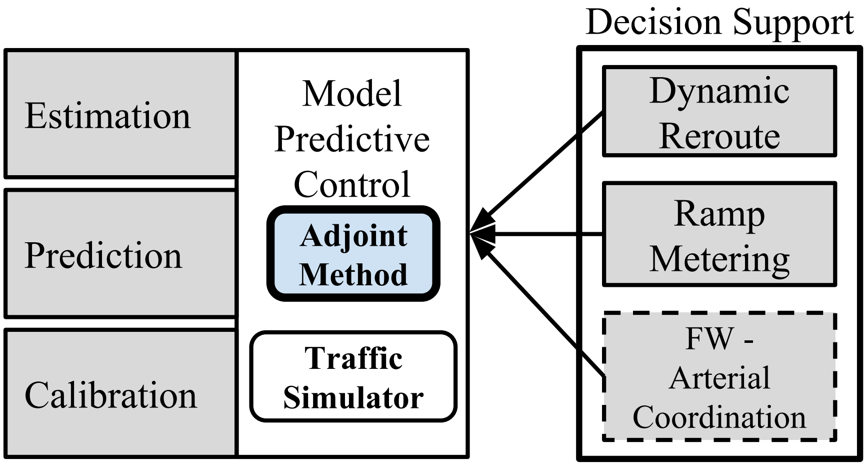


Figure 1.1: The freeway state estimation, prediction and calibration submodules enable an MPC-based framework that computes coordinated, predictive decision-support strategies for numerous applications. Limited customization is required to extend the adjoint-based MPC controller to specific objectives or actuation types.

To satisfy the above requirements, *Connected Corridors* focuses on a number of submodules which are developed independently, but composed in a variety of ways to create high-level, comprehensive support tools. Figure 1.1 shows several of the developed submodules and how they can be composed to create an MPC controller (explained in Section 1.3). Subsequently, the controller submodule is leveraged by a number of actuation strategies which have similar architectural requirements. For instance, both ramp metering and dynamic rerouting require real-time estimation and a calibrated freeway model to compute effective control strategies. Via submodule composition, much of the technology can be reused between the applications. This work contains the theory and implementation of constructing such controller submodules which efficiently exploit the structure of freeway networks.

1.3 Model Predictive Control

Central to the *Connected Corridors* approach of freeway traffic decision support is the concept of *model predictive control* (MPC) [84, 88, 38]. Generally speaking, MPC schemes successively compute a control policy which maximizes the performance, according to a provided criteria, of the system over the immediate future, where this work assumes the future to be of finite duration. The policy generated from the MPC scheme is recomputed as frequently as the real-time state estimation and computation times permit. Thus, associated with an MPC problem are two time periods: the *update time* representing how often the MPC policy is recomputed online, and the *time horizon* representing how far into the future for which the MPC policy will account. A more detailed treatment of MPC schemes is given in Section 3.2.2.

The requirements of an online MPC controller for a traffic system are depicted in Figure 1.1. At the base of the method is a mathematical model of the physical system, referred to as the *forward system.* These models often require calibration using historical sensor measurements to set the model parameters. To fully specify the simulation, an *initial condition* is specified (i.e. estimation), as well as the *boundary conditions* at the exteriors of the system over the entire time horizon of the MPC problem (i.e. prediction). All these requirements are satisfied by components within the *Connected Corridors* system [82, 28, 128], making MPC control a natural fit for research and development in the project.

The focus of this thesis is the efficient and flexible design of MPC optimization techniques and their applications to freeway control systems. At the heart of the developed approach is the *discrete adjoint* method for gradient computations within first-order gradient descent methods (Chapter 3). The efficiency of the developed method enables *online* application of MPC to the control of large freeway networks (on the order of tens of miles) with frequent recomputations (on the order of one minute) to “close the loop” with real-time measurements.

In addition to the well-established ramp metering control infrastructure, the emergence of smartphones and connected vehicles and crowd-sourcing data collection [101, 32, 26] has made mass rerouting strategies a new, viable form of control. This thesis showcases this potential by including an investigation of rerouting strategies for a subset of flow on networks. Inspired by the increasing penetration of navigation devices in vehicles (e.g. dedicated units, smartphone applications), a single navigation provider may advise a signification portion of the total flow on some networks, thus potentially affecting traffic conditions based on their advice. This potential can be harnessed to reduce the inefficiencies of *selfish routing* [110, 71] and drive network behavior towards *socially optimal* conditions.

1.4 Contributions

This thesis contains the following contributions to the problem of optimal control of networked dynamical systems, applied to the field of traffic control.

- **A novel continuous freeway traffic model suitable for finite-horizon optimal control problems** [27, 104].
  - The model represents an extension of the networked *Lighthill-Richards-Whitham* (LWR) PDE traffic model [75, 106] proposed in [42], where onramps are modeled as ODE buffers to guarantee *strong* boundary conditions and flow conservation at the network boundaries.
  - A discretized version of the continuous model is derived for optimal control application using Godunov's method [49, 73].
- **A method for optimal control of networked conservation law systems based on discrete adjoint calculations** [104, 113].
  - This work develops a framework for converting a continuous time and space control problem on a physical network, where edges behave according to a conservation law, into a discrete, finite-horizon optimal control problem using Godunov discretization.
  - The application of the discrete adjoint method [45, 30, 25, 55] to compute gradients for the above problem is presented, with an analysis of the linear scalability of the approach in discrete time and discrete space for sparse network structures.
  - An explicit formulation of the discrete adjoint method is given for the application of coordinated ramp metering control [93, 38, 69, 50] for freeway networks.
  - Simulations of MPC on a macroscopic model of the I15 South Freeway in San Diego, California demonstrate the practical nature and the robustness to measurement noise of the research.
- **A decentralized algorithm and control infrastructure for networked dynamical systems** [100].
  - This thesis presents a new distributed optimization method based on the *alternating directions methods of multipliers* (ADMM) [10, 41, 80] algorithm for solving optimal control problems over subnetworks in parallel. The splitting method is done in such a way to only require communication between physically-neighboring subnetworks.
  - Differing from similar work where subsystems only share control variables [80, 15], the presented method allows for subsystems to share both control *and state* variables, an assumption necessary for the distributed control of traffic networks and hydrological systems.

  - A discrete adjoint formulation is presented for efficient solution of subsystems with coupled control and state variables.
  - An implementation of a distributed ramp metering and variable speed limit controller is simulated on a realistic macroscopic freeway network, demonstrating advantages over other proposed communicative controllers.

- **An analysis of the security of traffic control systems and coordinated ramp metering attacks** [103].
  - A classification of traffic system vulnerability locations is constructed across such categories as cost of attack, effectiveness, and directness of control.
  - A novel analysis of the potential damage and impact of coordinated ramp metering attacks is conducted using adjoint-based optimal control and multi-objective optimization tools.
  - Illustrative attack scenarios are constructed and numerically investigated on realistic freeway networks. A web-based coordinated ramp metering attack tool was created to implement the interactive optimization approaches presented in the work [102].

- **A framework for rerouting a subset of vehicles on freeway networks**.
  - As opposed to standard system-optimal routing problems [130, 68] where all flow is controllable, we construct a network-flow optimization problem which allows one to specify a *compliance rate* of cooperative vehicles. The proposed model captures the delays induced by the noncompliant vehicles.
  - To account for the possibility of noncompliant drivers adapting to the new flow conditions, we propose a behavioral model, referred to as *bounded tolerance*, which assumes that noncompliant drivers have bounded rationality and will only switch routes under significant increases in delay.
  - Numerical examples for communication networks and freeway systems demonstrate the effectiveness of the method.

1.5 Organization

The rest of the article is organized as follows.

Chapter 2 presents the continuous and discrete freeway models which serves as the running application of the theory presented subsequently. After covering preliminaries on networked PDE systems, the derivation of and motivation behind the model are given.

Chapter 3 presents the discrete adjoint approach to optimal control of networked conservation laws. Presented first is a general overview of the discrete adjoint method, followed by its specific instantiation for discretized physical network systems. The application to coordinated ramp metering is then presented with numerical examples. The chapter concludes with the distributed, asynchronous formulation of the adjoint optimal control method via subnetwork splitting.

Chapter 4 presents a study of traffic control systems and their vulnerabilities. After defining the specific control system under consideration and its security weaknesses, the work gives an in-depth study of coordinated ramp metering attacks.

Finally, we conclude with a overview of the work presented, as well as directions for future research.

The material presented in Chapters 2 and 3 was done collaboratively as part of the *Optimal Reroute Strategies for Traffic Management* (ORESTE) project between UC Berkeley and INRIA research institute in Sophia-Antipolis, France. Chapter 4 also contains research stemming from collaborations with Sebastien Martin and Mathias Payer. Chapters 3.3.3 and 5 contains research conducted independently.

Chapter 2

Freeway Network Model

The problem of freeway oversaturation is well-documented [114], with $100 billion in costs and 56 billion lbs in CO_2 attributed to roadway congestion. More efficient freeway management systems are developed to counter the above costs. Examples of such control systems include the following:

- **Ramp metering:** Traffic lights installed on the onramps leading to freeway mainlines serve the purpose of limiting the amount of total flow entering the mainline during peak operation periods, when vehicle demand exceeds the total capacity of the mainline. Feedback-based ramp metering algorithms have been applied successfully in practice [94, 93, 95], while many predictive algorithms have shown promise in simulation environments [104, 50, 69].
- **Variable speed limits:** While metering on onramps is one way of reducing demand, the mainline flow can be reduced by limiting the maximum speed of its vehicles [84].
- **Flow rerouting:** In situations where excess demand exists on the neighboring road network or vehicles choose routes selfishly or suboptimally [71, 63, 70, 111], then route-choice intervention can lead to improved traffic conditions [113, 130].

In order to implement the above traffic control strategies, one requires an accurate and computationally efficient model of freeway dynamics which is sensitive to time-varying demands and temporal changes in the physical properties of the freeway (e.g. lane closure during reroutes, weather influencing maximum speeds). A common approach, which this thesis adopts, is to treat vehicle flow as a continuum of *vehicle density* and develop continuous, distributed parameter system models tracking the evolution of the traffic density. These *macroscopic* traffic models have been shown to accurately capture traffic dynamics [92] and possess better analytical and computational properties than *microscopic*, particle-based models. While microscopic models have a potential for greater extensibility and robustness, they are often prohibitively hard to calibrate due to the number of parameters and harder to analyze compared to macroscopic models. For these reasons, the following work focuses on macroscopic models for development of theory and models, while leveraging microscopic models occasionally for validation.

This section first covers the preliminaries of continuous, conservation laws, a type of *partial differential equation* (PDE) system, and discretization techniques applied to conservation laws for computational and numerical purposes. Building off the preliminaries, we then present novel continuous and discrete freeway traffic models [27] which are specifically developed for freeway traffic management applications.

2.1 Preliminaries of Networked Conservation Laws

2.1.1 Networked Conservation Laws

We consider the non-linear conservation equation of the form:

$$\partial_t \rho(t,x) + \partial_x f(\rho(t,x)) = 0 \quad (t,x) \in \mathbb{R}^+ \times \mathbb{R} \tag{2.1}$$

where $\rho = \rho(t,x) \in \mathbb{R}^+$ is the scalar conserved quantity and $f : \mathbb{R}^+ \to \mathbb{R}^+$ is a Lipschitz continuous flux function [12]. Throughout the article we suppose that f is a concave function. The Cauchy problem to solve for the evolution of the conservation law is then

$$\begin{cases} \partial_t \rho + \partial_x f(\rho) = 0, & (t,x) \in \mathbb{R}^+ \times \mathbb{R}, \\ \rho(0,x) = \rho^0(x), & x \in \mathbb{R} \end{cases} \tag{2.2}$$

where $\rho^0(x)$ is the initial condition. It can be shown that there exists a unique weak entropy solution for the Cauchy problem(2.2), as described in Definition 2.1.1.

Definition 2.1.1. *A function $\rho \in \mathcal{C}^0(\mathbb{R}^+; \mathbf{L}^1_{loc} \cap \mathbf{BV})$ is an admissible solution to* (2.2) *if ρ satisfies the Kružhkov entropy condition [72] on $(\mathbb{R}^+ \times \mathbb{R})$, i.e.,for every $k \in \mathbb{R}$ and for all $\varphi \in \mathcal{C}^1_c(\mathbb{R}^2; \mathbb{R}^+)$,*

$$\begin{aligned} \textstyle\int_{\mathbb{R}^+} \int_{\mathbb{R}} (|\rho - k| \partial_t \varphi + \operatorname{sgn}(\rho - k)(f(\rho) - f(k)) \partial_x \varphi) dx dt \\ + \textstyle\int_{\mathbb{R}} |\rho^0 - k| \varphi(0,x) dx \geq 0. \end{aligned} \tag{2.3}$$

For further details regarding the theory of hyperbolic conservation laws we refer the reader to [42, 33].

Networks A network of hyperbolic conservation laws such as (2.1) is defined as a set of N links $\mathcal{L} = \{1, \ldots, N\}$, with junctions $\mathcal{J}$. Each junction $j \in \mathcal{J}$ is defined as the union of two non-empty sets: the set of n_j incoming links $\text{Inc}(j) = \left(l_j^1, \ldots, l_j^{n_j}\right) \subset \mathcal{L}$ and the set of m_j outgoing links $\text{Out}(j) = \left(l_j^{n_j+1}, \ldots, l_j^{n_j+m_j}\right) \subset \mathcal{L}$. Each link $l \in \mathcal{L}$ has an associated upstream junction $j_l^{\text{U}} \in \mathcal{J}$ and downstream junction $j_l^{\text{D}} \in \mathcal{J}$, and has an associated spatial domain $(0, L_l)$ over which the evolution of the state on link l, $\rho_l(t,x)$, solves the Cauchy problem:

$$\begin{cases} (\rho_l)_t + f(\rho_l)_x & = 0 \\ \rho_l(0,x) & = \rho_l^0(x) \end{cases} \tag{2.4}$$

where $\rho_l^0 \in BV \cap L^1_{\text{loc}}(L_i;\mathbb{R})$ is the initial condition on link l. For simplicity of notation, this section considers a single junction $j \in \mathcal{J}$ with $\text{Inc}(j) = (1,\dots,n)$ and $\text{Out}(j) = (n+1,\dots,n+m)$.

Remark 2.1.1. *There is redundancy in the labeling of the junctions, if link i is directly upstream of link j, then we have $j_l^D = j_j^U$. See Fig. 2.2.*

2.1.2 Riemann Solvers

While the dynamics on each link $\rho_l(t,x)$ is determined by (2.4), the dynamics at junctions still needs to be defined. This section describes *Riemann solvers*, which provide the solution of the system at junction points. The solution of *Riemann problems* between 1×1 junctions serve as building blocks for Riemann solvers, and thus we describe Riemann problems first.

Definition 2.1.2. *Riemann Problem.*

A Riemann problem is a Cauchy problem (2.2) *with a piecewise-constant initial datum (called the Riemann datum):*

$$\bar{\rho}(x) = \begin{cases} \rho_- & x < 0 \\ \rho_+ & x \geq 0 \end{cases} \tag{2.5}$$

We denote the corresponding self-similar entropy weak solutions by $W_R\left(\frac{x}{t};\rho_-,\rho_+\right)$.

Definition 2.1.3. *Riemann problem at junctions.*

A Riemann problem at j is a Cauchy problem corresponding to an initial datum $(\bar{\rho}_1,\dots,\bar{\rho}_{n+m}) \in \mathbb{R}^{n+m}$ which is constant on each link l.

Definition 2.1.4. *A Riemann solver is a map that assigns a solution to each Riemann initial data. For each junction j it is a function*

$$\begin{aligned} RS: \quad & \mathbb{R}^{m+n} && \to \mathbb{R}^{m+n} \\ & (\bar{\rho}_1,\dots,\bar{\rho}_{n+m}) && \mapsto RS(\bar{\rho}_1,\dots,\bar{\rho}_{n+m}) = (\hat{\rho}_1,\dots,\hat{\rho}_{n+m}) \end{aligned}$$

where $\hat{\rho}_l$ provides the trace for link l at the junction for all time $t \geq 0$.

For a link $i \in \text{Inc}(j)$, the solution $\rho_i(t,x)$ over its spatial domain $x < 0$ is given by the solution to the following Riemann problem:

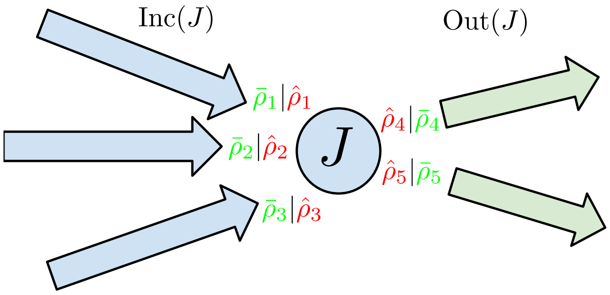

Inc(J)
Out(J)
$\bar{\rho}_1|\hat{\rho}_1$
$\bar{\rho}_2|\hat{\rho}_2$
$\bar{\rho}_3|\hat{\rho}_3$
$\hat{\rho}_4|\bar{\rho}_4$
$\hat{\rho}_5|\bar{\rho}_5$
J

Figure 2.1: Solution of boundary conditions at junction. The boundary conditions $(\hat{\rho}_1, \ldots, \hat{\rho}_5)$ are produced by applying the Riemann solver to the initial conditions, $(\bar{\rho}_1, \ldots, \bar{\rho}_5)$.

$$\begin{cases} (\rho_l)_t + f(\rho_l)_x & = 0 \\ \rho_l(0, x) & = \begin{cases} \bar{\rho}_l & x < 0 \\ \hat{\rho}_l & x \geq 0, \end{cases} \end{cases} \tag{2.6}$$

The Riemann problem for an outgoing link is defined similarly, with the exception that $\rho_l(0, x > 0) = \bar{\rho}_l$ and $\rho_l(0, x \leq 0) = \hat{\rho}_l$.

Fig. 2.1 gives a depiction of Riemann solution at the junction.

Note that the following properties for the Riemann Solver holds:

- All waves produced from the solution to Riemann problems on all links, generated by the boundary conditions at a junction, must emanate out from the junction. Moreover, the solution to the Riemann problem on an incoming link must produce waves with negative speeds, while the solution on an outgoing link must produce waves with positive speed.

- The sum of all incoming fluxes must equal the sum of all outgoing fluxes:

$$\sum_{i \in \text{Inc}(j)} f(\hat{\rho}_l) = \sum_{j \in \text{Out}(j)} f(\hat{\rho}_j).$$

  This condition guarantees mass conservation at junctions.

- The Riemann solver must produce self-similar solutions, i.e.

$$RS(RS(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m})) = RS(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}) = (\hat{\rho}_1, \ldots, \hat{\rho}_{n+m})$$

The justification for these conditions can be found in [42].

The above conditions are not always sufficient to guarantee a unique Riemann solver. Additional conditions are added for specific applications to achieve uniqueness, chosen to model physical phenomena at junctions. In Section 2.2.2, we detail the additional conditions added to the ramp-metering solver which enforce flux maximization along the freeway mainline sections and specify a merging priority model for vehicles entering from the onramps.

2.1.3 Godunov Discretization

In order to find approximate solutions we use the classical Godunov scheme [49]. We use the following notation: $x_{j+\frac{1}{2}}$ are the cell interfaces and $t^k = k\Delta t$ the time with $k \in \mathbb{N}$ and $j \in \mathbb{Z}$. x_j is the center of the cell, $\Delta x = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ the cell width, and Δt is the time step.

Godunov scheme for a single link. The Godunov scheme is based on the solutions of exact Riemann problems. The main idea of this method is to approximate the initial datum by a piecewise constant function, then the corresponding Riemann problems are solved exactly and a global solution is found by piecing them together. Finally one takes the mean on the cell and proceed by iteration. Given $\rho(t,x)$, the cell average of ρ at time t^k in the cell $C_j =]x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ is given by

$$\rho_j^k = \frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \rho(t^k,x)dx. \tag{2.7}$$

Then we proceed as follows:

1. We solve the Riemann problem at each cell interface $x_{j+\frac{1}{2}}$ with initial data (ρ_j^k, ρ_{j+1}^k).

2. Compute the cell average at time t^{k+1} in each computational cell and obtain ρ_j^{k+1}.

We remark that waves in two neighbouring cells do not intersect before Δt if the following Courant–Friedrichs–Lewy (CFL) condition holds, $\lambda^{\max} \leq \frac{\Delta x}{\Delta t}$, where $\lambda^{\max} = \max_a |f'(a)|$ is the maximum wave speed of the Riemann solution at the interfaces.
Godunov scheme can be expressed as follows:

$$\rho_j^{k+1} = \rho_j^k - \frac{\Delta t}{\Delta x}(g^G(\rho_j^k, \rho_{j+1}^k) - g^G(\rho_{j-1}^k, \rho_j^k)), \tag{2.8}$$

where g^G is the Godunov numerical flux given by

$$\begin{aligned} g^G : \quad & \mathbb{R}\times\mathbb{R} && \to \mathbb{R} \\ & (\rho_j, \rho_{j+1}) && \mapsto g^G(\rho_j, \rho_{j+1}) = f(W_R(0;\rho_j,\rho_{j+1})). \end{aligned}$$

where W_R is as defined in Definition 2.1.2.

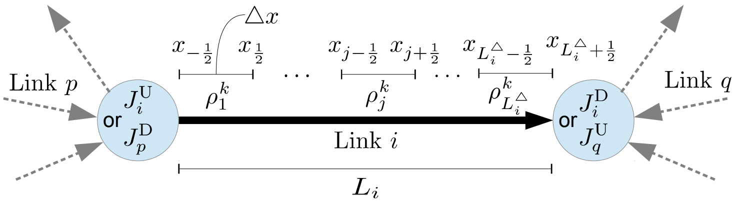

$\triangle x$
$x_{-\frac{1}{2}}$
$x_{\frac{1}{2}}$
$\cdots$
$x_{j-\frac{1}{2}}$
$x_{j+\frac{1}{2}}$
$\cdots$
$x_{L_i^{\triangle}-\frac{1}{2}}$
$x_{L_i^{\triangle}+\frac{1}{2}}$
Link p
Link q
ρ_1^k
ρ_j^k
$\rho_{L_i^{\triangle}}^k$
J_i^{U} or J_p^{D}
J_i^{D} or J_q^{U}
Link i
L_i

Figure 2.2: Space discretization for a link $l \in \mathcal{L}$. Step size is uniform Δx, with discrete value ρ_j^k representing the state between x^{j-1} and x^j.

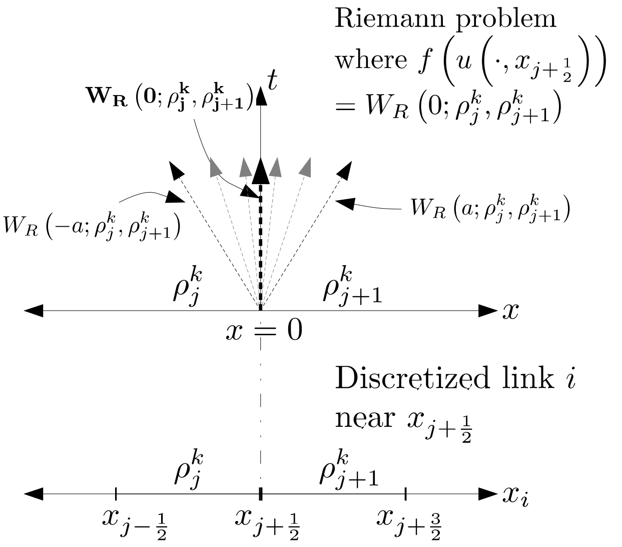

Riemann problem
where $f\left(u\left(\cdot, x_{j+\frac{1}{2}}\right)\right)$
$= W_R\left(0; \rho_j^k, \rho_{j+1}^k\right)$
$\mathbf{W_R}\left(\mathbf{0}; \boldsymbol{\rho}_\mathbf{j}^\mathbf{k}, \boldsymbol{\rho}_\mathbf{j+1}^\mathbf{k}\right)$
t
$W_R\left(-a; \rho_j^k, \rho_{j+1}^k\right)$
$W_R\left(a; \rho_j^k, \rho_{j+1}^k\right)$
ρ_j^k
ρ_{j+1}^k
x
$x = 0$
Discretized link i
near $x_{j+\frac{1}{2}}$
ρ_j^k
ρ_{j+1}^k
x_i
$x_{j-\frac{1}{2}}$
$x_{j+\frac{1}{2}}$
$x_{j+\frac{3}{2}}$

Figure 2.3: Self-similar solution for Riemann problem with initial data $\left(\rho_j^k, \rho_{j+1}^k\right)$. The self-similar solution at $\frac{x}{t} = 0$ for the top diagram (i.e. $W_R\big(0; \rho_j^k, \rho_{j+1}^k\big)$), gives the flux solution to the discretized problem in the bottom diagram.

Godunov scheme at junctions. The scheme just discussed applies to the case in which a single cell is adjacent to another single cell. Yet, at junctions, a cell may share a boundary with more than one cell. A more general Godunov flux can be derived for such cases. For incoming links near the junction, we have:

$$\rho^{k+1}_{L^\Delta_l} = \rho^k_{L^\Delta_l} - \frac{\Delta t}{\Delta x}(f(\hat{\rho}^k_{L^\Delta_l}) - g^G(\rho^k_{L^\Delta_i - 1}, \rho^k_{L^\Delta_i})), \qquad l \in \{1, \dots, n\}$$

where L^Δ_i are the number of cells for link i (see Fig. 2.2) and $\hat{\rho}^k_i$ is the solution of the Riemann solver $RS\big(\rho^k_1, \dots, \rho^k_{n+m}\big)$ for link l at the junction. The same can be done for the outgoing links:

$$\rho^{k+1}_1 = \rho^k_1 - \frac{\Delta t}{\Delta x}(g^G(\rho^k_1, \rho^k_2) - f(\hat{\rho}^k_1)), \qquad l \in \{n+1, \dots, n+m\}$$

Remark 2.1.2. *Using the Godunov scheme, each mesh grid at a given t^k can be seen as a node for a 1-to-1 junction with one incoming and one outgoing link. It is therefore more convenient to consider that every discretized cell is, rather, a link with both an upstream and downstream junction. Thus, we consider networks in which the state of each link $l \in \mathcal{L}$ at a time-step $k \in \{0, \dots, T-1\}$ is represented by the single discrete value ρ^k_l.*

The previous remark allows us to develop a generalized update step for all discrete state variables. We first introduce a definition in order to reduce the cumbersome nature of the preceding notation. Let the state variables adjacent to a junction $j \in \mathcal{J}$ at a time-step $k \in \{0, \dots, T-1\}$ be represented as $\vec{\rho}^{\,k}_j := \left(\rho^k_{l^1_j}, \dots, \rho^k_{l^{n_j+m_j}_j}\right)$. Similarly, we let the solution of a Riemann solver be represented as $\vec{\hat{\rho}}_j := RS\big(\vec{\rho}_j\big)$. Then, for a link $l \in \mathcal{L}$ with upstream and downstream junctions, j^{U}_l and j^{D}_l, and time-step $k \in \{0, \dots, T-1\}$, the update step becomes:

$$\begin{aligned}\rho^{k+1}_l &= \rho^k_l - \frac{\Delta t}{\Delta x}\Big(f\Big(\Big(RS\Big(\vec{\rho}^{\,k}_{j^{\mathrm{D}}_l}\Big)\Big)_l\Big) - f\Big(\Big(RS\Big(\vec{\rho}^{\,k}_{j^{\mathrm{U}}_l}\Big)\Big)_l\Big)\Big)\\ &= \rho^k_l - \frac{\Delta t}{\Delta x}\Big(f\Big(\Big(\vec{\hat{\rho}}_{j^{\mathrm{D}}_l}\Big)_l\Big) - f\Big(\Big(\vec{\hat{\rho}}_{j^{\mathrm{U}}_l}\Big)_l\Big)\Big) \end{aligned} \tag{2.9}$$

where $(s)_i$ is the ith element of the tuple s. This equation is thus a general way of writing the Godunov scheme in a way which applies everywhere, including at junctions.

Working directly with flux solutions at junctions. The equations can be simplified if we do not explicitly represent the solution of the Riemann solver, $\vec{\hat{\rho}}_j$, and, instead, directly calculate the flux solution from the Riemann data. We denote this direct computation by g^G_j, the Godunov flux solution at a junction:

$$\begin{aligned} g^G_j : \ \mathbb{R}^{n_j+m_j} &\to \mathbb{R}^{n_j+m_j}\\ \vec{\rho}_j &\mapsto f\big(RS\big(\vec{\rho}_j\big)\big) = (f(\hat{\rho}_1), \dots, f(\hat{\rho}_{n+m})). \end{aligned} \tag{2.10}$$

Algorithm 1 Riemann solver update procedure

```
Input: initial state at time $t = k\Delta t$, $\left(\rho_l^k : l \in \mathcal{L}\right)$
Output: resulting state at time $t = (k+1))\Delta t$, $\left(\rho_l^{k+1} : l \in \mathcal{L}\right)$

for junction $j \in \mathcal{J}$:
        # Apply Riemann solver to $j$
        $\vec{\hat{\rho}}_j^k = RS\left(\vec{\rho}_j^k\right)$
for link $l \in \mathcal{L}$:
        # update density on link $l$ with junction fluxes
        $\rho_l^{k+1} = \rho_l^k - \frac{\Delta t}{\Delta x}\left(f\left(\left(\vec{\hat{\rho}}_{j_l^{\mathtt{D}}}^k\right)_l\right) - f\left(\left(\vec{\hat{\rho}}_{j_l^{\mathtt{U}}}^k\right)_l\right)\right)$
```

Algorithm 2 Godunov junction flux update procedure

```
Input: initial state at time $t = k\Delta t$, $\left(\rho_l^k : l \in \mathcal{L}\right)$
Output: resulting state at time $t = (k+1))\Delta t$, $\left(\rho_l^{k+1} : l \in \mathcal{L}\right)$

for link $l \in \mathcal{L}$:
        # update density on link $l$ with direct Godonuv fluxes
        $\rho_l^{k+1} = \rho_l^k - \frac{\Delta t}{\Delta x}\left(\left(g_{j_l^{\mathtt{D}}}^G\left(\vec{\rho}_{j_l^{\mathtt{D}}}^k\right)\right)_l - \left(g_{j_l^{\mathtt{U}}}^G\left(\vec{\rho}_{j_l^{\mathtt{U}}}^k\right)\right)_l\right)$
```

This gives a simplified expressions for the update step:

$$\rho_l^{k+1} = \rho_l^k - \frac{\Delta t}{\Delta x}\left(\left(g_{j_l^{\mathrm{D}}}^G\left(\vec{\rho}_{j_l^{\mathrm{D}}}^k\right)\right)_l - \left(g_{j_l^{\mathrm{U}}}^G\left(\vec{\rho}_{j_l^{\mathrm{U}}}^k\right)\right)_l\right). \tag{2.11}$$

Full discrete solution method. We assume a discrete scalar hyperbolic network of PDEs with links $\mathcal{L}$ and junctions $\mathcal{J}$, and a known discrete state at time-step k, $\left(\bar{\rho}_l^k : l \in \mathcal{L}\right)$. The solution method for advancing the discrete system forward one time-step is given in Algorithm (1), or alternatively Algorithm (2).

Algorithm 1 takes as input the state at a time-step k for all links $\left(\rho_l^k : l \in \mathcal{L}\right)$ and returns the state advanced by one time-step $\left(\rho_l^{k+1} : l \in \mathcal{L}\right)$. The algorithm first iterates over all junctions j, calculating all the boundary conditions, $\vec{\hat{\rho}}_j^k$. Then, the algorithm iterates over all links $l \in \mathcal{L}$ to compute the updated state ρ_l^{k+1} using the previously computed boundary conditions, as in 2.9.

Algorithm 2 is similar to Algorithm 1, except that the boundary conditions $\vec{\hat{\rho}}_j^k$ are not explicitly computed, but rather the Godunov flux solution is used to update the state, as in 3.1.1. Algorithm 2 is more suitable if a Godunov flux solution is derived for solving junctions, while Algorithm 1 is more suitable if one uses a Riemann solver at junctions.

2.2 Continuous and Discrete Traffic Model for Freeway Control

In this section, we derive and motivate the continuous freeway network traffic model and discuss its improvements over existing models. We also present the discretized version of the continuous model, which is used extensively in applications in the remainder of the thesis.

2.2.1 LWR Equation

The *Lighthill-Whitham-Richards* (LWR) equation [75, 106] is a scalar conservation law used to represent the evolution of vehicle density on a section of linear roadway. The distinguishing assumptions in the LWR model deal with the flux function, $f(\rho)$, referred to as the *fundamental diagram of traffic*. Namely, we assume the following rules on f:

1. $f(\rho) = \rho v(\rho)$, where v is the velocity of the vehicle density.
2. $v(\rho)$ is a decreasing function of ρ.
3. f is defined over the values $[0, \rho^{\max}]$, where $\rho^{\max}$ is considered the *jam* density.
4. $f(0) = f(\rho^{\max}) = 0$.

The four rules above fit well with our intuition of road traffic. Rules 1 and 2 state that the flux varies as the velocity varies, and that as the roadway gets more congested, the speed of the vehicles will only decrease. Rule 3 fits with the physical interpretation of vehicle as being non-negative, and that there must be an upper limit of vehicle density (due to minimum car lengths). Rule 4 states that no vehicles will have no flow, and that flow completely breaks down at the maximum density.

An example of a quadratic fundamental diagram, known as the *Greenshields* flux function, is given in Figure 2.4. The term *critical density*, ρ^{cr}, is reserved for the density value where the maximum vehicle flux, $f^{\max}$, is obtained. The maximum flux can also be viewed as the *capacity* of the road under consideration, where demand in excess of the maximum flux will lead to congestion and traffic jams.

2.2.2 Continuous PDE-ODE Freeway Model

We concern ourselves with a *linear* freeway section, meaning that we are only interested in one freeway mainline, with any number of onramps and offramps coming together at junctions. While the approach can be readily extended to mainline-to-mainline junctions, we exclude the analysis for the sake of presentation.

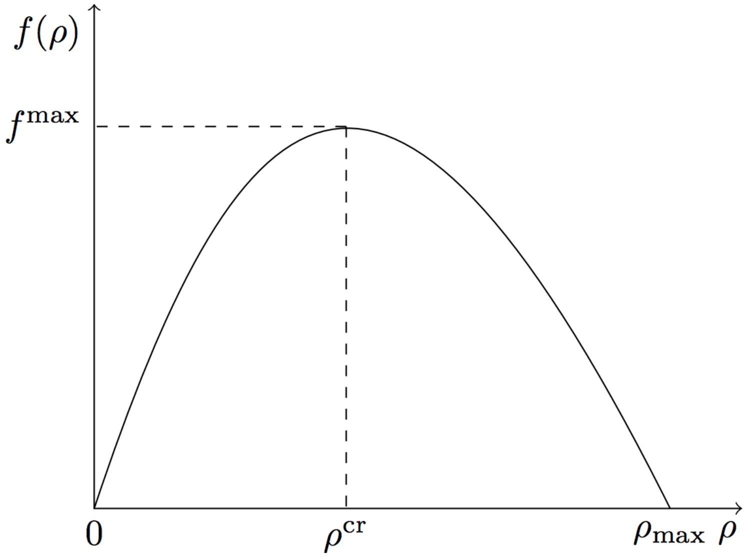


Figure 2.4: The Greenshields (quadratic) flux function is one example of a fundamental diagram.

Thus, a freeway network can be viewed as a sequence of junctions, where each junction contains four links: an upstream mainline, a downstream mainline, an onramp and an offramp, as visualized in Figure 2.5. Note that a single mainline link (i.e. a stretch of mainline in between two junction points) will serve as the upstream mainline of one junction and the downstream mainline of the subsequent junction.

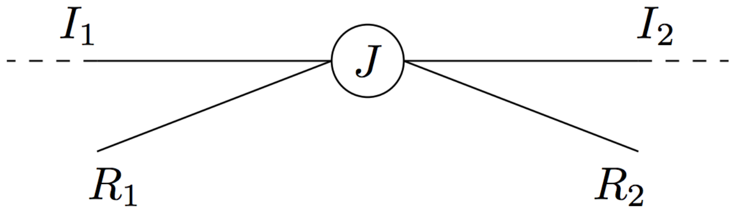


Figure 2.5: A freeway junction consisting of an upstream mainline I_1, downstream mainline I_2, onramp R_1 and offramp R_2.

Weak Boundary Conditions and Vehicle Conservation

In reality, one cannot consider the evolution of a stretch of freeway in complete isolation with respect to its surrounding traffic network, as the dynamics are coupled at every junction point via Riemann solvers (Section 2.1.2). Thus, to account for the behavior at the extremities of the network, one must consider boundary conditions.

The standard approach to boundary conditions is to prescribe a time-varying density $\rho^0(t)$ at each extremity point of the network. Due to the concave shape of the fundamental diagram of traffic, density waves may propagate from within the system outwards to the network extremities, in both the upstream and downstream directions.

As an example, one could consider the behavior upstream of onramp R_1 as being the solution of a Riemann problem of the form in Equation (2.5), where ρ_- is the upstream boundary condition, and ρ_+ is the state within an onramp. Whenever $f(\rho_+) < f(\rho_-)$ and $\rho_+ > \rho^{\text{cr}}$, then it can be shown [73, 42] that the vehicle flux across the boundary is $f(\rho_+)$ and is thus insensitive to the value of $f(\rho_-)$. One can view this event as a *loss of information* at the left boundary of the network, as the backward-moving congestion wave prevented information about the boundary condition from entering the network. Systems which possess this property are said to have *weak boundary conditions* [117].

This property of traffic network modeling is undesirable in traffic management applications, as the flux of vehicles at network boundaries is dependent upon the state of the system, which in turn is dependent upon the control scheme being applied. Summarizing, different control schemes can lead to different vehicle demands, which is not a realistic assumption, can actually be exploited by control schemes. As the goal of the current traffic model is to be used in control applications, we develop an alternative approach which effectively turns the weak boundary conditions into *strong* boundary conditions which guarantee vehicle flux conservation.

Onramps as ODE Buffers

Instead of modeling boundary conditions as vehicle densities, we consider a time-varying boundary *flux*, $D(t)$ entering onramp R_1 and make the simplifying assumption that the offramp R_2 has infinite capacity and thus does not influence the evolution of the system[1].

The onramp R_1 stores the boundary flux in a vehicle *buffer* modeled by the following ordinary differential equation (ODE):

$$\frac{dl(t)}{dt} = D(t) - r(t), \quad t \in \mathbb{R}^+, \tag{2.12}$$

where r is the flux of vehicles exiting the onramp onto the downstream mainline I_2.

The onramp ODE models the conservation of boundary flux in a *vertical* buffer of infinite capacity, as opposed to a spatially distributed *horizontal* queue with finite capacity, until there is enough capacity on the downstream mainline to empty the queue.

As the offramp R_2 possesses no state, it does not require an ODE buffer. The behavior of vehicles at the offramp is captured via a *split ratio* parameter $\beta(t) \in [0, 1]$ which specifies the fraction of vehicles which move from I_1 to I_2, where $1 - \beta(t)$ is the fraction of vehicles

[1] Motivation behind the offramp model is the focus on ramp-metering applications in this thesis, and the general lack of available sensor data on freeway offramps, making accurate modeling of offramp state difficult.

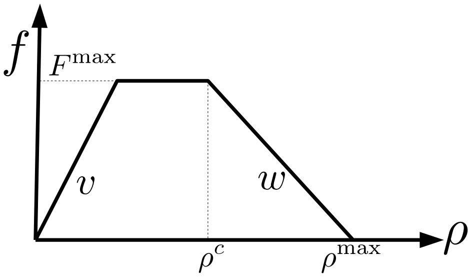


Figure 2.6: Fundamental diagram (the name of the flux function in transportation literature) with free-flow speed v, congestion wave speed w, max flux $F^{\max}$, critical density ρ^c, and max density $\rho^{\max}$.

leaving the freeway from I_1 to R_2. It is assumed that no vehicles from R_1 immediately exit to R_2.

Thus, the Cauchy problem we wish to solve across the four-link system is as follows:

$$\partial_t \rho_i + \partial_x f(\rho_i) = 0, \quad (t,x) \in \mathbb{R}^+ \times I_i,\ i = 1,2 \tag{2.13}$$

$$\frac{dl(t)}{dt} = D(t) - r(t), \quad t \in \mathbb{R}^+ \tag{2.14}$$

$$\rho_i(0,x) = \rho_{i,0}(x), \quad \text{On } I_i\, i = 1,2 \tag{2.15}$$

$$l(0) = l_0, \tag{2.16}$$

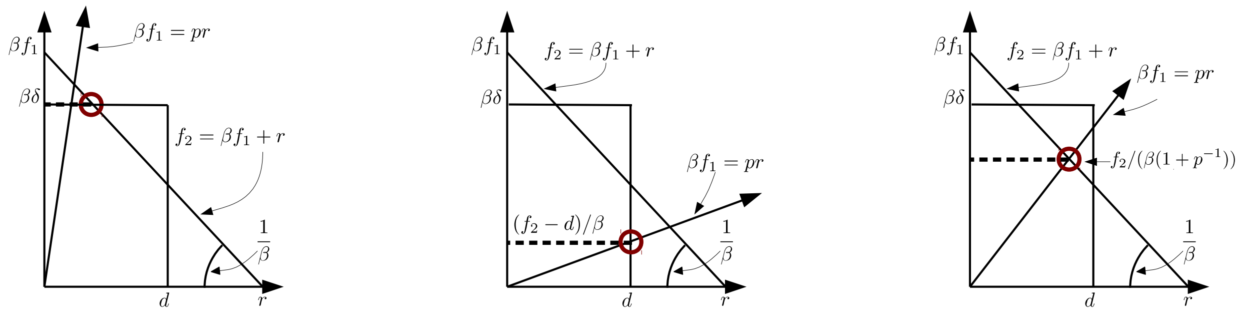
where $\rho_{i,0}$ is the initial condition on the mainline links I_i and l_0 is the initial number of vehicles in R_1.

Riemann Solver for PDE-ODE Model

We assume for our applications that the fundamental diagram has a trapezoidal form as depicted in Fig. 2.6, where v is the *free-flow* speed of traffic and w is referred to as the *congestion wave* speed.

There are many potential Riemann solvers that satisfy the properties required in Section 2.1.2. To guarantee a unique solution for each Riemann datum, we add two modeling decisions to solve the junction. Let ρ_1^+ and ρ_2^- be the densities on I_1 and I_2 (respective) adjacent to the junction. Let l be the queue length on R_1. Then let $\hat{\rho}_1^+$, $\hat{\rho}_2^-$ be the resulting Riemann solutions for I_1 and I_2, while $\hat{r}$ is the resulting Riemann flux from R_1. The additional modeling decisions are then:

1. The flux solution maximizes the outgoing mainline flux $f\left(\hat{\rho}_1^+\right)$
2. When (1) does not give a unique solution, the Riemann solver attempts to satisfy $f\left(\hat{\rho}_2^-\right) = pf\left(\hat{\rho}_1^+\right)$, where $p \in \mathbb{R}_+$ is a merging parameter. The p parameter sets the



(a) Case 1: Priority violated due to limited upstream mainline demand entering downstream mainline.

(b) Case 2: Priority violated due to limited on ramp demand entering downstream mainline.

(c) Case 3: Priority rule satisfied due to sufficient demand from both mainline and on ramp.

Figure 2.7: Godunov junction flux solution for freeway model. The rectangular region represents the feasible flux values for I_1 ($\beta\delta$) and R_1 (d) as determined by the upstream demand, while the line with slope $\frac{1}{\beta}$ represents feasible flux values as determined by mass balance. The βf_1 term accounts for only the flux out of I_1 that stays on the mainline. The flux solution, represented by the red circle, is the point on the feasible region that minimizes the distance from the priority line $f_1 = pr$.

priority of flow from I_1 over the flow from R_1 when there is limited capacity. Since (1) permits multiple flux solutions at the junction, (2) is necessary to obtain a unique solution.

With the necessary restrictions on the Riemann solver in place, we outline the solution method for the PDE-ODE junction problem. The well-posedness and self-similarity proofs are given in [27]. The method closely follows that of general LWR network solutions presented in [42].

For a Riemann datum of $\left(\rho_1^+, \rho_2^-, l\right)$, we introduce the following intermediate variables:

- $\delta = \min\left(F^{\max}, v\rho_1^+\right)$, the maximum allowable flux out of I_1.
- $d = \begin{cases} F^{\max} & \text{if } l > 0 \\ \min(F^{\max}, D(t)) & \text{if } l = 0 \end{cases}$, the maximum allowable flux out of R_1
- $\sigma = \min\left(F^{\max}, w\left(\rho^{\max} - \rho_2^-\right)\right)$, the maximum allowable flux into I_2.

The maximal flux into I_2 is computed as $f_2 = \min(\beta\delta + d, \sigma)$, the minimum between the upstream *demand*, and the downstream *supply*.

To compute the flux leaving I_1, we refer to Figure 2.7. The balance between the fluxes βf_1 (resp. r) entering I_2 from I_1 (resp. R_1) must minimize the deviation from the equation $\beta f_1 = pr$. Since flow must be conserved across the junction, we also have the constraint that the $(\beta f_1, r)$ flows must sum to f_2, and thus the resultant flow pair (f_1, r) must lie on the line $f_2 = \beta f_1 + r$, depicted in Figure 2.7. This results in three distinct cases for the f_1 solution.

- In Case 1, strict satisfaction of the priority line would lead to an f_1 value greater than δ when at the intersection with the supply line $f_2 = \beta f_1 + r$. Since δ is the maximum allowable flux from I_1, we can feasibly exactly satisfy the priority. Thus to minimize the deviation from the priority line, we select $f_1 = \delta$.

- In Case 2, the priority dictates a flux from R_1 in excess of d. To minimize deviation from priority, we select $r = d$, and $\beta f_1 = f_2 - r$.

- In Case 3, strict satisfaction of the priority line gives a feasible f_1 and r solution, and thus we have $f_1 = \frac{f_2}{\beta(1+p^{-1})}$.

Once we have determined f_1 and f_2, then flux balance across the junction dictates that $r = f_2 - \beta f_1$.

To satisfy the Riemann solver condition that only waves that travel outward from the junction may be created, we devise a mapping from the resultant mainline fluxes (f_1, f_2) to the Riemann solver densities $\left(\hat{\rho}_1^+, \hat{\rho}_2^-\right)$. The following conditions uniquely determine $\left(\hat{\rho}_1^+, \hat{\rho}_2^-\right)$:

$$\hat{\rho}_1^+ \in \begin{cases} \left\{\rho_1^+\right\} \cup]\tau(\rho_1^+), \rho^{\max}] & \text{if } 0 \leq \rho_1^+ \leq \rho^{\mathrm{cr}}, \\ [\rho^{\mathrm{cr}}, \rho^{\max}] & \text{if } \rho^{\mathrm{cr}} \leq \rho_1^+ \leq \rho^{\max}; \end{cases} \qquad f\left(\hat{\rho}_1^+\right) = f_1 \tag{2.17}$$

$$\hat{\rho}_2^- \in \begin{cases} [0, \rho^{\mathrm{cr}}] & \text{if } 0 \leq \rho_2^- \leq \rho^{\mathrm{cr}}, \\ \left\{\rho_2^-\right\} \cup [0, \tau(\rho_2^-)] & \text{if } \rho^{\mathrm{cr}} \leq \rho_2^- \leq \rho^{\max}; \end{cases} \qquad f\left(\hat{\rho}_2^-\right) = f_2, \tag{2.18}$$

where τ satisfies the following:

1. $f(\tau(\rho)) = f(\rho)$
2. $\tau(\rho) \neq \rho$

2.2.3 Discrete Freeway Model

The previous section derives a continuous traffic model based on the principle of mass conservation and matching the empirical flux-density relationship. Furthermore, the model possesses strong boundary conditions, allowing for the total flux through the network to be independent of any varying control parameters.

In order to develop computationally efficient optimization and control techniques, we work in the discrete time and space domain. As detailed in Section 2.1.3, we use the Godunov discretization technique.

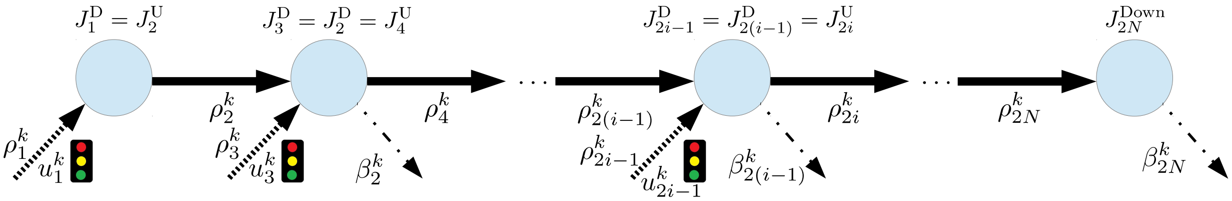


Figure 2.8: Freeway network model. For a junction $j_{2l-1}^{\mathrm{D}} = j_{2(l-1)}^{\mathrm{D}} = j_{2l}^{\mathrm{U}}$ at time-step $k \in \{0, \ldots, T-1\}$, the upstream mainline density are represented by $\rho_{2(l-1)}^k$, the downstream mainline density by ρ_{2l}^k, the on ramp density by ρ_{2l-1}^k, and the off-ramp split ratio by $\beta_{2(l-1)}^k$.

We now consider a freeway network with multiple junctions, as opposed to the presentation of the continuous model, which only considered a single junction.

Consider a freeway section with links $\mathcal{L} = \{1, \ldots, 2N\}$ with a linear sequence of mainline links $= \{2, 4, \ldots, 2N\}$ and connecting on ramp links $= \{1, 3, \ldots, 2N-1\}$. At discrete time $t = k\Delta t, 0 \leq k \leq T-1$, mainline link $2l \in \mathcal{L}, i \in \{1, \ldots, N\}$ has a downstream junction $j_{2l}^{\mathrm{D}} = j_{2(l+1)}^{\mathrm{U}}$ and an upstream junction $j_{2l}^{\mathrm{U}} = j_{2(l-1)}^{\mathrm{D}}$, while on ramp $2l-1 \in \mathcal{L}, i \in \{1, \ldots, N\}$ has a downstream junction $j_{2l-1}^{\mathrm{D}} = j_{2l}^{\mathrm{U}} = j_{2(l-1)}^{\mathrm{D}}$ and an upstream junction j_{2l-1}^{U}.

The off-ramp directly downstream of link $2l, i \in \{1, \ldots, N\}$ has, at time-step k, a split ratio β_{2l}^k Each link $l \in \mathcal{L}$ has a discretized state value $\rho_l^k \in \mathbb{R}$ at each time-step $k \in \{0, \ldots, T-1\}$, that represents the density of vehicles on the link. These values are depicted in Fig 2.8. Junctions that have no on ramps can be effectively represented by adding an on ramp with no demand while junctions with no off-ramps can be represented by setting the split ratio to 1.

As control input which is used extensively in applications in proceeding sections, an on ramp $2l-1 \in \mathcal{L}, l \in \{1, \ldots, N\}$ at time-step $k \in \{0, \ldots, T-1\}$ has a metering rate $u_{2l-1}^k \in [0,1]$ which limits the flux of vehicles leaving the on ramp. Intuitively, the metering rate acts as a fractional decrease in the flow leaving the on ramp and entering the mainline freeway. The domain of the metering control is to force the control to neither impose negative flows nor send more vehicles than present in a queue. Its mathematical model is expressed in (2.25).

For notational simplicity we define the set of densities of links incident to $j_{2l}^{\mathrm{U}} = j_{2(l-1)}^{\mathrm{D}}$ at time-step k as $\vec{\rho}_{j_{2l}^{\mathrm{U}}}^{\,k} = \left\{\rho_{2(l-1)}^k, \rho_{2i-1}^k, \rho_{2l}^k\right\}$. For $k \in \{1, \ldots, T-1\}$, the mainline density ρ_{2l}^k using the Godunov scheme from (2.8) is given by:

$$
\begin{aligned}
h^k_{2l}(\vec{\rho},\vec{u}) = \quad & \rho^k_{2l} - \rho^{k-1}_{2l} \quad + \frac{\Delta t}{L_{2l}} \left(g^G_{j^{\mathrm{D}}_{2l}} \left(\vec{\rho}^{\,k-1}_{j^{\mathrm{D}}_{2l}}, u^{k-1}_{2l+1} \right) \right)_{2l} \qquad (2.19) \\
& \qquad\qquad\quad - \frac{\Delta t}{L_{2l}} \left(g^G_{j^{\mathrm{U}}_{2l}} \left(\vec{\rho}^{\,k-1}_{j^{\mathrm{U}}_{2l}}, u^{k-1}_{2l-1} \right) \right)_{2l} \\
= \quad & \rho^k_{2l} - \rho^{k-1}_{2l} \quad + \frac{\Delta t}{L_{2l}} \left(g^{k-1}_{2l,\mathrm{D}} - g^{k-1}_{2l,\mathrm{U}} \right) = 0 \qquad (2.20)
\end{aligned}
$$

where we have introduced some substitutions to reduce the notational burden of this section: $g^k_{l,\mathrm{D}}$ is the Godunov flux at time-step k exiting a link l at the downstream boundary of the link, and $g^k_{l,\mathrm{U}}$ is the Godunov flux entering the link at the upstream boundary.

We also make the assumption that on ramps have infinite capacity and a free-flow velocity $v_{2l-1} = \frac{L_{2l-1}}{\Delta t}$ to prevent the ramp congestion from blocking demand from ever entering the network. Since the on ramp has no physical length, the length is chosen arbitrarily and the "virtual" velocity chosen above is chosen to replicate the dynamics in [27]. We can then simplify the on ramp update equation to be:

$$
\begin{aligned}
h^k_{2l-1}(\vec{\rho},\vec{u}) \quad &= \quad \rho^k_{2l-1} - \rho^{k-1}_{2l-1} - \frac{\Delta t}{L_{2l-1}} \left(\left(g^G_{j^{\mathrm{U}}_{2l}} \left(\vec{\rho}^{\,k-1}_{j^{\mathrm{U}}_{2l}}, u^{k-1}_{2l-1} \right) \right)_{2l-1} - D^{k-1}_{2l-1} \right) \qquad (2.21) \\
&= \quad \rho^k_{2l-1} - \rho^{k-1}_{2l-1} - \frac{\Delta t}{L_{2l-1}} \left(g^{k-1}_{2l-1,\mathrm{D}} - D^{k-1}_{2l-1} \right) = 0 \qquad (2.22)
\end{aligned}
$$

where D^{k-1}_{2l-1} is the on ramp *flux* demand, and the same notational simplification has been used for the downstream flux. This formulation results in "strong" boundary conditions at the on ramps which guarantees all demand enters the network.

The on ramp model in (2.21) differs from [27] in that we model the on ramp as a discretized PDE with an infinite critical density, while [27] models the on ramp as an ODE "buffer". While both models implement strong boundary conditions, the discretized PDE model makes the freeway network more aligned with the PDE network framework presented in this section.

Discrete Model Equations The following systems of equations give the flux solution of the Riemann solver at time-step $k \in \{1, \ldots, T-1\}$ and junction j^{U}_{2l} for $l \in \{1, \ldots, N\}$:

$$\delta^k_{2(l-1)} = \min\bigl(v_{2(i-1)}\rho^k_{2(l-1)}, F^{\max}_{2(l-1)}\bigr) \tag{2.23}$$

$$\sigma^k_{2l} = \min\bigl(w_{2i}\bigl(\rho^{\max}_{2i} - \rho^k_{2l}\bigr), F^{\max}_{2i}\bigr) \tag{2.24}$$

$$d^k_{2l-1} = u^k_{2l-1}\min\left(\frac{L_{2l-1}}{\Delta t}\rho^k_{2l-1}, F^{\max}_{2i-1}\right) \tag{2.25}$$

$$g^k_{2l,\mathrm{U}} = \min\bigl(\beta^k_{2(l-1)}\delta^k_{2(l-1)} + d^k_{2l-1}, \sigma^k_{2l}\bigr) \tag{2.26}$$

$$g^k_{2(l-1),\mathrm{D}} = \begin{cases} \delta^k_{2(l-1)} & \frac{p_{2(l-1)}g^k_{2l,\mathrm{U}}}{\beta^k_{2(l-1)}\left(1+p_{2(l-1)}\right)} \geq \delta^k_{2(l-1)} \text{[Case 1]} \\ \frac{g^k_{2l,\mathrm{U}}-d^k_{2l-1}}{\beta^k_{2(l-1)}} & \frac{g^k_{2l,\mathrm{U}}}{1+p_{2(l-1)}} \geq d^k_{2l-1} \qquad \text{[Case 2]} \\ \frac{p_{2(l-1)}g^k_{2l,\mathrm{U}}}{\left(1+p_{2(l-1)}\right)\beta^k_{2(l-1)}} & \text{otherwise} \qquad\qquad \text{[Case 3]} \end{cases} \tag{2.27}$$

$$g^k_{2l-1,\mathrm{D}} = g^k_{2l,\mathrm{U}} - \beta^k_{2(l-1)}g^k_{2(l-1),\mathrm{D}} \tag{2.28}$$

where, for notational simplicity, at the edges of of the range for l, any undefined state values (e.g. ρ^k_0) are assumed to be zero by convention.

Note that the equations can be solved sequentially via forward substitution. Also, we do not include the flux result for off-ramps explicitly here since its value has no bearing on further calculations, and we will henceforth ignore its calculation. To demonstrate that indeed the flux solution satisfies the flux conservation property, the off-ramp flux is trivially determined to be $\beta^k_{2(l-1)}g^k_{2(l-1),\mathrm{D}}$.

Chapter 3

Centralized and Decentralized Optimal Freeway Control via the Discrete Adjoint Method

In this chapter, we propose a discrete adjoint approach to compute optimal ramp-metering strategies on road networks modeled by conservation laws. Networks of one-dimensional conservation laws, described by systems of nonlinear first-order hyperbolic *partial differential equations* (PDEs), are an efficient framework for modeling physical phenomena, such as freeway traffic evolution [42, 128, 37] and supply chains [14]. Similarly, PDE systems of balance laws are useful in modeling gas pipeline flow [54, 108] and water channels [53, 98]. Optimization and control of these networks is an active field of research [55, 7, 69]. More generally, numerous techniques exist for the control of conservation laws, such as, for example, backstepping [22, 48], Lyapunov-based methods [22], and optimal control methods [9, 67, 61]. In particular, a common approach is to compute the gradient of the cost functional via the *adjoint method* [45, 62, 99]. Nevertheless, its implementation in the framework of nonlinear conservation laws presents several difficulties linked to the discontinuous character of the solutions. In particular, the presence of shocks in the solutions requires a careful sensitivity analysis based on the use of shift-differentials and generalized tangent vectors, see [13, 122, 123].

Extensive study has also been conducted on the choice of method for effectively computing the gradient via the adjoint. In particular, the continuous adjoint method [60, 55, 79, 105] operates directly on the PDE and a so-called adjoint PDE system, which when solved can be used to obtain an explicit expression of the gradient of the underlying optimization problem. Conversely, the discrete adjoint method [45, 55, 69] first discretizes a continuous-time PDE and then requires the solution of a set of linear equations to solve for the gradient. Finally, a third approach exists, which uses automatic differentiation techniques to automatically generate an adjoint solver from the numerical representation of the forward system [81, 43].

It is well known that the numerical treatment of the adjoint method imposes a careful choice of the discretization scheme to avoid the introduction of numerical errors at discontinuities [25]. Theoretical convergence results for optimization problems have been provided for Lax-Friedichs type schemes [46] and relaxation methods [5]. The case of road networks in free flow conditions is addressed in [55]. In our more general setting of PDE networks and applications to freeway traffic control, the presence of junction conditions with both forward and backward-moving shockwaves led us the use of a modified Godunov scheme that precisely takes into account the flows at the network nodes. An alternative approach involves using Lax-Friedichs-type discretization with higher-resolution interpolation schemes [85]. Moreover, general existence and stability results for the corresponding system of equations modeling traffic evolution on the network are still missing at the moment. Therefore, establishing rigorous convergence results for the gradient computation in this framework is out of the scope of this thesis. Here we made the choice of the discrete adjoint approach, which derives the gradient directly from the discretized system, thus avoiding dealing with weak boundary conditions in the continuous system [42, 128, 117].

There exist many applications of the adjoint method for control, optimization and estimation of physical systems in engineering. Shape optimization of aircraft [105, 44, 79] has applied the method effectively to reduce the computational cost in gradient methods associated with the large number of optimization parameters. The technique has also been applied in parameter identification of biological systems [99]. State estimation problems can be phrased as optimal control problems by setting the unknown state variables as control parameters and penalizing errors in resulting state predictions from known values. This approach has been applied to such problems as open water state estimation [16, 118] and freeway traffic state estimation [59].

Since conservation laws may be nonlinear by nature and lead to non-convex or nonlinear formulations of the corresponding optimization problem, fewer efficient optimization techniques exist for the discretized version of these problems than for convex problems for example. One approach is to approximate the system with a “relaxed” version in order to use efficient linear programming techniques. In transportation, by relaxing the Godunov discretization scheme, the linearization approach was used in [50] for optimal ramp metering, and in [130] for optimal route assignment which is exact when the relaxation gap can be shown to be zero. The ramp metering technique in [84] uses an additional control parameter (variable speed limits) to mimic the linearized freeway dynamics. While the upside of these methods is reduced computational complexity and the guarantee of finding a globally optimal solution, the downside is that the model of the linearized physical system may greatly differ from the actual system to which the control policies would be applied.

Another approach avoids discretization of the continuous system by taking advantage of certain simplifying assumptions in the dynamics. In [40], the problem of finding optimal split ratios on a traffic networks is efficiently solved by deriving non-linear and linear algebraic formulations of a simplified form of the continuous system dynamics which only considers forward-moving shockwaves. In [25], a *mixed-integer linear program* (MILP) formulation is posed for the optimal routing of goods on a supply chain, leading to efficient solutions of this particular application. The number of integer constraints needed in the MILP formulation is proportional to the number of non-linear constraints in the underlying system and has a direct impact on the efficiency of MILP solvers. Applications to non-linear and non-smooth systems such as freeway traffic may prefer non-linear programming approaches such as the adjoint method using non-linear discretization techniques, which avoid integer constraints and allow the constraints to capture more complex dynamics.

Alternatively, nonlinear optimization techniques can be applied to the discretized system without any modification to the underlying dynamics. This approach leads to more expensive optimization algorithms, such as gradient descent, and no guarantee of finding a global optimum. One difficulty in this approach comes in the computation of the gradient, which, if using finite differences, requires a full forward-simulation for each perturbation of a control parameter. This approach is taken in [39, 38] to compute several types of decentralized ramp metering strategies. The increased complexity of the finite differences approach for each additional control parameter makes the method unsuitable for real-time application on moderately-sized freeway networks.

Ramp metering is a common freeway control strategy, providing a means of dynamically controlling freeway throughput without directly impeding mainline flow or implementing complex tolling systems. While metering strategies have been developed using microscopic models [8], most strategies are based off macroscopic state parameters, such as vehicle density and the density's relation to speed [106, 75, 24]. Reactive metering strategies [93, 95, 66] use feedback from freeway loop detectors to target a desired mainline density, while predictive metering strategies [38, 69, 50, 19] use a physical model with predicted boundary flow data to generate policies over a finite time horizon. Predictive methods are often embedded within a model predictive control loop to handle uncertainties in the boundary data and cumulative model errors [84].

This section develops a framework for efficient control of discretized conservation law PDE networks using the adjoint method [45, 96] via Godunov discretization [49], while detailing its application to coordinated ramp metering on freeway networks. Note that the method can be extended without significant difficulty to other numerical schemes commonly used to discretize hyperbolic PDEs. We show how the complexity of the gradient computation in nonlinear optimal control problems can be greatly decreased by using the discrete adjoint method and exploiting the decoupling nature of the problem's network structure, leading to efficient gradient computation methods. After giving a general framework for computing the gradient over the class of scalar conservation law networks, we show that the system's partial derivatives have a sparsity structure resulting in gradient computation times linear in the number of state and control variables for networks of small vertex degree. Memory usage is also linear when sparse data structures are utilized. The results are demonstrated by running a coordinated ramp metering strategy on a 19 mile freeway stretch in California faster than real-time (i.e. the computational time is faster than physical time), while giving traffic performance superior to that of state of the art practitioners tools.

3.1 Discrete Adjoint Derivation for Networked Conservation Laws

Section 2.1 developed a method for discretizing networked, scalar conservation laws. Once a Riemann solver is selected for the junction behavior, the matter of discretization via Godunov's method becomes routine. This section uses Godunov's method to construct an efficient and general method for gradient computations of control objectives constrained by networked conservation law systems.

3.1.1 State, Control, and Governing Equations

We now focus on controlling systems of the form in Equation (3.1.1):

$$\rho_l^{k+1} = \rho_l^k - \frac{\Delta t}{\Delta x}\left(\left(g^G_{j_l^{\mathrm{D}}}\left(\vec{\rho}^{\,k}_{j_l^{\mathrm{D}}}\right)\right)_l - \left(g^G_{j_l^{\mathrm{U}}}\left(\vec{\rho}^{\,k}_{j_l^{\mathrm{U}}}\right)\right)_l\right),$$

in which some parts of the state can be controlled directly (for example, in the form of boundary control). We wish to solve the system in Equation (3.1.1) T time-steps forward, i.e. we wish to determine the discrete state values ρ_l^k for all links $l \in \mathcal{L}$ and all time-steps $k \in \{0, \ldots, T-1\}$. Furthermore, at each time-step k, we assume a set of "control" variables $\left(u_1^k, \ldots, u_M^k\right) \in \mathbb{R}^M$ that influence the solution of the Riemann problems at junctions, where M is the number of controlled values at each time-step, and each control may be updated at each time-step. We assume that a control may only influence a subset of junctions, which is a reasonable assumption if the controls have some spatial locality. Thus, for a junction $j \in \mathcal{J}$, we assume without loss of generality that a subset of the control parameters $\left(u^k_{j_j^1}, \ldots, u^k_{j_j^{M_j}}\right) \in \mathbb{R}^{M_j}$ influence the solution of the Riemann solver. Similar to the notation developed for state variables, for control variables, we define $\vec{u}_j^{\,k} := \left(u^k_{j_j^1}, \ldots, u^k_{j_j^{M_j}}\right)$ as the concatenation of the control variables around the junction j. To account for the addition of controls, we modify the Riemann problem at a junction $j \in \mathcal{J}$ at time-step k to be a function of the current state of connecting links $\vec{\rho}_j^{\,k}$, and the current control parameters $\vec{u}_j^{\,k}$. Then using the same notation as before, we express the Riemann solver as:

$$\begin{aligned} RS_j : \quad \mathbb{R}^{n_j+m_j} \times \mathbb{R}^{M_j} \quad &\to \mathbb{R}^{n_j+m_j} \\ \left(\vec{\rho}_j^{\,k}, \vec{u}_j^{\,k}\right) \quad &\mapsto RS_j\left(\vec{\rho}_j^{\,k}, \vec{u}_j^{\,k}\right) = \hat{\vec{\rho}}_j^{\,k}. \end{aligned}$$

We represent the entire state of the solved system with the vector $\vec{\rho} \in \mathbb{R}^{NT}$, where for $l \in \mathcal{L}$ and $k \in \{0, \ldots, T-1\}$, we have $\vec{\rho}_{Nk+l} = \rho_l^k$. Similarly, we represent the entire control vector by $\vec{u} \in \mathbb{R}^{MT}$, where $\vec{u}_{Mk+j} = u_j^k$.

For each state variable ρ_l^k, write the corresponding update equation h_l^k:

$$\begin{aligned} h_l^k :\ & \mathbb{R}^{NT} \times \mathbb{R}^{MT} && \to \mathbb{R} \\ & (\vec{\rho}, \vec{u}) && \mapsto h_l^k(\vec{\rho}, \vec{u}) = 0. \end{aligned}$$

This takes the following form:

$$h_l^0(\vec{\rho}, \vec{u}) = \rho_l^0 - \bar{\rho}_l \quad = 0 \tag{3.1}$$

$$\begin{aligned} h_l^k(\vec{\rho}, \vec{u}) = \rho_l^k - \rho_l^{k-1} &+ \frac{\Delta t}{L_l} f\left(RS_{j_l^{\mathrm{D}}}\left(\vec{\rho}^{\,k-1}_{j_l^{\mathrm{D}}}, \vec{u}^{\,k-1}_{j_l^{\mathrm{D}}}\right)\right)_l \\ &- \frac{\Delta t}{L_l} f\left(RS_{j_l^{\mathrm{U}}}\left(\vec{\rho}^{\,k-1}_{j_l^{\mathrm{U}}}, \vec{u}^{\,k-1}_{j_l^{\mathrm{U}}}\right)\right)_l \quad = 0 \quad \forall k \in \{2, \ldots, T-1\}, \end{aligned} \tag{3.2}$$

or in terms of the Godunov junction flux:

$$\begin{aligned} h_l^k(\vec{\rho}, \vec{u}) = \quad \rho_l^k - \rho_l^{k-1} \quad &+ \frac{\Delta t}{\Delta x}\left(g^G_{j_l^{\mathrm{D}}}\left(\vec{\rho}^{\,k}_{j_l^{\mathrm{D}}}, \vec{u}^{\,k-1}_{j_l^{\mathrm{D}}}\right)\right)_l \\ &- \frac{\Delta t}{\Delta x}\left(g^G_{j_l^{\mathrm{U}}}\left(\vec{\rho}^{\,k}_{j_l^{\mathrm{U}}}, \vec{u}^{\,k-1}_{j_l^{\mathrm{U}}}\right)\right)_l \end{aligned} \tag{3.3}$$

for all links $l \in \mathcal{L}$, where $\bar{\rho}_l$ is the initial condition for link l. Thus, we can construct a system of NT governing equations $H(\vec{\rho}, \vec{u}) = 0$, where the $h_{l,k}$ is the equation in H at index $Nk + l$, identical to the ordering of the corresponding discrete state variable.

Optimal Control Problem Formulation In addition to our governing equations $H(\vec{\rho}, \vec{u}) = 0$, where we assume each $h_i^k \in \mathcal{C}^1$, we also introduce a cost function $C \in \mathcal{C}^1$.

$$\begin{aligned} C :\ & \mathbb{R}^{NT} \times \mathbb{R}^{MT} && \to \mathbb{R} \\ & (\vec{\rho}, \vec{u}) && \mapsto C(\vec{\rho}, \vec{u}) \end{aligned}$$

which returns a scalar that serves as a metric of performance of the state and control values of the system. We wish to minimize the quantity C over the set of control parameters $\vec{u}$, while constraining the state of the system to satisfy the governing equations $H(\vec{\rho}, \vec{u}) = 0$, which is, again, the concatenated version of (3.2) or (3.3). We summarize this with the following optimization problem:

$$\begin{aligned} \min_{\vec{u}} \quad & C(\vec{\rho}, \vec{u}) \\ \text{subject to:} \quad & H(\vec{\rho}, \vec{u}) = 0 \end{aligned} \tag{3.4}$$

Both the cost function and governing equations may be non-convex in this problem.

3.1.2 Discrete Adjoint Method

We wish to use gradient information in order to find control values $\vec{u}^*$ that give locally optimal costs $C^* = C(\vec{\rho}(\vec{u}^*), \vec{u}^*)$. Since there may exist many local minima for this optimization problem (3.4) (which is non-convex in general), gradient methods do not guarantee global optimality of $\vec{u}^*$. Still, nonlinear optimization methods such as interior point optimization utilize gradient information to improve performance [125].

In a descent algorithm, the optimization procedure will have to descend a cost function, by coupling the gradient, which, at a nominal point $(\vec{\rho}', \vec{u}')$ is given by:

$$d_{\vec{u}} C(\vec{\rho}', \vec{u}') = \left.\frac{\partial C(\vec{\rho}, \vec{u})}{\partial \vec{\rho}}\right|_{\vec{\rho}', \vec{u}'} \frac{d\vec{\rho}}{d\vec{u}} + \left.\frac{\partial C(\vec{\rho}, \vec{u})}{\partial \vec{u}}\right|_{\vec{\rho}', \vec{u}'}. \tag{3.5}$$

Note. *For Equation* (3.5) *to be valid, all required partial and full derivatives must be well-defined, including $\frac{d\vec{\rho}}{d\vec{u}}$. In some applications, this assumption does not necessarily hold, either because f itself is not smooth or because g^G is not smooth (and thus $H \notin \mathcal{C}^1$), as is the case for the LWR equation with concave fundamental diagrams. There are several settings in which the conditions for differentiability are satisfied, see in particular [55, 36].*

The main difficulty is to compute the term $\frac{d\vec{\rho}}{d\vec{u}}$. We take advantage of the fact that the derivative of $H(\vec{\rho}, \vec{u})$ with respect to $\vec{u}$ is equal to zero along trajectories of the system:

$$d_{\vec{u}} H(\vec{\rho}', \vec{u}') = \left.\frac{\partial H(\vec{\rho}, \vec{u})}{\partial \vec{\rho}}\right|_{\vec{\rho}', \vec{u}'} \frac{d\vec{\rho}}{d\vec{u}} + \left.\frac{\partial H(\vec{\rho}, \vec{u})}{\partial \vec{u}}\right|_{\vec{\rho}', \vec{u}'} = 0. \tag{3.6}$$

The partial derivative terms, $H_{\vec{\rho}} \in \mathbb{R}^{NT \times NT}$, $H_{\vec{u}} \in \mathbb{R}^{NT \times MT}$, $C_{\vec{\rho}} \in \mathbb{R}^{NT}$, and $C_{\vec{u}} \in \mathbb{R}^{MT}$, can all be evaluated (more details provided in Section 3.1.3) and then treated as constant matrices. Thus, in order to evaluate $d_{\vec{u}} C(\vec{\rho}', \vec{u}') \in \mathbb{R}^{MT}$, we must solve a coupled system of matrix equations.

Forward system. If we solve for $\frac{d\vec{\rho}}{d\vec{u}} \in \mathbb{R}^{NT \times MT}$ in (3.6), which we call the *forward system*:

$$H_{\vec{\rho}} \frac{d\vec{\rho}}{d\vec{u}} = -H_{\vec{u}},$$

then we can substitute the solved value for $\frac{d\vec{\rho}}{d\vec{u}}$ into (3.5) to obtain the full expression for the gradient. Section 3.1.3 below gives details on the invertibility of $H_{\vec{\rho}}$, guaranteeing a solution for $\frac{d\vec{\rho}}{d\vec{u}}$.

Adjoint system. Instead of evaluating $\frac{d\vec{\rho}}{d\vec{u}}$ directly, the adjoint method solves the following system, called the adjoint system, for a new unknown variable $\lambda \in \mathbb{R}^{NT}$ (called the adjoint variable):

$$H_{\vec{\rho}}^T \lambda = -C_{\vec{\rho}}^T \tag{3.7}$$

Under certain additional conditions on the flux function and discretization scheme, the adjoint system in Equation (3.7) may be shown to converge to the continuous adjoint system as the discretization steps go towards zero, as described in the following works [5, 55, 123]. No such convergence results exist in our setting of using a Godunov discretization with general $n \times m$ junctions.

The expression for the gradient becomes:

$$d_{\vec{u}} C(\vec{\rho}', \vec{u}') = \lambda^T H_{\vec{u}} + C_{\vec{u}} \tag{3.8}$$

We note that Equations (3.7) and (3.8) can be alternatively derived using the first-order *Karush-Kuhn-Tucker* (KKT) conditions, coupled with the constraint qualification in Equation (3.4). Given the assumed non-convexity of the underlying system, first-order KKT conditions are necessary, but not sufficient conditions for optimality of $\tilde{\mathbf{u}}$ and λ. For practical applications to non-convex systems and for the purposes of this thesis, we do not necessarily seek global *or local* optimality, but rather the direction of steepest descent given in Equation (3.8) in order to *improve* the performance of the system.

We define $D_{\vec{\rho}}$ to be the maximum junction degree on the network:

$$D_{\vec{\rho}} = \max_{j \in \mathcal{J}} (n_j + m_j), \tag{3.9}$$

and also define $D_{\vec{u}}$ to be the maximum number of constraints that a single control variable appears in, which is equivalent to:

$$D_{\vec{u}} = \max_{u \in \vec{u}} \sum_{j \in \mathcal{J} : u \in \vec{u}_j^k} (n_j + m_j). \tag{3.10}$$

Note that $\left\{ u \in \vec{u}_j^k : j \in \mathcal{J} \right\}$ is a k-dependent set. By convention, junctions are either actuated or not, so there is no dependency on k, i.e. if $\exists k$ s.t. $u \in \vec{u}_j^k$, then $\forall k$, $u \in \vec{u}_j^k$.

Using these definitions, we show later in Section 3.1.4 how the complexity of computing the gradient is reduced from $O(D_{\vec{\rho}} NMT^2)$ to $O(T(D_{\vec{\rho}} N + D_{\vec{u}} M))$ by considering the adjoint method over the forward method.

A graphical depiction of $D_{\vec{\rho}}$ and $D_{\vec{u}}$ are given in Fig. 3.1. Freeway networks are usually considered to have topologies that are nearly planar, leading to junctions degrees which typically do not exceed 3 or 4, regardless of the total number of links. Also, from the locality argument for control variables in Section (3.1.1), a single control variable's influence over state variables will not grow with the size of the network. Since the $D_{\vec{\rho}}$ and $D_{\vec{u}}$ typically do not grow with NT or MT for freeway networks, the complexity of evaluating the gradient for such networks can be considered linear for the adjoint method.

3.1.3 Evaluating the Partial Derivatives

While no assumptions are made about the sparsity of the cost function C, the networked-structure of the PDE system and the Godunov discretization scheme allows us to say more

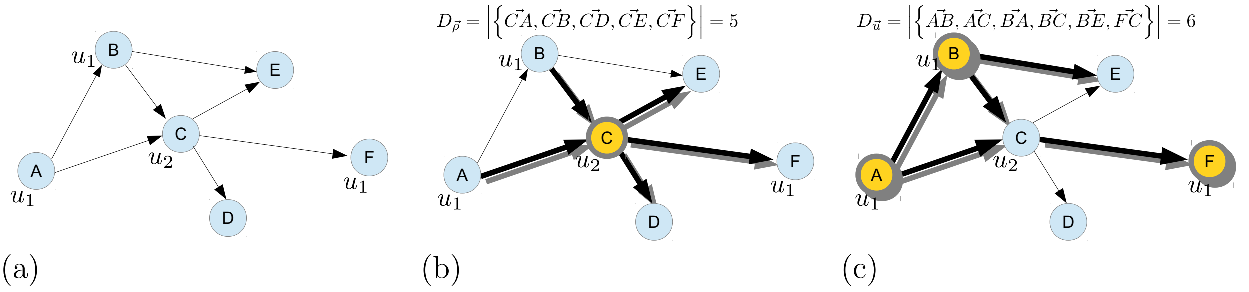

$D_{\vec{\rho}} = \left|\left\{\vec{CA}, \vec{CB}, \vec{CD}, \vec{CE}, \vec{CF}\right\}\right| = 5$
$D_{\vec{u}} = \left|\left\{\vec{AB}, \vec{AC}, \vec{BA}, \vec{BC}, \vec{BE}, \vec{FC}\right\}\right| = 6$
A
B
C
D
E
F
u_1
u_2
(a)
(b)
(c)

Figure 3.1: Depiction of $D_{\vec{\rho}}$ and D_v for an arbitrary graph. Fig. 3.1a shows the underlying graphical structure for an arbitrary PDE network. Some control parameter u_1 has influence over junctions A, B, and F, while another control parameter u_2 has influence over only junction C. Fig. 3.1b depicts the center junction having the largest number of connecting edges, thus giving $D_{\vec{\rho}} = 5$. Fig. 3.1c shows that control parameter u_1 influences three junctions with sum of junctions degrees equal to six, which is maximal over the other control parameter u_2. leading to the result $D_{\vec{u}} = 6$. Note that in Fig. 3.1c, the link going from junction A to junction B is counted twice: once as an outgoing link $\vec{AB}$ and once as in incoming link $\vec{BA}$.

about the structure and sparsity of $H_{\vec{\rho}}$ and $H_{\vec{u}}$.

Partial derivative expressions. Given that the governing equations require the evaluation of a Riemann solver at each step, we detail some of the necessary computational steps in evaluating the $H_{\vec{\rho}}$ and $H_{\vec{u}}$ matrices.

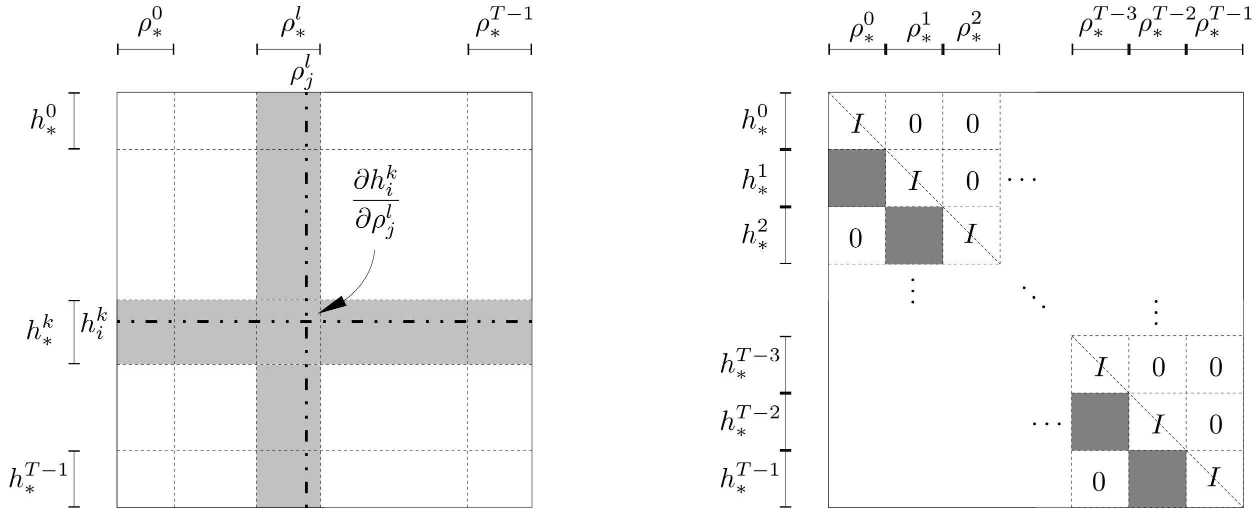
If we consider a particular governing equation $h_l^k(\vec{\rho}, \vec{u}) = 0$, then we may determine the partial term with respect to $\rho_j^l \in \vec{\rho}$ by applying the chain rule:

$$\begin{aligned}\frac{\partial h_l^k}{\partial \rho_j^l} = \frac{\partial \rho_l^k}{\partial \rho_j^l} - \frac{\partial \rho_l^{k-1}}{\partial \rho_j^l} &+ \frac{\Delta t}{L_i} f'\left(RS_{j_l^{\mathrm{D}}}\left(\vec{\rho}_{j_l^{\mathrm{D}}}^{k-1}, \vec{u}_{j_l^{\mathrm{D}}}^{k-1}\right)_l\right) \frac{\partial}{\partial \rho_j^l}\left(RS_{j_l^{\mathrm{D}}}\left(\vec{\rho}_{j_l^{\mathrm{D}}}^{k-1}, \vec{u}_{j_l^{\mathrm{D}}}^{k-1}\right)_l\right) \\ &- \frac{\Delta t}{L_i} f'\left(RS_{j_l^{\mathrm{U}}}\left(\vec{\rho}_{j_l^{\mathrm{U}}}^{k-1}, \vec{u}_{j_l^{\mathrm{U}}}^{k-1}\right)_l\right) \frac{\partial}{\partial \rho_j^l}\left(RS_{j_l^{\mathrm{U}}}\left(\vec{\rho}_{j_l^{\mathrm{U}}}^{k-1}, \vec{u}_{j_l^{\mathrm{U}}}^{k-1}\right)_l\right)\end{aligned} \tag{3.11}$$

or if we consider the composed Riemann flux solver g_j^G in (2.10):

$$\frac{\partial h_l^k}{\partial \rho_j^l} = \frac{\partial \rho_l^k}{\partial \rho_j^l} - \frac{\partial \rho_l^{k-1}}{\partial \rho_j^l} + \frac{\Delta t}{L_i}\left(\frac{\partial}{\partial \rho_j^l}\left(g_{j_l^{\mathrm{D}}}^G\left(\vec{\rho}_{j_l^{\mathrm{D}}}^{k-1}, \vec{u}_{j_l^{\mathrm{D}}}^{k-1}\right)\right)_l - \frac{\partial}{\partial \rho_j^l}\left(g_{j_l^{\mathrm{U}}}^G\left(\vec{\rho}_{j_l^{\mathrm{U}}}^{k-1}, \vec{u}_{j_l^{\mathrm{U}}}^{k-1}\right)\right)_l\right) \tag{3.12}$$

A diagram of the structure of the $H_{\vec{\rho}}$ matrix is given in Fig. (3.2a). Similarly for $H_{\vec{u}}$, we take a control parameter $u_j^l \in \vec{u}$, and derive the expression:



(a) Ordering of the partial derivative terms. Constraints and state variables are clustered first by time, and then by cell index.

(b) Sparsity structure of the $H_{\vec{\rho}}$ matrix. Besides the diagonal blocks, which are identity matrices, blocks where $l \neq k-1$ are zero.

Figure 3.2: Structure of the $H_{\vec{\rho}}$ matrix.

$$\begin{aligned}\frac{\partial h_l^k}{\partial u_j^l} = &+ \frac{\Delta t}{L_i} f'\left(RS_{j_l^{\mathrm{D}}}\left(\vec{\rho}_{j_l^{\mathrm{D}}}^{k-1}, \vec{u}_{j_l^{\mathrm{D}}}^{k-1}\right)_l\right) \frac{\partial}{\partial u_j^l}\left(RS_{j_l^{\mathrm{D}}}\left(\vec{\rho}_{j_l^{\mathrm{D}}}^{k-1}, \vec{u}_{j_l^{\mathrm{D}}}^{k-1}\right)_l\right) \\ &- \frac{\Delta t}{L_i} f'\left(RS_{j_l^{\mathrm{U}}}\left(\vec{\rho}_{j_l^{\mathrm{U}}}^{k-1}, \vec{u}_{j_l^{\mathrm{U}}}^{k-1}\right)_l\right) \frac{\partial}{\partial u_j^l}\left(RS_{j_l^{\mathrm{U}}}\left(\vec{\rho}_{j_l^{\mathrm{U}}}^{k-1}, \vec{u}_{j_l^{\mathrm{U}}}^{k-1}\right)_l\right)\end{aligned} \tag{3.13}$$

or for the composed Godunov junction flux solver g_j^G:

$$\frac{\partial h_l^k}{\partial u_j^l} = \frac{\Delta t}{L_i}\left(\frac{\partial}{\partial u_j^l}\left(g_{j_l^{\mathrm{D}}}^G\left(\vec{\rho}_{j_l^{\mathrm{D}}}^{k-1}, \vec{u}_{j_l^{\mathrm{D}}}^{k-1}\right)\right)_l - \frac{\partial}{\partial u_j^l}\left(g_{j_l^{\mathrm{U}}}^G\left(\vec{\rho}_{j_l^{\mathrm{U}}}^{k-1}, \vec{u}_{j_l^{\mathrm{U}}}^{k-1}\right)\right)_l\right). \tag{3.14}$$

Analyzing (3.11), the only partial terms that are not trivial to compute are $\frac{\partial}{\partial \rho_j^l}\left(RS_{j_l^{\mathrm{D}}}\left(\vec{\rho}_{j_l^{\mathrm{D}}}^{k-1}, \vec{u}_{j_l^{\mathrm{D}}}^{k-1}\right)_l\right)$ and $\frac{\partial}{\partial \rho_j^l}\left(RS_{j_l^{\mathrm{U}}}\left(\vec{\rho}_{j_l^{\mathrm{U}}}^{k-1}, \vec{u}_{j_l^{\mathrm{U}}}^{k-1}\right)_l\right)$. Similarly for (3.13), the only nontrivial terms are $\frac{\partial}{\partial u_j^l}\left(RS_{j_l^{\mathrm{D}}}\left(\vec{\rho}_{j_l^{\mathrm{D}}}^{k-1}, \vec{u}_{j_l^{\mathrm{D}}}^{k-1}\right)_l\right)$ and $\frac{\partial}{\partial u_j^l}\left(RS_{j_l^{\mathrm{U}}}\left(\vec{\rho}_{j_l^{\mathrm{U}}}^{k-1}, \vec{u}_{j_l^{\mathrm{U}}}^{k-1}\right)_l\right)$. Once one obtains the solutions to these partial terms, then one can construct the full $H_{\vec{\rho}}$ and $H_{\vec{u}}$ matrices and use (3.7) and (3.8) to obtain the gradient value.

As these expressions are written for a general scalar conservation law, the only steps in computing the gradient that are specific to a particular conservation law and Riemann solver are computing the derivative of the flux function f and the partial derivative terms just discussed. These expressions are explicitly calculated for the problem of optimal ramp metering in Section (3.2.1).

3.1.4 Complexity Analysis of Discrete Adjoint for Sparse Networks

This section demonstrates the following proposition:

Proposition 3.1. *The total complexity for the adjoint method on a scalar hyperbolic network of PDEs is* $O(T(D_{\vec{\rho}}N + D_{\vec{u}}M))$.

We can show the lower-triangular structure and invertibility of $H_{\vec{\rho}}$ by examining (3.1) and (3.2). For $k \in \{1, \ldots, T-1\}$, we have that h_l^k is only a function of ρ_l^k and of the state variables from the previous time-step $k-1$. Thus, based on our ordering scheme in Section 3.1.1 of ordering variables by increasing time-step and ordering constraints by corresponding variable, we know that the diagonal terms of $H_{\vec{\rho}}$ are always 1 and all upper-triangular terms must be zero (since those terms correspond to constraints with a dependence of *future* values). These two conditions demonstrate both that $H_{\vec{\rho}}$ is lower-triangular and is invertible due to the ones along the diagonal.

Additionally, if we consider taking partial derivatives with respect to the variable ρ_j^l, then we can deduce from Equation (3.2) that all partial terms will be zero except for the diagonal term, and those terms involving constraints at time $j+1$ with links connecting to the downstream and upstream junctions j_j^{D} and j_j^{U} respectively. To summarize, $H_{\vec{\rho}}$ matrices for systems described in Section 3.1.1 will be square, invertible, lower-triangular and each column will have a maximum cardinality equal to $D_{\vec{\rho}}$ in (3.9). The sparsity structure of $H_{\vec{\rho}}$ is depicted in Fig. 3.2b.

Using the same line of argument for the maximum cardinality of $H_{\vec{\rho}}$, we can bound the maximum cardinality of each column of $H_{\vec{u}}$. Taking a single control variable u_j^l, the variable can only appear in the constraints at time-step $j+1$ that correspond to a link that connects to a junction j such that $u_j^l \in \vec{u}_j^{l+1}$. These conditions give us the expression for $D_{\vec{u}}$ in (3.10), or the maximum cardinality over all columns in $H_{\vec{u}}$.

If we only consider the lower triangular form of $H_{\vec{\rho}}$, then the complexity of solving for the gradient using the forward system is $O((NT)^2 MT)$, where the dominating term comes from solving (3.5), which requires the solution of MT separate $NT \times NT$ lower-triangular systems. The lower-triangular system allows for forward substitution, which can be solved in $O((NT)^2)$ steps, giving the overall complexity $O((NT)^2 MT)$. The complexity of computing the gradient via the adjoint method is $O((NT)^2 + (NT)(MT))$, which is certainly more efficient than the forward-method, as long as $MT > 1$. The efficiency is gained by considering that (3.7) only requires the solution of a single $NT \times NT$ *upper*-triangular system (via backward-substitution), followed by the multiplication of $\lambda^T H_v$, an $NT \times NT$ and an $NT \times MT$ matrix in (3.8), with a complexity of $O((NT)^2 + (NT)(MT))$.

For the adjoint method, this complexity can be improved upon by considering the sparsity of the $H_{\vec{\rho}}$ and $H_{\vec{u}}$ matrices, as detailed in Section 3.1.4. For the backward-substitution step, each entry in the λ vector is solved by *at most* $D_{\vec{\rho}}$ multiplications, and thus the complexity of solving (3.7) is reduced to $O(D_{\vec{\rho}}NT)$. Similarly, for the matrix multiplication of $\lambda^T H_v$, while λ is not necessarily sparse, we know that each entry in the resulting vector requires at most $D_{\vec{u}}$ multiplications, giving a complexity of $O(D_{\vec{u}}MT)$. Furthermore, if a sparse implementation of the $H_{\vec{\rho}}$ and $H_{\vec{u}}$ matrices are used, then memory usage will also scale linearly with the number of state and control variables.

3.2 Adjoint-based Model Predictive Control for Coordinated, Predictive Ramp Metering

Problem Statement Including the initial conditions as specified in (3.1) with (2.19) and (2.25) gives a complete description of the system $H(\vec{\rho},\vec{u}) = 0$, $\vec{\rho} \in \mathbb{R}^{2N}$, $\vec{u} \in \mathbb{R}$, where:

$$\begin{aligned} \vec{\rho}_{2Nk+l} &:= & \rho_l^k \quad & 1 \le i \le 2N, 0 \le k \le T-1 \\ \vec{u}_{Nk+l} &:= & u_{2l}^k \quad & 1 \le i \le N, 0 \le k \le T-1 \end{aligned}$$

The objective of the control is to minimize the *total travel time* on the network, expressed by the cost function C:

$$C(\vec{\rho},\vec{u}) = \Delta t \sum_{k=1}^{T} \sum_{i=1}^{2N} L_i \rho_l^k.$$

The optimal coordinated ramp-metering problem can be formulated as an optimization problem with PDE-network constraints:

$$\begin{aligned} \min_{\vec{u}} \quad & C(\vec{\rho},\vec{u}) & & (3.15) \\ \text{subject to:} \quad & H(\vec{\rho},\vec{u}) & = 0 & \\ & 0 \le u \le 1 & \forall u \in \vec{u} & \quad (3.16) \end{aligned}$$

Standard methods exist for the handling of geometric constraints on $\vec{u}$ in descent methods (such as box constraints in Equation (3.16)), such as projection methods [25] and barrier methods [35, 11].

3.2.1 Partial Derivative Calculations for Ramp Metering

To use the adjoint method as described in Section 3.1.2, we need to compute the partial derivative matrices $H_{\vec{\rho}}$, $H_{\vec{u}}$, $C_{\vec{\rho}}$ and $C_{\vec{u}}$. Computing the partial derivatives with respect to the cost function is straight forward:

$$\begin{aligned}
\frac{\partial C}{\partial \rho_l^k} &= \quad \Delta t L_i \qquad 1 \le i \le 2N, 0 \le k \le T-1 \\
\frac{\partial C}{\partial u_{2l}^k} &= \epsilon\Big(\tfrac{1}{1-u_{2l}^k} - \tfrac{1}{u_{2l}^k}\Big) \quad 1 \le i \le N, 0 \le k \le T-1
\end{aligned}$$

To compute the partial derivatives of H, we follow the procedure in Section 3.1.3. For an upstream junction $j_{2l}^{\mathrm{U}} \in \mathcal{J}$ and time-step $k \in \{1, \dots, T-1\}$, we only need to compute the partial derivatives of the flux solver $g_{j_{2l}^{\mathrm{U}}}^{G}\left(\vec{\rho}_{j_{2l}^{\mathrm{U}}}^{k}, u_{2l-1}^{k}\right)$ with respect to the adjacent state variables $\vec{\rho}_{j_l}^k$ and ramp metering control u_l^k. We calculate the partial derivatives of the functions in (2.23)-(2.28) with respect to either a state or control variable$s \in \vec{\rho} \cup \vec{u}$:

$$\begin{aligned}
\frac{\partial \delta_{2(l-1)}^k}{\partial s} &= \begin{cases} v_{2(i-1)} & s = \rho_{2(l-1)}^k, v_i \rho_{2(l-1)}^k \le F_{2(i-1)}^{\max} \\ 0 & \text{otherwise} \end{cases} \\
\frac{\partial \sigma_{2l}^k}{\partial s} &= \begin{cases} -w_{2i} & s = \rho_{2l}^k, w_{2i}\big(\rho_{2i}^{\max} - \rho_{2l}^k\big) \le F_{2i}^{\max} \\ 0 & \text{otherwise} \end{cases} \\
\frac{\partial d}{\partial s} &= \begin{cases} u_{2l-1}^k & s = \rho_{2l-1}^k, \rho_{2l-1}^k \le F_{2l-1}^{\max} \\ \min\big(\rho_{2l-1}^k, F_{2i-1}^{\max}\big) & s = u_{2l-1}^k \\ 0 & \text{otherwise} \end{cases} \\
\frac{\partial}{\partial s} g_{2l,\mathrm{U}}^k &= \begin{cases} \beta_{2(l-1)}^k \frac{\partial \delta_{2(l-1)}^k}{\partial s} + \frac{\partial d_{2(l-1)}^k}{\partial s} & \beta_{2(l-1)}^k \delta_{2(l-1)}^k + d_{2l-1}^k \le \sigma_{2l}^k \\ \frac{\partial \sigma_{2l}^k}{\partial s} & \text{otherwise} \end{cases} \\
\frac{\partial}{\partial s} g_{2(l-1),\mathrm{D}} &= \begin{cases} \frac{\partial \delta_{2(l-1)}^k}{\partial s} & \frac{g_{2l,\mathrm{U}}^k p_{2(l-1)}}{1+p_{2(l-1)}} \ge \frac{\delta_{2(l-1)}^k}{\beta_{2(l-1)}^k} \\ \frac{1}{\beta_{2(l-1)}^k}\Big(\frac{\partial}{\partial s} g_{2l,\mathrm{U}}^k - \frac{\partial d_{2l-1}^k}{\partial s}\Big) & \frac{g_{2l,\mathrm{U}}^k}{1+p_{2(l-1)}} \ge d_{2(l-1)}^k \\ \frac{p_{2(l-1)}}{\beta_{2(l-1)}^k\big(1+p_{2(l-1)}\big)} \frac{\partial}{\partial s} g_{2l,\mathrm{U}}^k & \text{otherwise} \end{cases} \\
\frac{\partial}{\partial s} g_{2l-1,\mathrm{D}} &= \frac{\partial}{\partial s} g_{2l,\mathrm{U}}^k - \beta_{2(l-1)}^k \frac{\partial}{\partial s} g_{2(l-1),\mathrm{D}}
\end{aligned}$$

These expressions fully specify the partial derivative values needed in (3.12) and (3.14). Thus we can construct the $H_{\vec{\rho}}$ and $H_{\vec{u}}$ matrices. With these matrices and $C_{\vec{\rho}}$ and $C_{\vec{u}}$, we can solve for the adjoint variable $\lambda \in \mathbb{R}^{2NT}$ in (3.7) and substitute its value into (3.8) to obtain the gradient of the cost function C with respect to the control parameter $\vec{u}$.

3.2.2 Model Predictive Control Overview

There are a number of underlying assumptions that permit finite-horizon optimal control techniques to be useful.

- The mathematical dynamics accurately models the actual physical system being controlled.

- The current state of the system is known.

- The future state of the system can be accurately predicted as a function of the applied control.

Under an idealized simulation, these conditions are met by assuming perfect knowledge of initial and boundary conditions, and validating performance against using forward simulation identical to the controller's assumed model. In application, the above conditions will not be met exactly due to model inaccuracy, sensor noise, and prediction error. Thus, when applying control policies generated from optimal control procedures in noisy environments, one would expect the future physical state to diverge from that predicted from the controller's model, and thus a degradation in the performance of the controller.

This stems from optimal control being an *open loop* control method, where the error in the state estimation is not observed as the physical system evolves. Alternatively, *closed loop* controllers [93, 95] incorporate the state estimation at frequent intervals (e.g. less than one minute for freeway systems), and choose a control policy which is optimal for only the *next* time-step. Such schemes are also referred to as *reactive* schemes, which react to real-time conditions rather than attempt to anticipate future road conditions (predictive schemes).

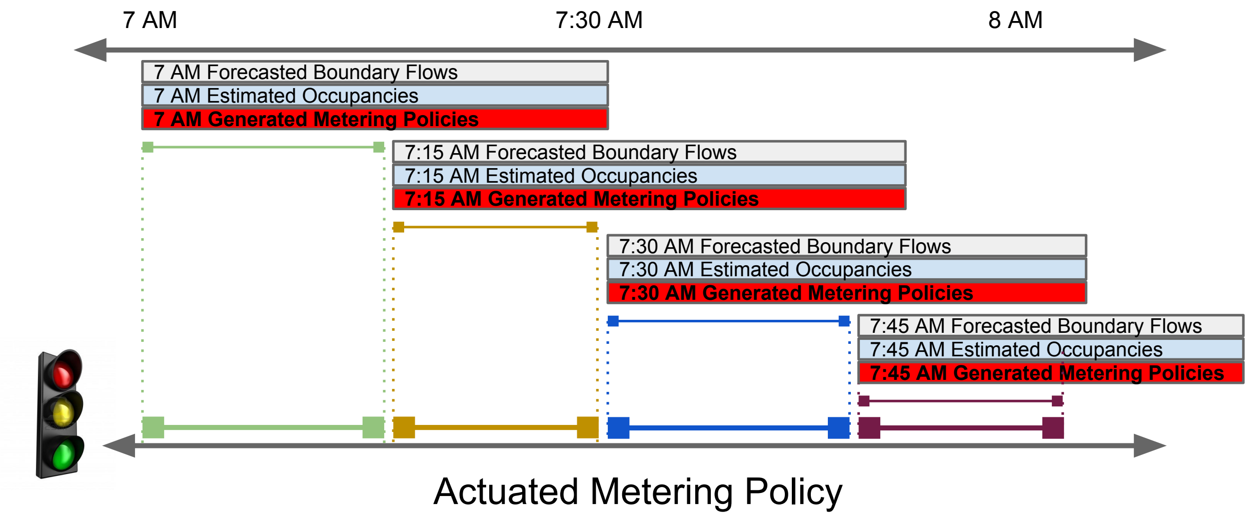

7 AM
7:30 AM
8 AM
7 AM Forecasted Boundary Flows
7 AM Estimated Occupancies
7 AM Generated Metering Policies
7:15 AM Forecasted Boundary Flows
7:15 AM Estimated Occupancies
7:15 AM Generated Metering Policies
7:30 AM Forecasted Boundary Flows
7:30 AM Estimated Occupancies
7:30 AM Generated Metering Policies
7:45 AM Forecasted Boundary Flows
7:45 AM Estimated Occupancies
7:45 AM Generated Metering Policies
Actuated Metering Policy

Figure 3.3: Diagram of rolling-horizon MPC. At 15 minute intervals ($T_{\text{update}} = 15$ minutes), the MPC controller requires an estimate of the current traffic (i.e. initial conditions) as well as predictions over the next 30 minutes ($T_{\text{horizon}} = 30$ minutes) for the future vehicle demands on the on ramps (i.e. boundary conditions).

Model predictive control (MPC) is a control technique which leverages the predictive benefits of optimal control approaches without the drawbacks of open loop control. At time instances occurring with an update period T_{update}, the MPC controller constructs an optimal control problem for the time period between the current time t and some future time $t + T_{\text{horizon}}$, where T_{horizon} is typically much larger than T_{update} in order to properly leverage the predictive nature of optimal control. A new control policy is produced for the time period of the current optimal control problem, and the new control policy is then applied to the physical system. After T_{update} time has elapsed, an updated control policy will be generated which leverages the newest available initial and boundary conditions. Given that $T_{\text{update}} < T_{\text{horizon}}$, the updated policy will be generated before the previous policy is completely applied, at which point the old control policy is discarded in favor of the new policy.

This process is summarized in Figure 3.3. An MPC-based controller was implemented within the Connected Corridors system, leveraging the adjoint framework inside the optimal control problem. Numerical results with respect to adjoint control are given in Sections 3.2.3 and 3.3.6.

3.2.3 Numerical Results

To demonstrate the effectiveness of using the adjoint ramp metering method to compute gradients, we implemented the algorithm on practical scenarios with field experimental data. The algorithm can then be used as a gradient computation subroutine inside any descent-method optimization solver that takes advantage of first-order gradient information. Our implementation makes use of the open-source *IpOpt* solver [125], an interior point, nonlinear program optimizer. To serve as comparisons, two other case scenarios were run:

1. No control: the metering rate is set to 1 on all on-ramps at all times.

2. Alinea [93]: a well-adopted, feedback-based ramp metering algorithm commonly used in the practitioner's community. Alinea is computationally efficient and decentralized, making it a popular choice for large networks, but does not take estimated boundary flow data as input. Since Alinea has a number of tuning parameters, we perform a *modified* grid-search technique over the different parameters that scales linearly with the number of on-ramps, and select the best-performing parameters, in order to be fair to this framework. A *full* grid-search approach scales exponentially with the number of on-ramps, rendering it infeasible for moderate-size freeway networks.

All simulations were run on a 2012 commercial laptop with 8 GB of RAM and a dual-core 1.8 GHz Intel Core i5 processor.

Note. *To demonstrate the reduced running time associated with the adjoint approach, we also implemented a gradient descent using a finite differences approach similar to [38, 39], which requires an $O(T^2NM)$ computation for each step in gradient descent, but it proved to be computationally infeasible for even small, synthetic networks. Running ramp metering on even a network of 4 links over 6 time-steps for 5 gradient steps took well over 4 minutes,* *rendering the method useless for real-time applications. The comparison of running times of finite differences versus the adjoint method is given in Fig. 3.4. Due to the impractically large running times associated with finite differences, we do not consider the finite differences in further results, which only becomes worse as the problem scales to larger networks and time horizons.*

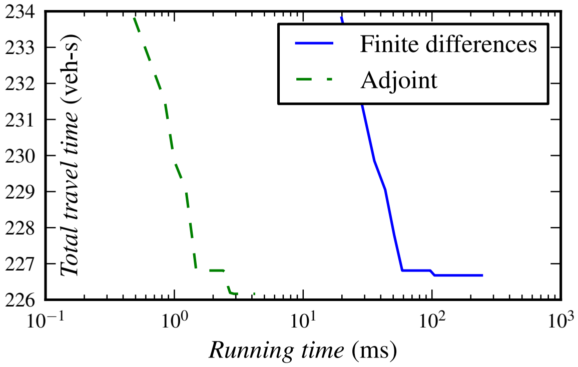


Figure 3.4: Running time of ramp metering algorithm using IpOpt with and without gradient information. Network consists of 4 links and 6 time-steps with synthetic boundary flux data. The method using gradient information via the adjoint method converged well before the completion of the *first* step of the finite differences descent method.

Implementation of I15S in San Diego

As input into the optimization problem, we constructed a model of a 19.4 mile stretch of the I15 South freeway in San Diego, California between San Marcos and Mira Mesa. The network has $N = 125$ links, and $M = 9$ on-ramps, with boundary data specified for $T = 1800$ time-steps, for a time horizon of 120 minutes given Δt =4 seconds. The network is shown in Fig. 3.5.

Link length data was obtained using the Scenario Editor software developed as part of the *Connected Corridors* project. Fundamental diagram parameters, split ratios, and boundary data were also obtained using calibration techniques developed by *Connected Corridors*.

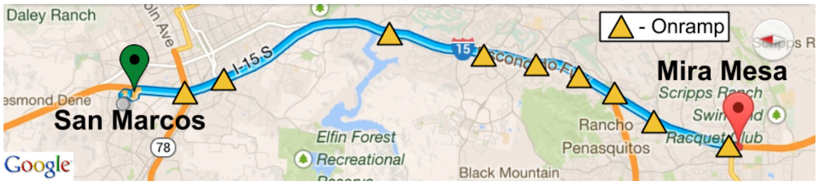
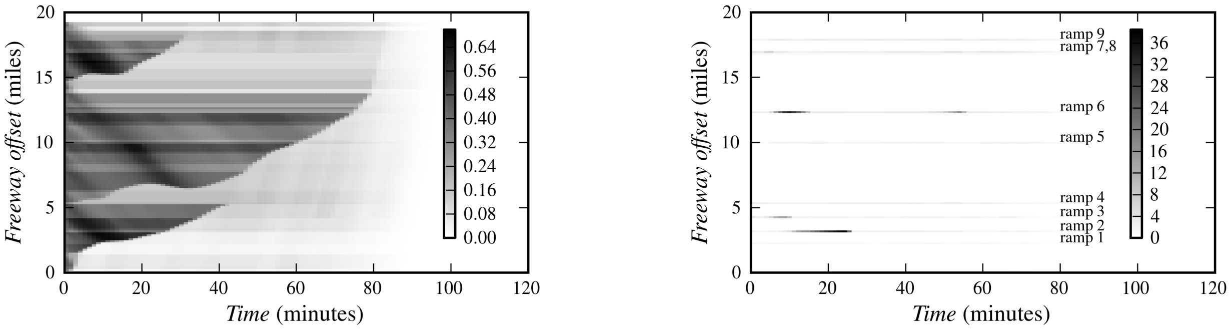


Figure 3.5: Model of section of I15 South in San Diego, California. The freeway section spanning 19.4 miles was split into 125 links with 9 on-ramps.



(a) Density profile. The units are the ratio of a link's vehicle density to a link's jam density.

(b) On-ramp queue profile in units of vehicles.

Figure 3.6: Density and queue profile of no-control freeway simulation. In the first 80 minutes, congestion pockets form on the freeway and queues form on the on-ramps, then eventually clear out before 120 minutes.

Densities resulting in free-flow speeds were chosen as initial conditions on the mainline and on-ramps. The data used in calibration was taken from PeMS sensor data [18] during a morning rush hour period, scaled to generate congested conditions. The input data was chosen to demonstrate the effectiveness of the adjoint ramp metering method in a real-world setting. A profile of the mainline and on-ramps during a forward-simulation of the network is shown in Fig. 3.6 under the described boundary conditions.

Finite-Horizon Optimal Control

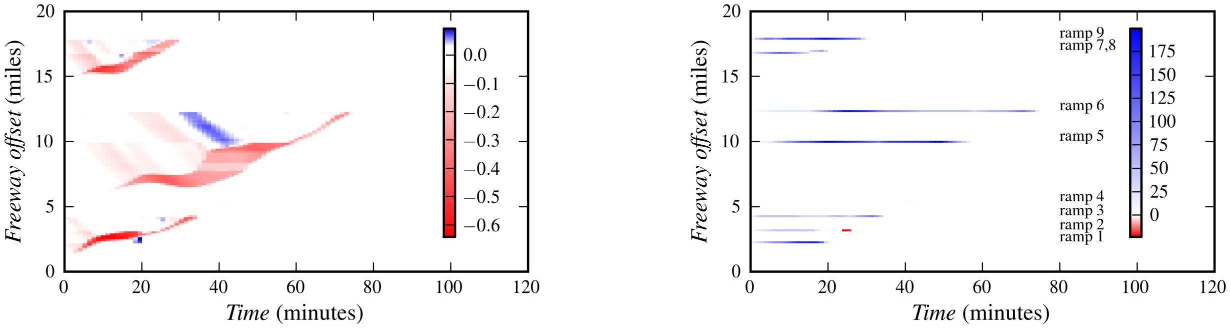
Experimental Setup. The adjoint ramp metering algorithm is compared to the reactive Alinea scheme, for which we assume that perfect boundary conditions and initial conditions are available. The metric we use to compare the different strategies is *reduced-congestion* percentage, $\bar{c} \in (-\infty, 100]$, which we define as:

$$\bar{c} = 100\left(1 - \frac{c_{\mathrm{c}}}{c_{\mathrm{nc}}}\right)$$

where $c_{\mathrm{c}}, c_{\mathrm{nc}} \in \mathbb{R}_+$ are the *congestion* resulting from the *control* and *no-control* scenarios, respectively. We use the metric for congestion as defined in [115]; for a given section of road S and time horizon T, the congestion is given as

$$c(S,T) = \sum_{(s\in S, \tau \in T)} \max\left[\mathrm{TTT}(s,\tau) - \frac{\mathrm{VMT}(s,\tau)}{v_s}, 0\right]$$

where v_s is the free-flow velocity, VMT is total vehicle miles traveled, and TTT is total travel time over the link s and time-step τ. Since it is infeasible to compute the global optimum for all cases, a reduced congestion of 100% serves as an upper bound on the possible amount of improvement.



(a) Density difference profile in units of *change in density* from the control scenario to the no control scenario over the jam density of the link.

(b) Queue difference profile in units of vehicles.

Figure 3.7: Profile differences for mainline densities and on-ramp queues. Evidenced by the mainly negative differences in the mainline densities and the mainly positive differences in the on-ramp queue lengths, the adjoint ramp metering algorithm effectively limits on-ramp flows in order to reduce mainly congestion. *Best viewed in color.*

Results. Fig. 3.7 shows a difference profile for both density and queue lengths between the no control simulation and the simulation applying the ramp metering policy generated from the adjoint method. Negative differences in Figs. 3.7a and 3.7b indicate where the adjoint method resulted in fewer vehicles for the specific link and time-step. The adjoint method was successful in appropriately deciding which ramps should be metered in order to improve throughput for the mainline.

Running time analysis shows that the adjoint method can produce beneficial results in real-time applications. Fig. 3.8 details the improvement of the adjoint method as a function of the overall running time of the algorithm. After just a few gradient steps, the adjoint method outperforms the Alinea method. Given that the time horizon of two hours is longer than the period of time one can expect reasonably accurate boundary flow estimates, more practical simulations with shorter time horizons should permit more gradient steps in a real-time setting.

While the adjoint method leads to queues with a considerable number of cars in some on-ramps, this can be addressed by introducing barrier terms into the cost function that limit the maximum queue length. The Alinea method tested for the I15 network had no prescribed maximum queue lengths as well, but was not able to produce significant improvements in total travel time reduction, while the adjoint method was more successful.

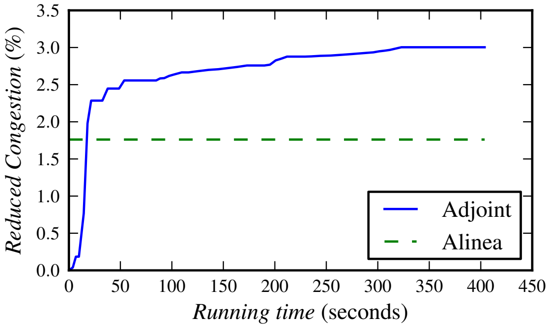


Figure 3.8: Reduced congestion versus simulation time for freeway network. The results indicate that the algorithm can run with performance better than Alinea if given an update time of less than a minute.

Model Predictive Control

To study the performance of the algorithm under noisy input data, we embed both our adjoint ramp metering algorithm and the Alinea algorithm inside of a *model predictive control* (MPC) loop.

Experimental Setup. The MPC loop begins at a time t by estimating the initial conditions of the traffic on the freeway network and the predicted boundary fluxes over a certain time horizon T_h. These values are noisy, as exact estimation of these parameters is not possible on real freeway networks. The estimated conditions are then passed to the ramp metering algorithm to compute an optimal control policy over the T_h time period. The system is then forward-simulated over an update period of $T_u \leq T_h$, using the exact initial conditions and boundary conditions, as opposed to the noisy data used to compute control parameters. The state of the system and boundary conditions at $t + T_u$ are then estimated (with noise) and the process is repeated.

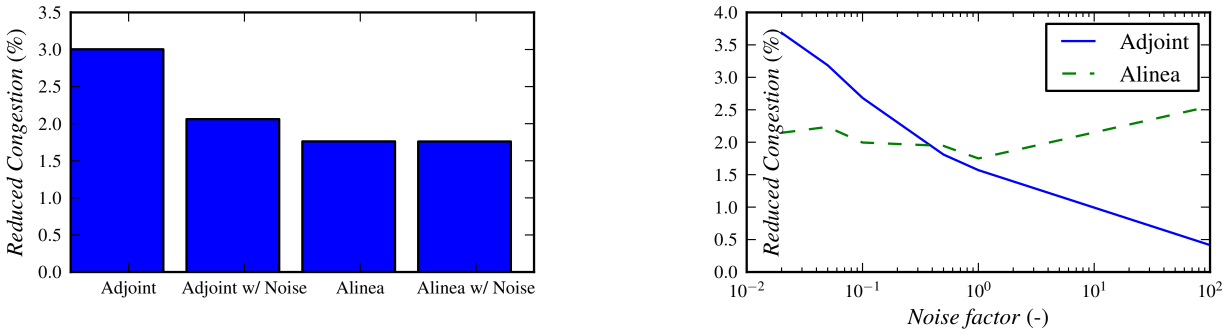
A non-negative *noise factor*, $\sigma \in \mathbb{R}_+$, is used to study how the adjoint method and Alinea perform as the quality of estimated data decreases. If ρ is the actual density for a cell and time-step, then the density $\bar{\rho}$ passed to the control schemes is given by:

$$\bar{\rho} = \rho \cdot (1 + \sigma \cdot R)$$

where R is a uniformly distributed random variable with mean 0 and domain $[-0.5, 0.5]$. The noise factor was applied to both initial and boundary conditions.

Two different experiments were conducted:

1. **Real-time I15 South**: MPC is run for the I15 South network with $T_h = 80$ minutes and $T_u = 26$ minutes. A noise factor of 2% was chosen for the initial and boundary conditions. The number of iterations was chosen in order to ensure that each MPC iteration finished in the predetermined update time T_u.



(a) Reduced congestion.

(b) Reduced congestion with increasing sensor noise for network with synthetic data.

Figure 3.9: Summary of model predictive control simulations. The results indicate that the adjoint method has superior performance for moderate noise levels on the initial and boundary conditions.

2. **Noise Robustness**: MPC is for over a synthetic network with length 12 miles and boundary conditions over 75 minutes. The experiments are run over a profile of noise factors between 1% and 8000%.

Results. **Real-Time I15 South.** The results are summarized in Fig. 3.9a. The adjoint method applied once to the entire horizon with perfect boundary and initial condition information serves as a baseline performance for the other simulations, which had noisy input data and limited knowledge of predicted boundary conditions. The adjoint method still performs well under the more realistic conditions of the MPC loop with noise, resulting in 2% reduced congestion or 40 car-hours in relation to no control, as compared to the 3% reduced (60 car-hours) congestion achieved by the adjoint method with no noise and full time horizon ($T_h = T$). In comparison, the Alinea method was only able to achieve 1.5% reduced congestion (30 car-hours) for both the noisy and no-noise scenarios. The results indicate that, under a realistic assumption of a 2% noise factor in the sensor information, the algorithm's ability to consider boundary conditions results in an improvement upon strictly reactive policies, such as Alinea.

Robustness to Noise. Simulation results on the synthetic network with varying levels of noise are shown in Fig. 3.9b. The adjoint method is able to outperform the Alinea method when the noise level is less than 80%, a reasonable assumption for data provided by well-maintained loop detectors. As the initial and boundary condition data deteriorates, the adjoint method becomes useless. Since Alinea does not rely on boundary data, it is able to produce improvements, even with severely noisy data. The results indicate that the adjoint method will outperform Alinea under reasonable noise levels in the sensor data.

3.2.4 Summary of Key Results

This section has detailed a simple framework for finite-horizon optimal control methods on a network of scalar conservation laws derived from first discretizing the network via the Godunov method, then applying the discrete adjoint to this system. To tailor the framework to a specific application, one need only provide the partial derivatives of the Riemann solver at a network junction as well as the partial derivatives of the objective. Furthermore, we show that for this class of problems, the sparsity pattern allows the problem to be implemented with only linear memory and linear computational complexity with respect to the number of state and control parameters. We demonstrate the scalability of the approach by implementing a coordinated ramp metering algorithm using the adjoint method and applying the algorithm to the I-15 South freeway in California. The algorithm runs in a fraction of real-time and produces significant improvements over existing algorithms. The ramp metering algorithm has been fully implemented within *Connected Corridors* [1] system as a component of the traffic simulator module.

3.3 Decentralized Control of Flow Networks

Finite-horizon optimal control is a popular method for computing predictive control strategies for dynamical systems [104, 7], its applicability growing with the increase of computational power and pervasiveness of physical sensing. In general, a finite-horizon optimal control problem will take the following form:

$$\min_{x \in X} \quad f(s, x) \tag{3.17}$$

$$\text{subject to:} \quad s = g(x) \tag{3.18}$$

where x represents the vector of control variables belonging to the set of feasible controls X (which we may assume to be $\mathbb{R}^n$ for simplicity), s represents the vector of "state" variables, constrained to be a deterministic function $g(x)$ of the control, and f is some objective function of the control and state we wish to minimize.

Related Work in Distributed Optimization Much attention has recently been given to distributed methods for finite-horizon optimal control problems, where g is assumed to be linear and f is assumed to be quadratic or convex. Distributed optimization has been found useful for at least two reasons. Firstly, the parallelizability of the individual sub-problems allows for faster computation time and better overall convergence properties [88, 38, 97, 47]. Secondly, physical systems often have controls physically distributed in space, creating a need for distributed control algorithms which limit the amount of shared information and communication between subsystems [80, 15, 124].

Different assumptions on the structure, smoothness, and convexity of f, X and g leads to different convergence bounds and communication bounds. In optimal control, a method presented in [88] for decoupling the quadratic terms from the nonquadratic terms leads to efficient caching techniques shown to be effective in FPGA applications. A distributed gradient descent-based approach is given in [15], which has $O\left(\frac{1}{\sqrt{k}}\right)$ convergence to the global optimum in the general case, where k is the number of iterations of the algorithm. A common dual-decomposition technique employed for distributed optimal control is the *alternating directions method of multipliers* [41, 10, 88] (ADMM), which has been shown to have $O\left(\frac{1}{k}\right)$ convergence under certain assumptions of the smoothness and decomposability of the objectives [127]. Additionally, an accelerated version of ADMM, based on Nesterov's algorithm [86] can give $O\left(\frac{1}{k^2}\right)$ convergence when the decomposed objectives are smooth [97].

When the coupling between systems takes on some sparse form, then one can devise algorithms with limited communication, which can be beneficial from a latency and architectural standpoint. Optimal control problems where subsystems have disjoint state variables but coupled control variables have been shown to be amenable to decomposition techniques for distributed optimization [47, 15], where [80] shows how ADMM decomposition leads to less communication without a decrease in solution accuracy.

In [80, 47, 15], the subsystems with disjoint state are modeled as agents tasked with optimizing over their own subsystem, where agents which share some control variables are connected by some edge in a communication graph. Thus, the more sparse the coupling of systems, the lesser the communication requirements. Such a model is referred to as *multi-agent optimization* [127]. In systems with coupling due to physical proximity, this consequence has the added benefit of requiring only physically local communication, and removes the need for any centralized controller or hub for communication. In [127], an asynchronous form of ADMM (subsequently referred to as A-ADMM) is presented for multi-agent optimization, which permits agents to update themselves in arbitrary order, with communication only required between neighboring agents. The method in [127] does not present an accelerated version and is shown to have $O\left(\frac{1}{k}\right)$ convergence.

Subsystems with Coupled State One recurring assumption in the distributed optimization literature above is that subsystems have disjoint state variables. For network flow problems, where subsystems correspond to partitions of a network into subnetworks, such an assumption does not hold. To see this, one can imagine a traffic light timing plan causing a traffic jam which spreads across the entire freeway network [104, 84] or a bottleneck of planes in an airspace affecting flight times throughout the air network [7]. As a result, it is not possible to decompose the subsystems by only sharing control parameters without coupling each subsystem to all control variables and modeling the evolution of the entire network within each subsystem.

Yet, freeway traffic and air traffic subsystems have a very sparse coupling in their state variables. For instance, discrete traffic models [24, 27] often assume that the speed of traffic on a particular section of road is only a function of the speed of traffic on neighboring links. Thus, each subnetwork subsystem would only share a small number of control variables and state variables with other subsystems, precisely those which physically share a border with the subsystem.

To exploit the sparsity of such systems, we develop a multi-agent optimization algorithm based on A-ADMM [127] which permits each agent (subsystem) to share both control and state variables with neighboring agents, while still converging to the globally optimal control, given the standard assumption of convex objectives and linear constraints. At a high level, the algorithm “relaxes” the state variables *external* to an agent while constraining *internal* state variables to adhere to the subsystem’s dynamics. Since A-ADMM eventually brings all shared variables between agents into *consensus* (i.e. the difference between shared variables converges to zero), the relaxed external state variables will converge to satisfying the original constraints.

The rest of the section is structured as follows. Section 3.3.1 presents the general problem of posing a multi-agent optimal control problem, with the additional assumption that an agent may share both state and control variables with other agents. The problem is then posed in a form amenable to using the A-ADMM algorithm in Section 3.3.2. A systematic approach to modeling an optimal control problem over a dynamical network as a multi-agent distributed optimization over subnetworks is given in Section 3.3.3, as well as a discussion on the suitability of the method for scaling model predictive control on dynamical networks. In Section 3.3.4, we give an adjoint-based approach to solving the agent's subnetwork optimal control problem, suitable for applications with complex, non-convex dynamics. We then present the application of distributed, predictive ramp-metering and *variable speed limit* (VSL) control on freeway networks in Section 3.3.5 followed by numerical results in Section 3.3.6 with comparisons to existing distributed approaches.

Notation For a vector x, let $x[i]$ be the i'th element of x, and similarly let $y[i,j]$ be the element of the two-dimensional vector in the i'th row and j'th column. If we have a vector x with $card(x) = N$ and let w be a subset of $\{1, \dots, N\}$, then let x_w denote the vector selecting only those elements $x[i]$ where $i \in w$. If a vector d is the concatenation $d = (a, b, c)$, then let $[d]_a$ be the sub-vector of d corresponding to the original element a.

3.3.1 Optimization over Systems with Shared State

We wish to solve an optimization problem with a "free" global variable $x \in \mathbb{R}^n$ and a "dependent" variable $s \in \mathbb{R}^m$ which is a deterministic function of x. We assume there is a partition of s into D disjoint subsets,

$$s = \left(s_{u(1)}, \dots, s_{u(D)}\right),$$

where $u(i)$ are subsets of $\{1, \dots, m\}$. The objective function is assumed to be the sum of D sub-objectives, where sub-objective $f_i, i \in \{1, \dots, D\}$ is a convex function of only variable $s_{u(i)}$. [1] Furthermore, $s_{u(i)}$ is assumed to be a function of some subset of x and s. Explicitly, for each $i \in 1, \dots, D$, there is well-defined, linear function g_i and subsets $v(i)$ and $w(i)$ ($w(i) \cap u(i) = \emptyset$) where

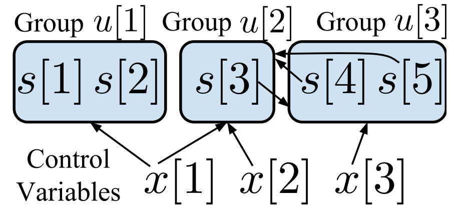
$$s_{u(i)} = g_i\left(\left(x_{v(i)}, s_{w(i)}\right)\right). \tag{3.19}$$

The tuple $\left(x_{v(i)}, s_{w(i)}\right)$ is the concatenation vector of $x_{v(i)}$ and $s_{w(i)}$. We omit the double parenthesis in the rest, for simplicity. One can view $u(i), v(i), w(i)$, as the *internal* state, the control, and the *external* state, respectively, of group i. We can now express the optimization problem we wish to solve as:

$$\min_{x,s} \quad \sum_{i=1}^{D} f_i\left(s_{u(i)}\right) \tag{3.20}$$

$$\text{subject to:} \quad s_{u(i)} = g_i\left(x_{v(i)}, s_{w(i)}\right) \quad \forall i \tag{3.21}$$

[1] We omit the dependency of the objective on the control variable in this presentation for simplicity. It is still easy in this form to add control variables into the objective by duplicating a control variable into the state.



(a) Free and dependent variable coupling diagram.

Group i	$u(i)$	$v(i)$	$w(i)$
$i = 1$	{1,2}	{1}	{}
$i = 2$	{3}	{1,2}	{4,5}
$i = 3$	{4,5}	{3}	{3}

(b) Summary of resultant state, control, and external state subsets.

Edge (i, j)	$v(i) \cap v(j)$	$u(i) \cap w(j)$	w$(i) \cap u(j)$
$i, j = 1, 2$	{1}	{}	{}
$i, j = 2, 3$	{}	{3}	{4,5}
$i, j = 1, 3$	{}	{}	{}

(c) Summary of shared control and state between groups

Figure 3.10: Example of optimization problem partitioned into $D = 3$ disjoint state variable groups with shared control and external state variables. Figure 3.10a shows the partitioned problem, where an arrow depicts a dependency of a partition group on an external state variable or control variable. The arrows allow us to compute the $u(i), v(i), w(i)$ subsets for each group i, where $\rightarrow$ indicates functional dependency through Equation 3.19. Table 3.10b summarizes the construction. The dependency graph (V, E) is computed using the subsets in Table 3.10b, which is summarized in Table 3.10c and reveals that edges exist for groups $(1, 2)$ and $(2, 3)$, but not for $(1, 3)$.

Figure 3.10a shows an example of how different sub-objectives may be coupled and Table 3.10b summarizes how one constructs the $u(i), v(i), w(i)$ subsets from the state and control coupling.

Dependency Graph There are no assumptions on the subsets $v(i)$ and $w(i)$, which implies that the value of each sub-objective f_i is coupled to not just the sub-vector $s_{u(i)}$, but also the global variable x, and other sub-vectors $s_{u(j)}$. We can express this coupling as a dependency graph (V, E), where vertices V are each sub-problem $i \in \{1, \ldots, D\}$ and an edge $(i, j) \in E$ exists whenever

1. $w(i) \cap u(j) \neq \emptyset$ (g_i is a function of some variable in $s_{u(j)}$), **or**

2. $v(i) \cap v(j) \neq \emptyset$ (there is some $x[k]$ which both g_i and g_j depend upon).

Let the neighboring edges of node $i \in V$ be denoted by $E(i)$. A dependency graph construction for the example in Figure 3.10a is summarized in Table 3.10c.

In Section 3.3.2, we devise a distributed algorithm solve Problem (3.20) with the following requirements:

1. Each processing node corresponds to a sub-objective node in the dependency graph.
2. Each node can be updated in parallel.
3. Each node i only exchanges information with its neighbors $E(i)$ in the dependency graph (V, E).
4. The algorithm is asynchronous and decentralized, i.e. no central process is required and nodes can be updated arbitrarily.

3.3.2 Asynchronous-ADMM Algorithm

We reformulate Problem (3.20) to permit a distributed solution method via A-ADMM. For each node $i \in V$, we duplicate the "shared variables" $x_{v(i)}$ and $s_{w(i)}$ as $\bar{x}_i$ and $\bar{s}_i$ respectively, and reformulate Problem (3.20) as:

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{D} f_i\big(s_{u(i)}\big) && (3.22) \\ \text{subject to:} \quad & s_{u(i)} = g_i(\bar{x}_i, \bar{s}_i) \quad \forall i && (3.23) \\ & \bar{s}_i = s_{w(i)} \quad \forall i && (3.24) \\ & \bar{x}_i = x_{v(i)} \quad \forall i && (3.25) \end{aligned}$$

The variable replication allows Constraint (3.23) in Problem (3.22) to be decoupled across nodes. To decouple Constraints (3.24) and (3.25), we follow a modified process from [127].

First, we duplicate each subset $s_{u(i)}$ with a vector s_i local to node $i \in V$, and then concatenate all local variables into a single variable $y_i = (s_i, \bar{x}_i, \bar{s}_i)$, such that y_i is restricted to the space:

$$Y_i = \{(s_i, \bar{x}_i, \bar{s}_i) : s_i = g_i(\bar{x}_i, \bar{s}_i)\}.$$

Finally, we can repose Constraints 2 and 3 in an *edge-wise* fashion as follows. For each edge $e = (i, j) \in E$, let $y_{i,e}$ and $y_{j,e}$ be the sub-vectors of y_i and y_j that are coupled through g_j and g_i, respectively. Then Problem (3.20) becomes:

$$\begin{aligned} \min_{(y_i \in Y_i)_{i \in V}} \quad & \sum_{i=1}^{D} f_i([y_i]_s) && (3.26) \\ \text{subject to:} \quad & y_{i,e} = y_{j,e} \quad \forall e \in E && (3.27) \end{aligned}$$

Algorithm 3 Asynchronous Edge Based ADMM

```
1: while Not Converged do
2:     Select edge $(i,j) \in E$
3:     for $q \in (i,j)$ do
4:         $y_q^{k+1} \leftarrow \arg\min_{y \in Y_q} f_q([y]_s) - \sum_{e \in E(q)} \Lambda_{q,e} \lambda_e^{k,T} \big(y_{q,e} - \bar{y}_e^k\big) + \frac{\psi}{2} \|y_{q,e} - \bar{y}_e^k\|_2^2$
5:     end for
6:     $\lambda_e^{k+1} \leftarrow \lambda_e^{k+1} - \frac{\psi}{2}\big(y_{i,e}^{k+1} - y_{j,e}^{k+1}\big)$
7:     for $q \notin (i,j)$ do
8:         $a^{k+1} \leftarrow a^k$
9:     end for
10: end while
11: Note: $\tilde{y}_e^k = \frac{1}{2}\big(y_{i,e}^k + y_{j,e}^k\big)$
12: Note: $\Lambda_{q,e} = \begin{cases} 1 & q = i \\ -1 & q = j \end{cases} \quad e = (i,j)$
```

By moving the edge constraints into the objective through a standard Lagrange multiplier approach, and adding a regularization term which is equal to zero for feasible solutions [10], we can construct the augmented Lagrangian $\mathcal{L}$ formulation (with tunable augmenting coefficient ψ), and express the optimization problem as:

$$\min_{y=(y_i)_{i \in V}} \max_{\lambda=(\lambda_e)_{e \in E}} \mathcal{L}(y,\lambda) := \sum_{i=1}^{D} f_i([y_i]_s) + \sum_{e \in E} \lambda_e^T (y_{i,e} - y_{j,e}) + \psi \|y_{i,e} - y_{j,e}\|_2^2, \tag{3.28}$$

The above form permits us to apply the A-ADMM algorithm as proposed and analyzed in [127], and shown in Algorithm 3. At a high-level, the algorithm iterates by first randomly selecting an edge $e = (i,j)$ from E. Then, nodes i and j update y_i and y_j respectively by minimizing the Lagrangian in Equation (3.28) in parallel, while holding all other variables $\{\lambda_e'\}_{e' \neq e}, \{y_k\}_{i \notin \{i,j\}}$ constant. The new y_i and y_j values are used to update the dual λ_e variables by applying a dual-ascent method [10]. Finally, the process is repeated *ad-infinitum* by updating a new edge selected from E, until some convergence or termination criteria are reached.

Section 3.3.4 presents an efficient solution method, based on discrete adjoint computations, to solving the subproblem on Line 4 of Algorithm 3.

Remark 3.1. *The equation in Line 4 differs slightly from the augmented Lagrangian in Equation* (3.28) *and is the result of a number of algebraic manipulations, which are explicitly derived in [10, 127].*

Remark 3.2. *We introduce the asymmetric coefficient $\Lambda_{q,e}$ to account for the fact that the terms for edge $e \in E(q)$ in Line 4 depend upon whether the updating problem q was the first or second term (i or j) in the edge pair.*

3.3.3 Distributed Optimization on Coupled Dynamical Systems

Physical transport systems, such as freeway traffic networks [104, 24] or gas pipelines [54] are often naturally expressed as a network of individual dynamical systems which influence one another at contact points, or *junction points.* Given the coupling in dynamics across the entire network, optimizing over partitioned sub-systems, with no communication between systems, will lead to *greedy* solutions over the individual systems and sub-optimal global results [38]. Thus, any distributed, globally optimal control scheme applied to such systems must account for the *shared state* between the systems. We now show how this can be done using the multi-agent A-ADMM approach. Furthermore, we show how the algorithm naturally leads to a communication scheme which mirrors the physical structure of the underlying physical network.

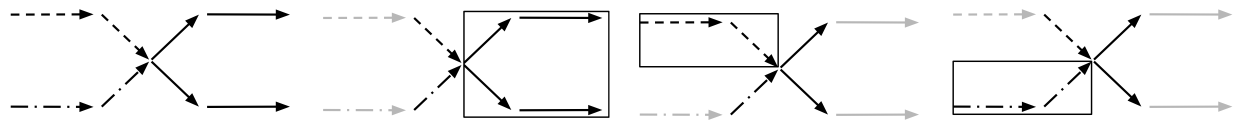
Assume some discrete-time, discrete-space dynamical system which possesses a network-like dynamical coupling in space. Specifically, consider a graph $\left(V^d, E^d\right)$ (not to be confused with the dependency graph (V, D) in Section 3.3.1, where the d superscript is added to denote the *dynamical* network) where E^d represent the discrete-space *cells* and V^d are the *junction points* where cells connect to one another, i.e. each cell in E^d has a corresponding upstream and downstream junction both in V^d. Each discrete space "cell" $c \in \{1, \ldots, N_d\}$ has for each discrete time step $k \in \{1, \ldots, T_d\}$ both a control variable $x[c, k] \in \mathbb{R}$ and a state variable $s[c, k] \in \mathbb{R}$. The variable $s[c, k]$ is assumed to be a function of all state and control variables that satisfy two conditions:

- the time-step is $k - 1$, and
- the cell must share a junction with cell c.

Next, we wish to express a distributed optimization problem subject to the above dynamics in the form of Problem (3.20). To do so, we assume a partition of $\left(V^d, E^d\right)$ into D *sub-networks*, which implies a partition of E^d into D subsets $\left(E^d_1, \ldots, E^d_D\right)$ and assume an objective f which is splittable across the state variables internal to each sub-network. This leads to a state partitioning $s = \left(s_{u(1)}, \ldots, s_{u(D)}\right)$, where $(c, k) \in u(i)$ iff $c \in E^d_i$.

Based on the two conditions for state dependencies above, we can deduce that the state of a sub-network depends on the control and state both internal to the sub-network and directly *neighboring* the sub-network. Explicitly, for sub-network i, we can express the dependent control variables as $x_{v(i)}$ where $(c, k) \in v(i)$ iff $c \in E^d_i$ or c neighbors a cell in E^d_i. Similarly, the shared state for sub-network i is $s_{w(i)}$, where $(c, k) \in w(i)$ iff $c \notin E^d_i$ and c neighbors a cell in E^d_i. Finally, we conclude that there exists some update equation g_i, specific to the particular dynamical system, where the constraint on $s_{u(i)}$ can be expressed familiarly as $s_{u(i)} = g_i\left(x_{v(i)}, s_{w(i)}\right)$.

As an example, we can consider the network in Figure 3.11a, which is partitioned into three subnetworks based on line-style. We see that four of the edges share a single junction between the three subnetworks. Thus, the dynamics assumed above implies that each subnetwork will share state with each other subnetwork. Specifically, the solid-lined network



(a) Complete network (b) Solid subnetwork with two shared links (c) Dotted subnetwork with three shared links (d) Dash-dotted subnetwork with three shared links

Figure 3.11: A network is partitioned into three subnetworks: solid, dashed, and dash-dotted. Each subnetwork will share state with neighboring subnetworks. For a subnetwork i, the cells neighboring i, denoted by E_i^d, are shown in black, while those excluded from E_i^d are shown in gray.

in Figure 3.11b shares one cell each from the other two subnetworks, while the dashed and dash-dotted subnetworks in Figures 3.11c and 3.11d share two cells with the solid subnetwork and one cell with the opposite subnetwork. We note again that while each optimizing agent may have different values of the state on a particular cell in the network during intermediate stages of the A-ADMM algorithm, each copy of the state will eventually come into consensus as the shared-state A-ADMM algorithm converges.

Local Communication Requirements At this point, all relevant parameters to Problem (3.20) have been specified. The assumption on the dynamical network coupling leads to a desirable dependency graph (V, E) for the system above. Since each sub-network only requires shared state from neighboring sub-networks in the sense of the *physical* network $\left(V^d, E^d\right)$, then the dependency graph (V, E) is constructed by assigning a sub-network to each node V and adding an edge (i, j) to E only for those sub-networks i and j which physically neighbor each other. Thus, the A-ADMM algorithm guarantees that communication only take place between physically neighboring systems. This is useful for situations where there are limitations in the networking capabilities due to physical distance, such as freeway traffic control systems, where collaborations may only exist for those districts near each other.

Furthermore, the formulation allows for a completely decentralized and asynchronous implementation of the global optimization problem. If, for instance, all nodes are managed by independent agencies with varying computational limits, then there are several practical benefits to the approach. For a single sub-network, since only information that is directly adjacent to other sub-networks needs to be shared with other sub-networks, much of the internal formulation of the sub-network can be made completely hidden from the larger network. The asynchronicity of the algorithm also permits for neighboring agencies to exchange information in an ad-hoc manner, and not be bottlenecked by slower updates between separate sub-networks.

Scalability of Subnetwork Splitting for Model Predictive Control A common application of finite-horizon optimal control is in the context of model predictive control (MPC) [104, 38], where optimal control policies are recomputed in a *rolling-horizon* fashion. Given the optimal control problem beginning at a time-step t,

$$\begin{aligned} \min_{x=\{x_t,\ldots,x_{t+T}\}} \quad & f_t^{t+T}(s,x) \\ \text{subject to:} \quad & s = g_t^{t+T}(x), \end{aligned} \tag{3.29}$$

MPC chooses the control policy x_t to apply at time-step t by solving for $x = \{x_t, \ldots, x_{t+T}\}$ in Equation (3.29) using a prediction horizon of T and updating the objective f_t^{t+T} and constraints g_t^{t+T} based on the latest estimates of the initial conditions and boundary conditions.

In applications such as freeway onramp metering, a limiting factor in choosing an optimization time-horizon is the accuracy of the predictions of the boundary conditions, or specifically, anticipating future vehicle demands on freeway onramps. At some point, increasing the time-horizon will only decrease the effectiveness of the control due to the deviation in predicted model state versus reality. Thus, it is often practical to consider the time-horizon fixed in MPC applications, at which point the scalability with respect to network size becomes of importance.

For freeway networks with very small branching factors, it is reasonable to assume the following:

- For each subnetwork, the number of bordering links is *constant*.
- The number of shared state and control variables grows *linearly* with the time-horizon for each subnetwork.
- The number of subnetworks scales linearly with network size (for fixed-size subsystems).

One concludes that the amount of communication required for the A-ADMM subnetwork splitting method would scale linearly with the network size and quadratically with time-horizon length. If we were to instead decompose our system, for instance, across time-slices, the communication requirement would scale quadratically with network size and linearly with time-horizon length. Given our assumption of a fixed time-horizon, the subnetwork splitting approach for network-flow MPC has the added benefit of better scaling in the communication requirements.

3.3.4 Solving Sub-problems via the Adjoint Method

What is not explicitly expressed in Algorithm 3 is a solution method for Step 4:

$$y_i^{k+1} = \arg\min_{y \in Y_i} f_i([y]_s) - \sum_{e \in E(i)} \Lambda_{i,e} \lambda_e^{k,T} \big(y_{i,e} - \bar{y}_e^k\big) + \frac{\psi}{2} \|y_{i,e} - \bar{y}_e^k\|_2^2 \tag{3.30}$$

In the more general case of non-convex update equations g_i and objectives f_i, it is difficult to find even local optima for y_i over the space Y_i using gradient-descent methods: a result of the difficulty of projecting and expensiveness of computing gradients in Y_i.

Since $[y_i]_s$ is a deterministic function of the unconstrained variables $[y_i]_{\bar{x}}$ and $[y_i]_{\bar{s}}$, it becomes more efficient to eliminate $[y_i]_s$ from the search space and concatenate $[y_i]_{\bar{x}}$ and $[y_i]_{\bar{s}}$ into a single "free" variable $\bar{r}_i := ([y_i]_{\bar{x}}, [y_i]_{\bar{s}})$. Similar to the convention for $y_{i,e}$ and $y_{j,e}$, we denote $(\bar{r}_{i,e}, \bar{r}_{j,e})$ and $(\bar{s}_{i,e}, \bar{s}_{j,e})$ as the free variables and constrained state variables, respectively, shared between nodes i and j. Then we can repose the sub-optimization Problem (3.30) in the following way. We let

$$\begin{aligned}\bar{f}_i(s_i, \bar{r}_i) :=& f_i([y]_s) - \\ & \sum_{e \in E(i)} \Lambda_{i,e} \lambda_e^{k,T} \big(r_{i,e} - \bar{r}_e^k\big) + \frac{\psi}{2} \|r_{i,e} - \bar{r}_e^k\|_2^2 + \\ & \sum_{e \in E(i)} \Lambda_{i,e} \lambda_e^{k,T} \big(s_{i,e} - \bar{s}_e^k\big) + \frac{\psi}{2} \|s_{i,e} - \bar{s}_e^k\|_2^2\end{aligned}$$

be the "augmented" sub-objective accounting for the additional ADMM terms for subproblem i, where $\bar{r}_e, \bar{s}_e$ denotes the vector mean of $r_{i,e}, r_{j,e}$ and $s_{i,e}, s_{j,e}$ respectively. Also, if we let the concatenated subsystem equations be:

$$H_i(s, r) := s - g_i([r]_{\bar{x}}, [r]_{\bar{s}}),$$

then we have

$$\big(s_i^{k+1}, \bar{r}_i^{k+1}\big) = \arg\min_{s,r} \bar{f}_i(s, r) \tag{3.31}$$

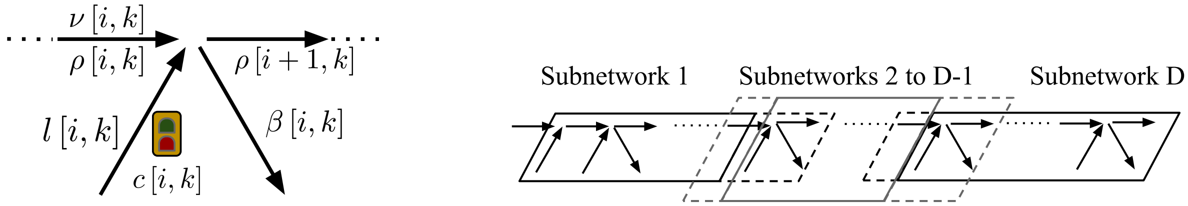
$$\text{subject to:} \quad H_i(s, r) = 0 \tag{3.32}$$

The form of Problem (3.31) permits us to apply the *discrete adjoint method* (Section 3.1.2) to compute gradients of $\bar{f}_i$ at some search point $\bar{r}_i^0$. If we let s_i^0 be defined so that $H_i(s_i^0, \bar{r}_i^0) = 0$, then we arrive at the following expression for the gradient:

$$\nabla_r \bar{f}_i\big(s_i^0, \bar{r}_i^0\big) = \gamma^T \frac{\partial H_i(s_i^0, \bar{r}_i^0)}{\partial r} + \frac{\partial \bar{f}_i(s_i^0, \bar{r}_i^0)}{\partial r} \tag{3.33}$$

$$\text{subject to:} \quad \frac{\partial H_i(s_i^0, \bar{r}_i^0)}{\partial s}^T \gamma = - \frac{\partial \bar{f}_i(s_i^0, \bar{r}_i^0)}{\partial s}^T, \tag{3.34}$$

where γ is the *discrete adjoint* variable and Equation (3.34) is the *discrete adjoint* system.



(a) A single freeway junction near link i.

(b) Diagram of freeway network with A-ADMM subnetwork splitting.

Figure 3.12: Overview of the freeway ramp metering network and state evolution. Figure 3.12a shows the dynamical state and control variables of a particular junction i on the freeway. The relation between mainline density $\rho[i,k]$, onramp queues $l[i,k]$, metering control rate $c[i,k]$, VSL $\nu[i,k]$, and boundary condition split ratios $\beta[i,k]$ for a given time-step k are depicted, and mathematically expressed in Equations (3.37)-(3.41). Figure 3.12b shows how one may partition the linear network into subnetworks. While subnetworks may have internal links and onramps, they will also include links and onramps immediately upstream and downstream as part of their shared state (denoted by the dashed-line boxes), giving the appearance of overlapping subnetworks.

3.3.5 Distributed, Coordinated Optimal Ramp Metering and VSL

We apply distributed optimization via subnetwork splitting to the problem of coordinated, predictive freeway onramp metering and VSL control [93, 38, 104, 84], where traffic lights on freeway onramps are used to regulate the flow entering freeway mainlines and speed limits are dynamically adapted in order to prevent congestion and improve such metrics as driver travel time and speed variability. The term *coordinated* indicates that many traffic lights and VSL signs along a freeway stretch will act cooperatively, given that conditions near one onramp or VSL sign may eventually affect conditions at a neighboring onramp or VSL sign. The term *predictive* indicates that the metering/VSL strategy should anticipate future conditions on the roadway using traffic demand predictions and an underlying model of the evolution of the freeway system.

Similar to discretized freeway models following the cell transmission model (CTM) approach [24] taken in [27, 104], we adopt the *Link-Node* CTM model presented in [84]. The network is given as a linear sequence of mainline link, onramp and offramp triples[2], as depicted in Figure 3.12. We establish the state variables of the system as $s = \{\rho[i,k], l[i,k] : i \in [1,N], k \in [1,T]\}$, where $\rho[i,k]$ is the number of vehicles on the mainline link i (with unit length) and $l[i,k]$ is the number of vehicles queued on onramp i, both at time-step k. Additionally, the control variables are $x = \{(c[i,k], \nu[i,k]) : i \in [1,N], k \in [1,T]\}$, where $c[i,k] \in \mathbb{R}_+$ is the maximum vehicles that can leave onramp i at time k (ramp metering rate),

[2] Freeway models with more general network topologies exist [42] and allow direct application of the subnetwork splitting method presented herewithin. We limit our discussion to linear freeway networks to simplify the presentation.

and $\nu[i,k]$ is the maximum speed of vehicles on link i at time k (VSL rate). The following system of equations relate the state of the freeway at time-step $k-1$ to k:

$$\begin{aligned}
\delta[i,k] &= \min(\nu[i,k]\rho[i,k], f^{\max})(1-\beta[i,k]) + d[i,k] && (3.35)\\
\sigma[i,k] &= \min(w(\rho^{\max} - \rho[i,k]), f^{\max}) && (3.36)\\
d[i,k] &= \min(c[i,k], l[i,k]) && (3.37)\\
f[i,k] &= \min(\nu[i,k]\rho[i,k], f^{\max}) && (3.38)\\
&\times \frac{\min(\delta[i,k], \sigma[i+1,k])}{\delta[i,k]} &&\\
r[i,k] &= d[i,k]\frac{\min(\delta[i-1,k], \sigma[i,k])}{\delta[i-1,k]} && (3.39)\\
l[i,k] &= l[i,k-1] + D[i,k] - r[i,k-1] && (3.40)\\
\rho[i,k] &= \rho[i,k-1] + f[i-1,k-1](1-\beta[i,k-1]) && (3.41)\\
&+ r[i,k-1] - f[i,k-1] &&
\end{aligned}$$

The above model is similar to the model in Section 2.2.3, differing mainly in the junction model. Explicitly, the variables ρ and l are analogous to the discrete mainline density and queue length, equations (3.37)-(**??**) have the same supply and demand analogies of equations (2.23)-(2.25), and finally equations (**??**) and (3.39) are the outgoing vehicle flows corresponding to equations (2.27) and (2.28).

The recursive definitions above require an initial condition,

$$s^0 = \{\rho^0[i], l^0[i] : i \in [1,N]\},$$

and boundary conditions at the left and right extremes of the network,

$$\left(s^L, s^R\right) = \{\left(s^L[k], s^R[k]\right) : k \in [0,T]\},$$

both of which are assumed given. Equations (3.37)-(3.39) can be seen as intermediate computations required to update the state variables given in Equations (3.40)-(3.41), and not explicitly part of the state vector. We note that the offramps are modeled as stateless, infinite-capacity sinks, and thus are only captured through $\beta[i,k]$, the fraction of vehicles which desire to exit offramp i rather than continue to mainline link $i+1$ at time-step k. A diagram of the state and control variables for a single junction is given in Figure 3.12a. The above dynamics are non-convex, but it is shown in [84] that, assuming some maximum velocity V and ramp flow C, if a set of variables satisfy the following linear inequalities and equalities:

$$
\begin{aligned}
& f[i,k] \leq \min(\rho[i,k]V, f^{\max}) && (3.42)\\
& f[i,k](1-\beta[i+1,k]) + r[i+1,k] && (3.43)\\
& \qquad \leq \min(w(\rho^{\max} - \rho[i+1,k]), f^{\max}) &&\\
& r[i,k] \leq \min(C, l[i,k]) && (3.44)\\
& \qquad \text{Eqns } (3.40) - (3.41), &&
\end{aligned}
$$

then a control c, ν can be constructed such that f, r, ρ, l, c, ν satisfy Equations (3.37)-(3.41). Thus, we can employ the adjoint method presented in Section 3.3.4 on the relaxed problem in order to improve sub-objectives during each iteration of the A-ADMM algorithm with a guarantee of convergence to the global optimum. We omit the explicit c, ν reconstruction procedure and refer the reader to [84] for details.

As an objective, we use *total travel time*, or the cumulative time spent by all vehicles on the network. Total travel time is mathematically expressed as

$$f_{\text{TTT}} = \sum_{i,k} \rho[i,k] + l[i,k],$$

and is decomposable across subnetwork splits.

It is clear from the definitions of s and x above that each state variable is a direct function of only the state and control variables of neighboring links at the previous timestep, and as such, can be decomposed using the subnetwork splitting method in Section 3.3.3. Figure 3.12b depicts such a splitting, where each subnetwork also includes the neighboring upstream and downstream links as boundary conditions.

The dependency graph (V, E) for such a network has a natural structure, where an edge (i, j) is in E if and only if $j = i + 1$, and thus a subnetwork need only communicate with the linear subnetworks immediately upstream and downstream of itself. Furthermore, only information pertaining to the bordering links and onramps of a subnetwork needs to be shared with its neighbors, allowing a subnetwork to conceal the particular implementation of its internal freeway model from the rest of the system.

3.3.6 Numerical Results

Convergence with Number of Subnetworks We first investigate the numerical convergence of the A-ADMM metering and VSL controller on a model 4-lane freeway network spanning 12 miles ($N = 12$) with 3 onramps and 2 offramps over a 2 hour simulation ($T = 120$). We consider three different partitionings by splitting the network into 2, 3 and 4 subnetworks, respectively. We also simulated the following alternative controllers for comparison:

- No control: Metering rates are set to maximum ramp flux rates C and speeds are set to free flow velocity V.

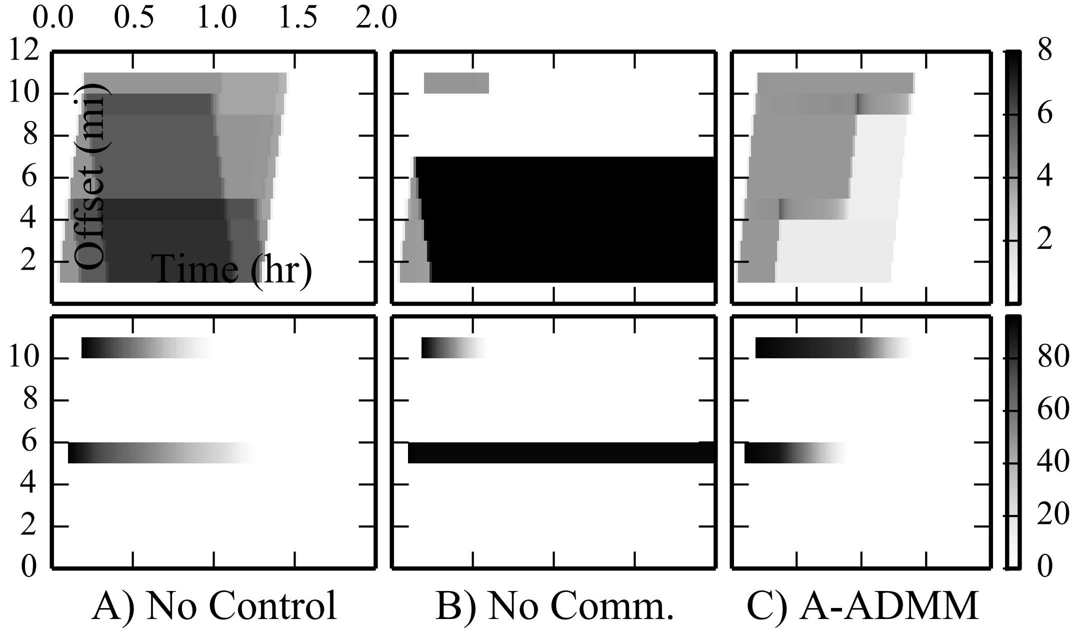


Figure 3.13: Space-time diagrams of mainline (row 1) and onramp (row 2) vehicle count evolution for 12 mile network. Large congestion pockets appearing for (A) no control case are reduced using coordinated holding of vehicles on onramps and decreased speed limits during periods of congestion (C). Lack of communication between subsystems (B) leads to an ineffective control policy.

- Centralized: A single optimal control problem over the entire freeway is solved as a convex optimization problem. This solution gives the theoretical lower bound on total travel time.
- No communication: Individual subnetworks optimize over their own decomposed total travel time objective, with no exchange of information between subnetworks.
- Communicative: Subnetworks iteratively optimize over decomposed objectives and exchange the resulting predicted boundary conditions with neighbors until resulting boundary conditions converge (see [38]). There is no guarantee of convergence of boundary conditions or of finding the global optimum.

Figure 3.13 gives a space-time depiction of the mainline and onramp vehicle evolution for the no control, no communication, and A-ADMM controllers.

The convergence results for the 12 mile freeway network are summarized in Figure 3.14. The centralized approach is faster than the distributed approaches (A-ADMM and communicative) as the former does not require an outer communication loop. As the number of

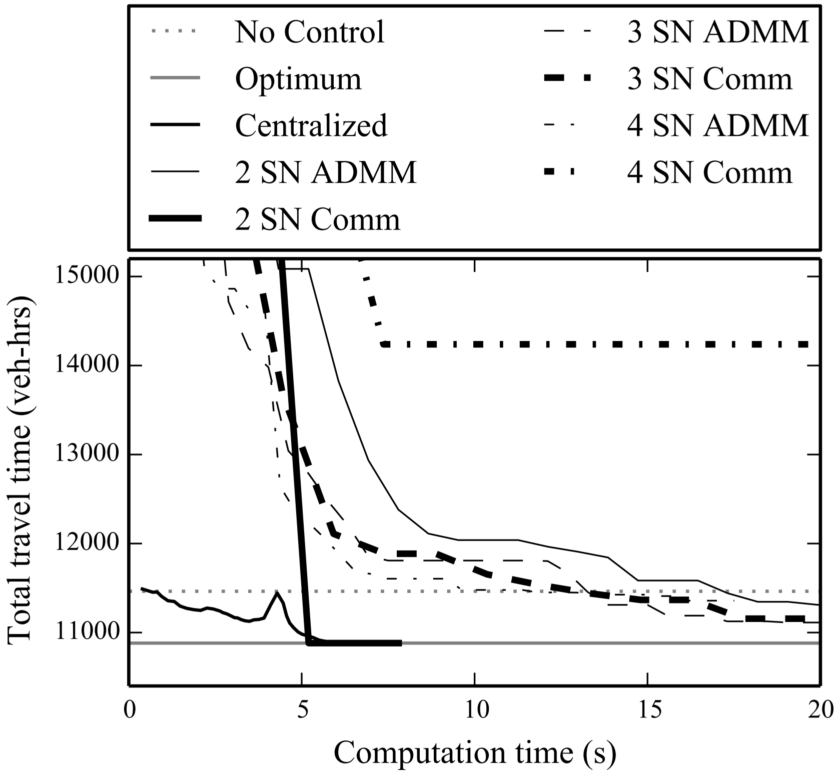
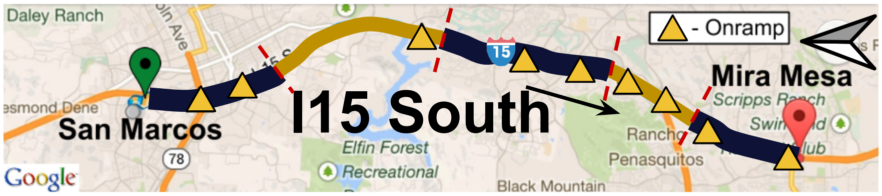


Figure 3.14: Total travel time vs. computation time for several different different control schemes and subnetwork (SN) partitionings. The no-communication results are omitted due to poor performance.

network partitions increases, A-ADMM converges faster to the optimum due to the parallelization of the subnetwork optimizations. Furthermore, the communicative algorithm degrades in performance with increasing number of partitions due to the increase in communication requirements and lack of global objective coordination. If a decentralized algorithm is required for architectural reasons, then the A-ADMM approach is shown to be most desirable due to the lower degree of coordination than the centralized approach and better convergence than the communicative approach.

Distributed MPC for I15 Network MPC simulations were run on a calibrated model of the I15 South freeway in San Diego, CA with boundary flow data taken from measurements recording during a morning rush hour. The simulation spans 20 miles ($N = 32$), contains 9 onramps and runs over a 170 minute window ($T = 1000$) with an MPC update time of 17 minutes and a horizon of 25 minutes. The network is partitioned into 5 subnetworks and is depicted geographically in Figure 3.15a.

Table 3.15b gives a summary of the performance of the A-ADMM MPC controller



(a) Geographical depiction.

Opt.	No Con.	Cent. MPC	No Comm.	Comm.	A-ADMM
2.572	2.605	2.584	4.529	8.453	2.589

(b) Total travel time summary in 1000 vehicle hours.

Figure 3.15: I15 South MPC simulation summary. Figure 3.15a shows the freeway under consideration partitioned into 5 subnetworks, while Table 3.15b gives a summary of the performance of the different ramp metering/VSL controllers.

along with other controllers. The results indicate that the A-ADMM controller performs nearly as well as the centralized MPC controller, which can be viewed as a lower-bound on the performance of MPC controllers with limited horizons. The communicative approach performed worse than the non-communicative approach because its iterative terminated after reaching a set number iterations on a highly inefficient solution, due to its lack of convergence guarantees.

Chapter 4

Security of Traffic Control Systems

Public traffic infrastructure is arriving in the cyber age with increasing connectivity between the different segments of roadways. For example, freeways are commonly instrumented with loop detectors that allow for real-time monitoring of roadway speeds [65]. Estimates of road traffic conditions are then fed directly into onramp traffic light metering algorithms which regulate traffic flow to improve congestion [93]. Finally, these metering algorithms can be coordinated and controlled by a remote command and monitoring center, leading to a regional network of interconnected sensors and controllers [104].

Increased efforts to build systems which understand and utilize the interconnectivity are evidenced by *integrated-corridor-management* (ICM) projects such as *Connected Corridors* [78] and mobile applications which use GPS probe data to improve navigation [128].

This connectivity offers great potential to better analyze, control and manage traffic but also poses a significant security risk. A compromise at any level of the traffic control infrastructure can lead to both direct access of an attacker to alter traffic lights and changeable message signs, and indirect access via spoofing of sensor readings, which may *trick* the control algorithms to respond to false conditions.

A number of traffic-related attacks of infrastructure systems have already been demonstrated in the past few years. A man-in-the-middle attack on GPS coordinate transmissions from mobile navigation applications showed it is possible to trick navigation services into inferring non-existent jams [64], while a similar attack used a fleet of mobile phone emulators to mimic the presence of many virtual vehicles on a roadway [121]. A popular vehicle-detection sensor was revealed to use a type of wireless protocol vulnerable to data injection attacks, and a demonstration showed that the access point could be tricked into receiving arbitrary readings [129]. Cyber attacks on a centralized command center remain a serious threat given the frequent discovery of networking vulnerabilities, such as the Heartbleed bug [21]. Even insider attacks on command centers have precedent as two Los Angeles traffic engineers in 2009 were found guilty of intentionally creating massive delays by adjusting signal times at key intersections [52].

Given the existence of such vulnerabilities and the scale at which they can be exploited, understanding the nature and costs of such attacks becomes paramount to public safety.

In this chapter, we present a systematic approach to analyzing the topic of traffic control system vulnerabilities and their potential impact.

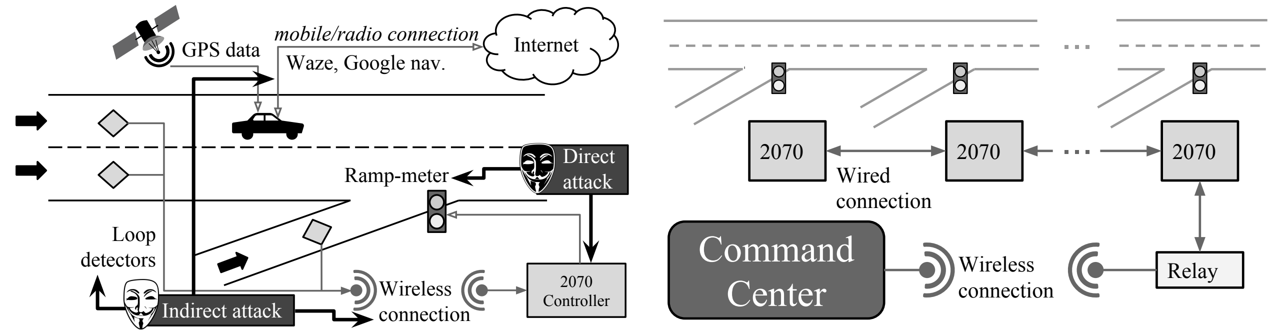
To do so, we begin by constructing a taxonomy of different vulnerability locations in traffic control systems, defining three distinct layers: physical, close-proximity, and virtual. Difficulty, impact, and cost values are also associated with each potential attack. We motivate our classifications by presenting two scenarios that combine a number of attacks to accomplish a high-level goal.

We then focus our analysis on an in-depth exploration of freeway attacks using coordinated, ramp metering. To achieve this, we develop a method based on adjoint computations and finite-horizon optimal control for finding optimal metering rates to create a desired disruption on the freeway. We additionally give an overview of multi-objective optimization and discuss how such an approach is useful for solving high-level attack objectives which contain many conflicting sub-goals, such as permitting a fleeing vehicle to escape pursuants on a particular freeway stretch without overly congesting freeway regions irrelevant to the pursuit.

Central to the plausibility of intricate freeway attacks is the efficiency in which the control and state space of the freeway can be explored. For large systems, brute force exploration would be infeasible, and one could not expect an effective strategy to be computed in a reasonable amount of time. To overcome the large control and state space, this work suggests application of the adjoint method (Chapter 3) and its distributed extension (Chapter 3.3.3) within the multi-objective optimization framework. While the derivations are given explicitly for the centralized form of adjoint control, the methodology extends the distributed case.

The contributions of this chapter are as follows. We present a classification of a broad set of attacks on traffic control systems with their relation to the underlying physical and cyber infrastructure. Mathematical formulations based optimal control and adjoint-based methods are used to show exactly how an attacker can exploit these weaknesses. Explicit algorithms using these tools for coordinated ramp metering attacks are derived and presented. Finally, we provide numerical evidence and novel results of the feasibility of these attacks via simulations modeled after actual freeway networks.

The rest of the chapter is organized as follows. Section 4.1.2 summarizes and classifies the vulnerabilities of traffic control systems. Section 4.2 gives a mathematical approach for carrying out a class of the presented attacks. Sections 4.2.3 and 4.2.4 give two detailed applications of the mathematical approach to ramp metering attacks. The first application shows how ramp metering can allow an attacker to cause congestion in precise locations and at precise moments in time along a freeway. Simulations are applied to a full-sized model of a 19.4 mile stretch of the I15 South Freeway in San Diego, California. Results are shown for both a custom macroscopic flow simulator as well as an Aimsun [6] microscopic model. The second application finds a strategy to solve the aforementioned problem of allowing a fleeing vehicles to escape pursuants. Numerical results are presented, as well as a discussion of the benefits of the multi-objective optimization method. We conclude with some future areas of study for traffic system security.



(a) Local freeway control system.

(b) Global freeway control system.

Figure 4.1: The physical roadway, sensors, connected vehicles and controllers near a freeway/onramp junction in Figure 4.1a form a cyber-physical network we refer to as a local freeway control system. The mask icons (white/black masks for indirect/direct vulnerabilities) denote vulnerability points in the local control network. In Figure 4.1b, the local controllers are wired together, then connected to a command center via a relay box to form the global control system. This chapter analyzes vulnerability locations associated with each component.

4.1 Traffic Control Systems and Vulnerabilities

In the later part of the chapter we propose attacks to create congestion based on user-defined needs. This section reviews the current architecture of freeway control systems to show that these attacks can be implemented in practice on such systems.

4.1.1 The Freeway Control System

Modern freeways encompass control and monitoring mechanisms which enable traffic management to mitigate congestion and improve traffic flow in real-time. While the exact combination of sensors, controllers and transmitters differ from location to location, this chapter chooses one particular instantiation of a freeway control system, which we find to be representative. Figure 4.1a shows a control system installed near a junction of a freeway and an onramp. We consider three elements of the control system:

- Sensors, used to gather information about the freeway state. For example, loop detectors are used to acquire the flow of vehicles along the freeway and onramps/offramps, while the trajectory of vehicles equipped with GPS (or containing GPS-powered smartphone applications) can be used for estimating real-time traffic conditions [128].

- Actuators, used to influence the evolution and efficiency of the freeway. The most common actuation strategy is *ramp metering*, where traffic lights installed on freeway onramps control the influx of vehicles to the mainline. Other actuators include variable speed limit control [84] and variable message signs. For the purposes of this chapter, the ramp meters are the only actuators we will consider.

- Local controllers, such as *2070* boxes [3] and the older *170* boxes [34], which allows interaction between the sensors and ramp meters.

We assume control boxes are wired to the nearby metering light and have a wireless connection to nearby sensors. Vehicles with navigation devices such as TomTom automatically analyze radio-broadcasted traffic reports from traffic control centers to improve their navigating functionality.

In order to allow coordinated control and sensing across a freeway stretch with many onramps, the local control systems are connected to allow for a more global configuration. Figure 4.1b depicts our representative global communication architecture. The local control boxes are wired together along the freeway to form the actuation network, with intermediary *relay boxes* allowing for an uplink and downlink to a remote *command center*. The command center contains instrumentation and personnel for monitoring traffic conditions and setting the metering lights accordingly.

4.1.2 Vulnerability Classification

The traffic control infrastructure is built up of several layers and each layer poses individual security risks, starting from tampering with the actual devices, cables or wireless signals, to attacking the software of deployed devices or attacking the command center. Attackers can leverage vulnerabilities in the infrastructure to control or disrupt these connected systems. Individual attacks can thereby target the physical layer, the communication layer, the layer of the control center, or any combination thereof.

Direct physical access: The physical layer is the lowest attackable layer and involves direct access to individual wires, opening and accessing the control box, or tampering with individual sensors. Physical attacks involve clipping, tampering, removing, or replacing of wires or hardware. For instance, copper wire theft near freeways is a common occurrence [119, 107]. Such attacks need low sophistication, are easy to carry out, and are hard to protect against as each device must be physically protected given that software-based protection is not effective against physical attacks. On the other hand, the attack is costly as (i) direct physical access is needed, (ii) the attacker is exposed, and (iii) the attack does not scale (i.e., each piece of equipment is attacked individually). Examples of such an attack in Figure 4.1a include clipping or removing wires between sensors and the *2070* controller, tampering with individual sensors, the ramp meter, or the *2070* controller.

Proximity access (locality): Figure 4.1b depicts multiple control boxes chained together to form a corridor where actuators have a coordinated plan between the different control boxes. An attack on the communication layer forges, removes, replaces, or inserts attacker-controlled measurements into the control system, which may then make further decisions based on forged data. An attacker can either replace or add sensors to the current sensor

Attack Description	Access	Control	Complexity	Cost
copper theft/clipping wires	physical	low	low	low
replacing a single sensor/actuator	physical	low	low	low
attacking a single sensor/actuator	locality	low	medium	low
replacing a single control box	physical	medium	medium	medium
replacing a set of sensors/actuator	physical	medium	medium	medium
attacking a set of sensors/actuator	locality	low	medium	low
replacing a corridor of control boxes	physical	high	medium	medium
attacking a corridor of control boxes	network	high	high	medium
attacking the control center	network	high	high	high
spoofing GPS data	network	medium	high	medium
attacking navigation software	network	medium	medium	medium

Table 4.1: List of possible infrastructure attacks with access to different layers that is needed, level of control that the attacker gains, sophistication of the attack, and cost.

network to inject new measurements or attack the software running on sensors and/or actuators to take over control. Both aspects of the attack are feasible; the first aspect needs additional hardware and an attacker that delivers the hardware, the second aspect needs to find a software vulnerability with a security analysis of the existing devices. These attacks need higher sophistication and knowledge but no longer need direct hardware access to the existing sensors and scales to some extent.

Networked/virtual access: Remote connections from the physical freeway infrastructure to the command center defines another layer with potential vulnerabilities. An attack on this layer can be done by forging or controlling messages from/to the command center and possibly even compromises the command center itself. For this scenario an attacker needs to find software vulnerabilities in the software running in the command center. Direct access to these centers is usually not given and this attack therefore is highly sophisticated (or needs insider access). This attack is the hardest possible attack as command centers and back links are usually guarded but allows a great scaling effect as many control boxes can be controlled directly.

Table 4.1 gives a (partial) list of vulnerabilities in our freeway control system along with classifications for each attack.

Motivating Examples

We will consider two fictional but realizable attack scenarios and study their consequences on the compromised network. The first scenario involves indirect control of the freeway, through spoofing the sensors, to achieve a local objective. The second scenario involves direct control of the ramp meters to achieve a global objective along a larger stretch of freeway.

The distinction between direct and indirect control is illustrated in Figure 4.1a via the white mask (indirect) and black mask (direct) icons; direct control can set arbitrary metering rates to a single traffic light or to many lights in a coordinated fashion, while indirect control only modifies sensor readings, with the anticipation that the uncompromised metering system will respond to the spoofed sensors in a predictable manner. Examples of direct attacks include a compromise of the *2070* boxes which are directly wired to the meters and a compromise of the command center, which issues upstream metering plans to the *2070* controllers. Examples of indirect attacks include sending fake loop-detector readings to access points and broadcasting false traffic reports to GPS devices which may respond with poor routing advice.

Direct Attack: catch-me-if-you-can The objective of the attacker is to escape from pursuants along a large section of freeway. A compromise of all the ramp meters is assumed, as it permits the attacker to selectively congest certain sections of the roadway (see Section 4.2). One approach is to hack the command center itself, with the downside being the expensiveness and complexity of such an attack (see Table 4.1). Another solution is to begin by hacking one of the *2070* boxes, and since all the *2070* boxes are networked along the freeway (see Figure 4.1b), a single hacked box can serve as a means of compromising the other nearby boxes, leading to a cascading attack. The attacker can then acquire full control of all the *2070* boxes, and in turn, the ramp metering lights.
The current traffic control architecture presented above supports the class of attacks described in the next section. Specifically, a mathematical approach to coordinated ramp metering attacks is developed to permit an attacker to effectively exploit vulnerabilities in the metering control system.

Indirect Attack: VIP-lane The objective of the attacker is to clear a predetermined section of a regularly congested freeway. The attacker drops low-cost wireless transmitters near the *2070* controllers along the freeway section. As the actual loop-detector sensors communicate with the control box wirelessly, the attacker will be able to override the loop-detector signals and send false data that indicates a fully congested freeway. This will indirectly affect the ramp meters, which will respond by limiting on ramp flow and thus eliminating significant freeway mainline flow. The attacker will then transmit false GPS location data via a set of hacked cellphones to trick navigation software into believing the freeway is congested. Approaching vehicles using navigation software will then be rerouted around the fake congestion which leads to a further reduction in incoming flow. The net effect of the attack is a congestion-free commute for the attacker: a private VIP lane created purely by indirect, sensor-based attacks.

A depiction of the VIP-lane attack is shown in Figure 4.2. The attack was also im-

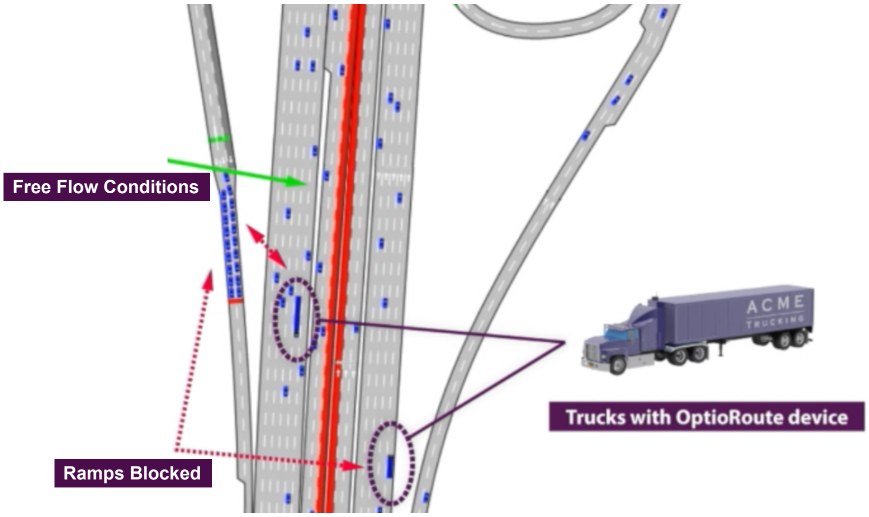


Figure 4.2: Diagram of the VIP-Lane attack on the Aimsun micro-simulation. Trucks (dramatized in [102] as belonging to a hypothetical delivery company, OptiRoute) passing by transmit fake traffic count data in an attempt to spoof the readings received by the traffic count sensors. The count sensors are deceived into inferring high-density conditions, causing large amounts of metering on the on ramps while creating free-flow conditions on the mainline.

plemented numerically as a part of the SmartRoads project, which is discussed in the next section.

4.1.3 SmartRoads and SmartAmerica

As part of the White House SmartAmerica 2014 Conference on cyber-physical systems, A UC Berkeley PATH institute and Vanderbilt University collaboration presented a functional microscopic simulation framework for conducting and analyzing cyber attacks on freeway control infrastructure. The project, nicknamed SmartRoads, consists of three main components:

- UC Berkeley *Connected Corridors*: A simulation and decision support system for traffic systems which implements estimation and ramp metering control.
- TSS Aimsun [6]: a microscopic traffic simulator with API's for broadcasting loop-detector information and modifying metering rates dynamically.
- Vanderbilt C2WindTunnel [17]: A computer network communication simulator and visualization tool.

The ultimate mission of the collaboration is to allow for a comprehensive modeling of traffic control system vulnerabilities to enable real-time security compromise detection and

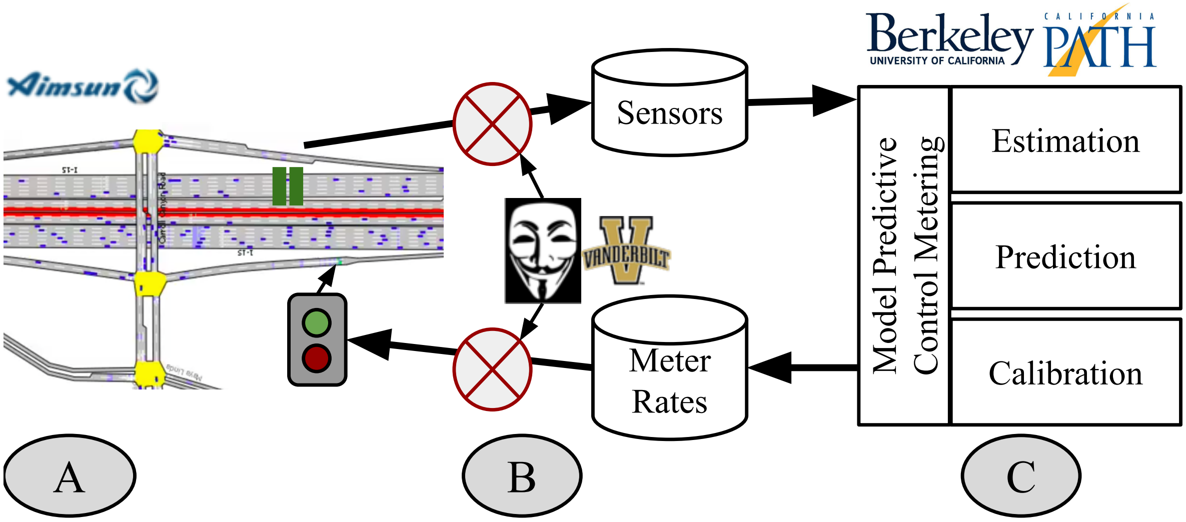


Figure 4.3: System diagram for SmartRoads project. The Aimsun micro-simulator (Label A) interfaces with the Connected Corridors Control System (Label B) through databases. The C2WindTunnel (Label C) system models networking and serves as the attack point. Sensors information intercepts are indirect attacks, while metering command intercepts are direct attacks.

prevention. SmartRoads aims to accomplish this by modeling the flow of information and communication between the Connected Corridors system and the Aimsun simulator.

As a initial task, a simulation of both indirect and direct attacks was conducted on a calibrated microscopic model of the I15 Freeway. As the communication between Aimsun and Connected Corridors is conducted through database intermediaries, the C2WindTunnel software was installed on the communication link between Aimsun and the databases. Before sensor information reached the database from Aimsun and before metering commands reached Aimsun from the database, C2WindTunnel simulated a communication interception and modified the contents of the packets. A sensor intercept is viewed as a simulated indirect attack, while a metering rate intercept is viewed as a simulated direct attack. The system architecture of SmartRoads is summarized in Figure 4.3.

The VIP-Lane attack was fully simulated on the SmartRoads system, where a video playback of the simulation clearly indicates that a spoofing attack on the loop detectors results in an under-utilized freeway mainline, while the metering algorithms are "tricked" into over-metering. See our link [102] for a Youtube video depiction.

A direct attack was also simulated in the SmartRoads framework, demonstrating the interception of metering commands to a series of traffic lights along I15 South. The attack, referred to as a *box attack*, attempts to create a precise pocket of congestion over a predetermined section of road and period of time. The details of box attacks, as well as a visual summary of the direct attack within SmartRoads, are given in Section 4.2.3.

4.2 Coordinated Ramp Metering Attacks

An attacker can negatively influence the performance of the freeway network or achieve some criminal goal by setting the metering lights to a particular configuration. The impact of such an attack can be maximized by leveraging a discrete dynamical freeway model to compute metering rates which achieve the desired goal using finite-horizon optimal control and multi-objective optimization techniques.

4.2.1 Optimal Control Model

Optimal Control Formulation

Using the discrete model in Section 2.2 mathematically expressed as follows:

$$\delta[i,k] = \min(v\rho[i,k], f^{\max}) \tag{4.1}$$

$$\sigma[i,k] = \min(w(\rho^{\max} - \rho[i,k]), f^{\max}) \tag{4.2}$$

$$d[i,k] = u[i,k]\min(l[i,k]/\triangle t, r^{\max}) \tag{4.3}$$

$$f^{\text{in}}[i,k] = \min(\sigma[i,k], d[i-1,k] + \beta[i,k]\delta[i,k]) \tag{4.4}$$

$$f^{\text{out}}[i,k] = \begin{cases} \delta[i,k] & \text{if } \frac{pf^{\text{in}}[i+1,k]}{\beta[i,k](1+p)} \geq \delta[i,k] \\ \frac{f^{\text{in}}[i+1,k]-d[i+1,k]}{\beta[i,k]} & \text{if } \frac{f^{\text{in}}[i+1,k]}{1+p} \geq d[i+1,k] \\ \frac{pf^{\text{in}}[i+1,k]}{(1+p)\beta[i,k]} & \text{otherwise} \end{cases} \tag{4.5}$$

$$r[i,k] = f^{\text{in}}[i,k] - \beta[i,k]f^{\text{out}}[i,k] \tag{4.6}$$

$$\rho[i,k+1] = \rho[i,k] + \frac{\triangle t}{\triangle x}\left(f^{\text{in}}[i,k] - f^{\text{out}}[i,k]\right) \tag{4.7}$$

$$l[i,k+1] = l[i,k] + \triangle t(D[i,k] - r[i,k]), \tag{4.8}$$

we seek a method to compute a coordinated ramp metering policy $u[i,k]$ over all space $i \in [1,N]$ and time $k \in [1,T]$, which minimizes (or reduces) some specified objective. We cast the problem as a finite-horizon optimal control problem, as done in Section 3.1.1.

We succinctly express the discrete, controllable dynamical system given in Section 2.2 by:

$$H(\mathbf{u},\rho) = 0. \tag{4.9}$$

Given some objective function $J(\mathbf{u},\rho)$, our goal is now to find the optimal $\mathbf{u}^*$ which solves the following constrained *finite-horizon optimal control* problem:

$$\min_{u} J(\mathbf{u},\rho) \tag{4.10}$$

$$\text{subject to: } \textit{Equation } (4.9). \tag{4.11}$$

We utilize the discrete adjoint method discussed in Section 3.1.2 to efficiently compute gradients within a first-order descent method.

4.2.2 Multiple Objectives: Interactive Multi-objective Optimization

A high-level attack goal often requires satisfaction of many *sub-goals* at once, and oftentimes the sub-goals can be competing or conflicting. For example, in the *catch-me-if-you-can* scenario, the attacker wants to escape from his chasers. Hence the attacker wants to travel the freeway as quickly as possible, but also wants to slow down the chasers behind. As a consequence, we have two simpler but competing objectives.

Such a situation with multiple, competing objectives can be described as a *multi-objective optimization problem.*

Multi-objective Optimization and Pareto Front

Definition 4.1 (Multi-objective optimization problem)**.** *Given $N \in \mathbb{N}$, let $(f_i(\mathbf{u}, \rho))$ be a set of N objective functions describing the goal of a freeway attack. The* multi-objective optimization problem *we consider is the following simultaneous minimization problem:*

$$\min_{x \in X} \left(f_1(x), f_2(x), \ldots, f_N(x)\right) \tag{4.12}$$

As we are now minimizing a vector and not a scalar, we need to define how a solution of equation (4.12) can be "better" than another.

Definition 4.2 (Pareto front)**.** *An solution $x \in X$ is said to* Pareto dominate *another solution x' if:*

- $\forall i \leq N \quad f_i(x) \leq f_i(x')$
- $\exists j \leq N \quad f_j(x) < f_j(x')$

A solution $x \in X$ is called Pareto optimal *if there is no other solution x' that dominates it. The set of all Pareto-optimal solutions is called the* Pareto front*, $P \subseteq X$.*

Hence, we consider Pareto-optimal solutions to be the solutions of Equation (4.12).

Decision Maker

There are many ways to find a Pareto-optimal solution. For example if we have three objective functions, we can minimize f_1 first, minimize f_2 on the subset $\arg\min_{x \in X} f_1(x)$ and finally minimize f_3 on the remaining subset to obtain a Pareto-optimal solution. But we could also do the same in any order, with potentially very different results. Thus, the Pareto front can sometimes be very large and hard to explore. As a consequence, we need to be able to identify the most desirable solutions within the potentially large Pareto front.

A *Decision Maker* (DM) represents the human whose expertise is used to discern a preference between two control values. As we only wish to judge controls which are Pareto optimal, The DM only observes and discerns values on the Pareto front to limit the search space and improve the efficiency of the method. As a consequence, the DM has a hidden objective function: $u(\mathbf{u}, \rho)$, the *utility function*, which can only be indirectly observed through probing the DM. With u, we can reformulate the multi-objective optimization problem as:

$$\min_{x \in P} \; u(x) \tag{4.13}$$

The DM is essential to multi-objective optimization problems with large Pareto fronts. There are several ways to interact with him:

- He can evaluate his utility function u on any given Pareto-optimal solution.
- He can give more general preferences on the Pareto front, for example a preference for one of the objective functions, or for a given subset of the Pareto front.

Finite-horizon Optimal Control and Multi-objective Optimization

Scalarization In order to find Pareto-optimal solutions, we will reduce the problem to the common scalar minimization problem, which can be solved with the optimal control tools of Section 4.2.1. This process is called *scalarization.* As our particular scalarization, we use a linear combination of the individual objective functions:

$$f(x) = \sum_{i \leq N} a_i f_i(x). \tag{4.14}$$

The DM can favor a specific objective f_i over other objectives by increasing the a_i coefficient.

It is easy to show that any solution of Equation (4.14) will also belong to the Pareto front. As a consequence, we can explore at least a subset of the Pareto front (with the hope that this subset is representative) by minimizing a linear combination of the objective functions.

For example, the next proposition shows that a linear combination of the objectives is a type of scalarization.

Proposition 4.2.1. *If a solution $x \in X$ satisfies:*

$$\exists \; (a_i)_{i \leq N} \in \mathbb{R}_+^* \qquad x \in \arg\min_{y \in X} \sum_{i \leq N} a_i f_i(y) \tag{4.15}$$

Then x is a Pareto optimal solution

Proof. If x is not a Pareto optimal solution, a solution $y \in X$ Pareto - dominate x, by Definition 4.2. As a consequence, we have:

- $\forall i \leq N \quad f_i(y) \leq f_i(x)$

- $\exists j \leq N \quad f_j(y) < f_j(x)$

And immediately:

$$\sum_{i \leq N} a_i f_i(y) \quad < \quad \sum_{i \leq N} a_i f_i(x)$$

So $x \notin \arg\min_{y \in X} \sum_{i \leq N} a_i f_i(y)$, hence the proof by contraposition. □

A Posteriori Method Equation (4.14) allows one to sample the Pareto front by exploring the space of the coefficients which can provide to the DM a representative subset of Pareto-optimal solutions. The DM can then choose *a posteriori* his preferred solutions. And as such this method is called an *a posteriori method*.

This method can be computationally costly as many different optimal control problems need to be solved, but provides a good overview of the Pareto front. In particular, it gives an estimation of the lower and upper bounds of each objective function. Thus one can scale each objective function to take values only between 0 and 1, allowing the different objectives to be easily compared.

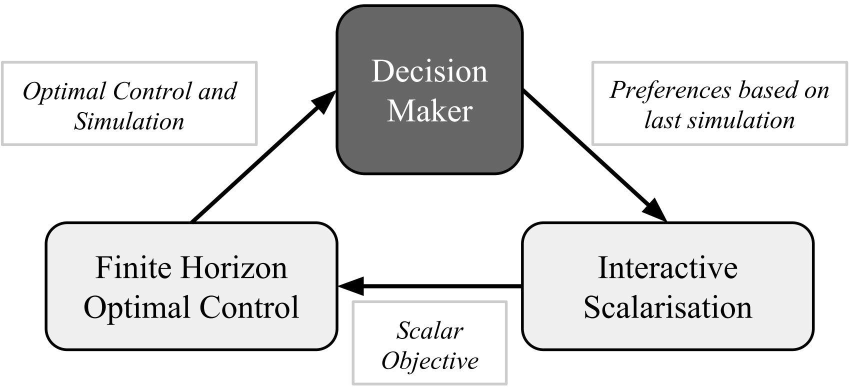

Decision Maker
Optimal Control and Simulation
Preferences based on last simulation
Finite Horizon Optimal Control
Interactive Scalarisation
Scalar Objective

Figure 4.4: The interactive method for multi-objective optimization embeds the Decision Maker (DM) in the optimization loop, allowing the DM to direct the search of the Pareto front. The optimal controller adapts the advice of the DM to *scalarize* the multiple objectives and solve a new optimization problem. The results of the optimization are then fed back to the DM, and the cycle repeats until satisfaction.

Interactive Method Unlike with the a posteriori method, *Interactive methods* are based upon a repeated interaction with the Decision Maker.

1. The DM gives an indication of how to compute the next Pareto-optimal solution — for example, he may give an idea for the next set of coefficients (a_i) to use and his evaluation of the previous simulation.

2. The interactive scalarization process uses this indication to create a scalar objective — for example using Equation (4.14), we obtain a scalar objective with the set of coefficients given by the DM.

3. The finite-horizon optimal control method is used to solve the corresponding optimization problem, and gives the result to the DM.

This process is repeated until the DM is satisfied with the results. Figure 4.4 shows the general process of interactive methods.

The important part of the interactive method is the kind of indications that can be given by the DM, and how the indications and the simulation history will be used in the scalarization process. Section 4.2.4 gives an example of an interactive method.

4.2.3 Congestion-on-demand Attack

We will now apply the tools of *adjoint-based finite-horizon optimal control* and *multi-objective optimization* from Section 4.2 to two families of attacks. The first attack highlights the precision of coordinated ramp metering attacks, while the second showcases the benefits of multi-objective optimization.

Following reproducible research practices [29, 116], the software and data used to produce the numerical results and diagrams in this section is made available [102] to permit the reader to reproduce the presented results.

Simulation Network

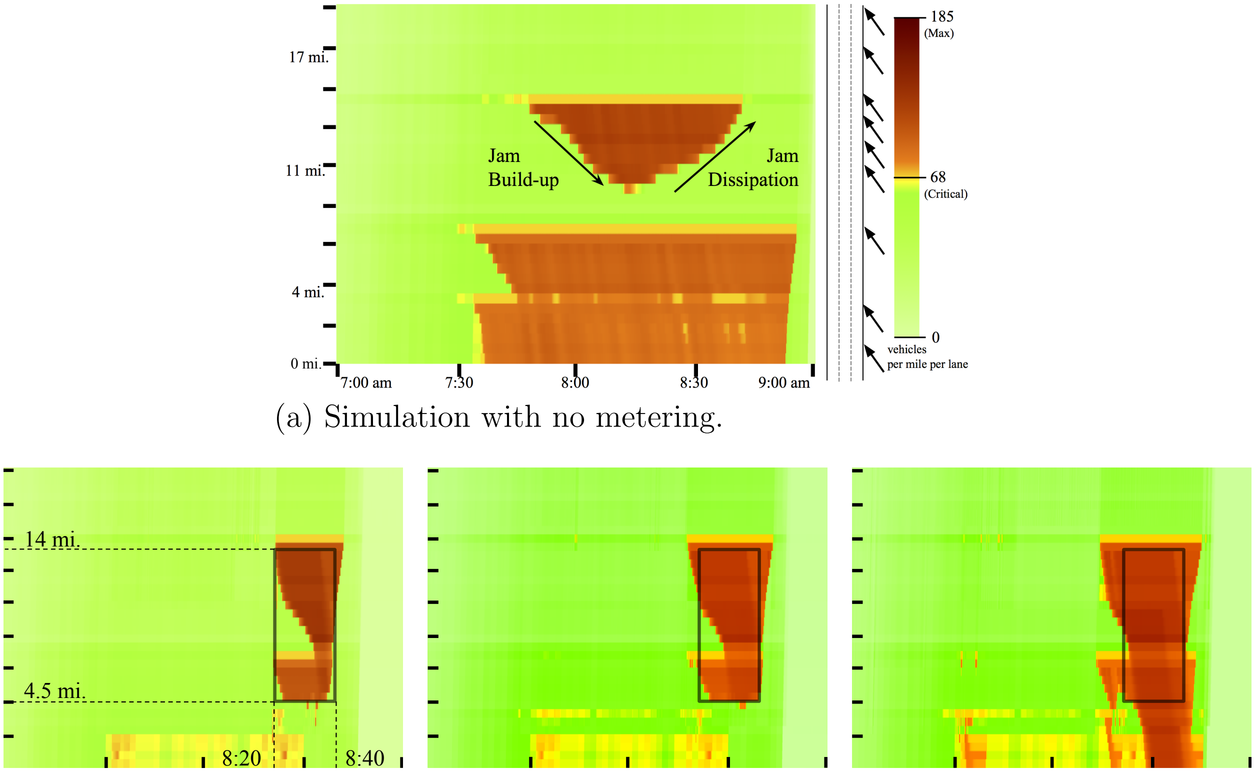
The attacks for the first example, *box objective* (to be described), use a macroscopic freeway model of a 19.4 mile stretch of the I15 South Freeway in San Diego California. The model was split into 125 links with 9 onramps and was calibrated [28, 83] using loop-detector measurements available through the PeMS loop-detector system [65]. Figure 4.5a is a *Space-time diagram* of the I15 freeway. There is no ramp metering control applied to the simulation in Figure 4.5a, i.e. the ramp meters are always set to green.

Constructing the objective function

In order to achieve the *congestion-on-demand* objective, we will use the finite-horizon optimal control technique introduced in Section 4.2.1. Therefore, we need to create a class of objective functions able to represent any jam pattern on the freeway. The method we have chosen is to maximize the traffic density where we want to put the congestion, while minimizing it everywhere else.

For every cell density value at position i and time k, we assign a coefficient $a_i^k \in \mathbb{R}$. We can then define the corresponding objective function:

$$J(\mathbf{u}, \rho) = \sum_{i=1}^{N} \sum_{k=1}^{T} a_i^k \, \rho[i, k] \tag{4.16}$$



(a) Simulation with no metering.

(b) Trade-off: $\alpha = 0.3$

(c) Trade-off: $\alpha = 0.5$

(d) Trade-off: $\alpha = 0.9$

Figure 4.5: Figure 4.5a depicts a space-time diagram of vehicle densities on 19.4 mile stretch of I15 Freeway with no ramp metering. The box objective, and example of *congestion-on-demand*, is applied in Figures 4.5b-4.5d. The user specifies a "desired" traffic jam between postmile 4.5 and 14, for a duration of 20 minutes between 8:20 and 8:40. For this, the α parameter (introduced in Equation (4.19)) enables the proper design of tradeoffs in the objective.

When J is minimized, a positive coefficient a_i^k will encourage the minimization of the traffic density at position i and time k, whereas a negative coefficient will encourage congestion. The absolute value of the coefficient represents the importance given to the fulfillment of the objective at the particular time and location of the simulation.

Congestion On Demand Examples

Box Objective The *box objective* creates a box of congestion in the space-time diagram, i.e. congestion will be created on a specific segment of the freeway during a user-specified time interval.

As we have two competing goals (maximize congestion in the box, minimize congestion elsewhere), we apply the multi-objective optimization procedure in Section 4.2.2. Indeed, we have the following two objective functions:

$$f_1(\mathbf{u}, \rho) = -\sum_{(i,k)\in\text{Box}} \rho[i,k] \tag{4.17}$$

$$\text{and } f_2(\mathbf{u}, \rho) = \sum_{(i,k)\notin\text{Box}} \rho[i,k] \tag{4.18}$$

To solve this multi-objective problem, we balance our two objectives using a linear combination. As we limit ourselves to one degree of freedom, we introduce a single parameter $\alpha \in [0, 1]$ and minimize the following objective function:

$$J_\alpha(\mathbf{u}, \rho) = \alpha\, f_1(\mathbf{u}, \rho) \;+\; (1-\alpha)\, f_2(\mathbf{u}, \rho), \tag{4.19}$$

where α is a trade-off parameter: $\alpha = 1$ is complete priority on the congestion inside the box, while $\alpha = 0$ is complete priority on limiting density outside the box.

The results of the box objective are presented in Figures 4.5b-4.5d. We give space-time diagrams for three different values of the parameter α. The box of the objective is shown as a black frame with an actual size of 10 miles and 20 minutes. As the trade-off moves from $\alpha = 0.3$ to 0.9, there is a clear increase in the congestion within the box, at the expense of allowing the congestion to spill outside the desired bounds. In fact, Figure 4.5d ($\alpha = 0.9$) activates the bottleneck near the top-left of the box earlier than Figure 4.5b ($\alpha = 0.3$) to congest the middle portion of the box, which leads to a propagation of a congestion wave outside the bounds of the bottom-right of the box.

As a part of the SmartRoads project, we also implemented the box objective on an Aimsun microsimulation model [6] of the I15 freeway network. This model originates from the I15 integrated corridor management project ran in San Diego in 2010, ref [78]. The geographical location of the I15 network is given in Figure 4.6 and shows San Marcos as the southbound start and Mira Mesa as the end, with the desired box of congestion placed approximately 5 miles before Mira Mesa. A snapshot of the northern and southern extents of the box at the time of 8:30 are shown below the map. The south-bound lanes in the snapshot indicate that congestion was more or less confined to the desired box. A summary

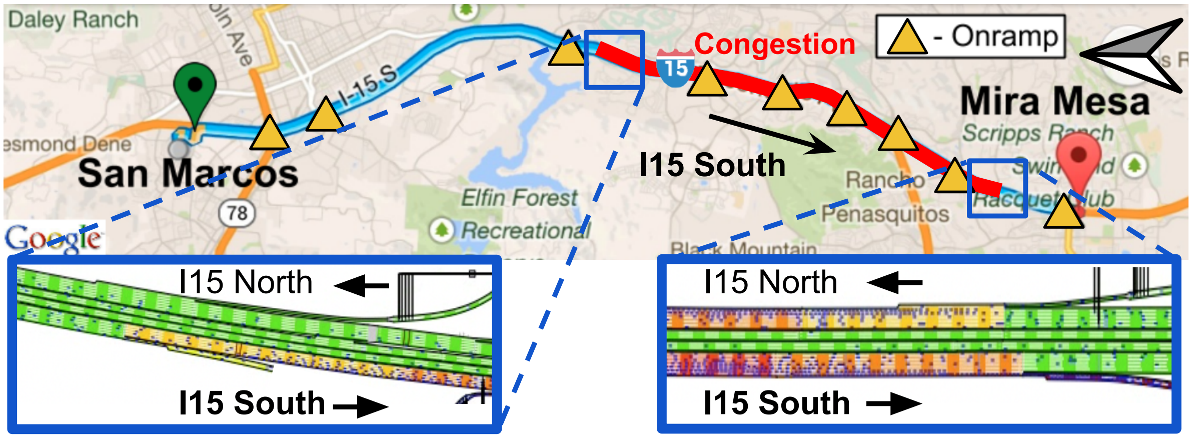


Figure 4.6: Box objective attack implemented on a microscopic model of I15 Freeway produced with Aimsun software. The metering lights were set using the *congestion-on-demand* strategy. Snapshots of traffic at the north and south extents of the box show that the strategy maintains congestion within the box and free-flow conditions outside the box. A link to a video of the microsimulation is provided [102]. Best viewed in color.

video [102] of the I15 microsimulation shows the formation and dissipation of congestion within the predetermined freeway section.

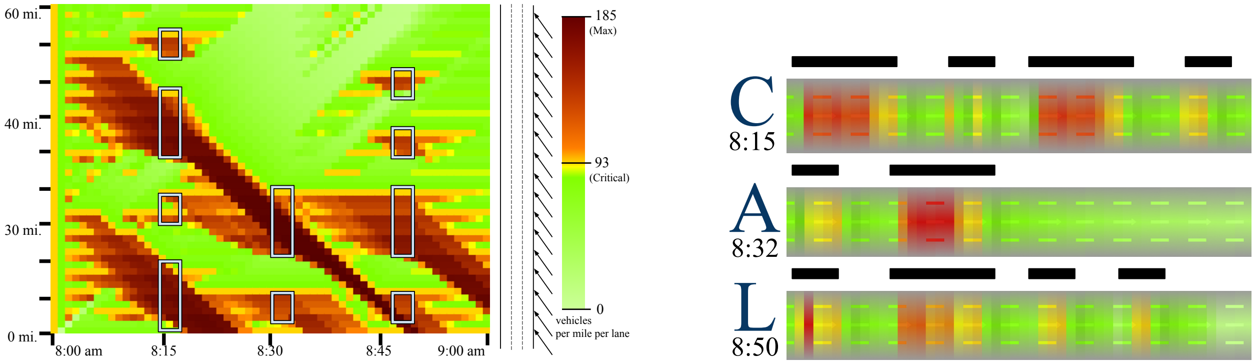
Attack to Create Traffic Patterns in the Form of Morse Code

- **Network** Since the I15 network does not have enough controllable onramps for the following attacks to be precise, we now consider a 60 mile freeway network with onramps and offramps spaced every 3.75 miles and a fixed demand on the onramps.

- **Attack** Figure 4.7 represents the space-time diagram of a *Morse code attack*. The objective is to create the Morse code representation of the three letters "C-A-L"[1], spelled with congestion blocks on the freeway. The corresponding objective function is the superposition of several box objectives on three thin time stripes of the space-time diagram. Everywhere else, the coefficients are put to zero. The result demonstrates that even with a reasonable number of ramps, one can achieve complex attack patterns. In particular, the optimal control approach was able to identify that creating a single backwards-moving jam was the most effective way to produce the second dash for "C", the first dash for "A" and the first dot for "L".

Arbitrary Patterns Provided the right controllability conditions are satisfied, any congestion pattern may be created if the network has enough control ramps. To work towards this, we can choose the negative and positive coefficients of the *congestion-on-demand* method

[1] Short for University of California.



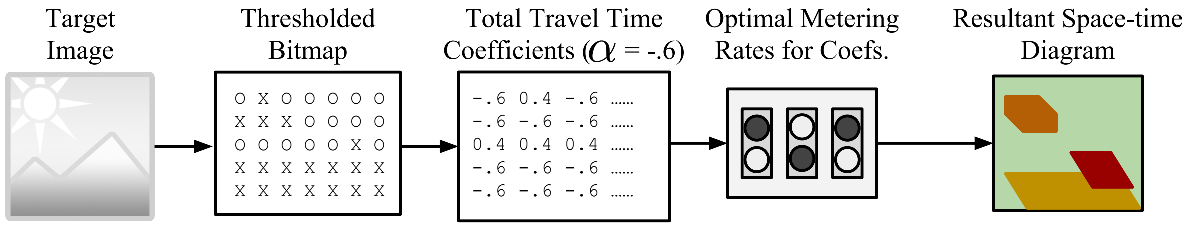
(a) Space-time diagram.

(b) Density profiles over space for three moments in time spaced 17 minutes apart. Best viewed in color.

Figure 4.7: Attack to create traffic patterns in the form of Morse code. A coordinated ramp metering attack using the proposed algorithm is able to spell out "C-A-L" in Morse code over successive time-slices of the space-time diagram: C= − · −·, A= ·−, L= · − ··. The entire space-time diagram of the attack is shown in Figure 4.7a, while three snapshots of the freeway are shown in Figure 4.7b, each slice spelling out one of the letters in "C-A-L" in blocks of congestion.

carefully to match a desired pattern. The following process, as depicted in Figure 4.8, gives a methodological approach to constructing arbitrary *congestion-on-demand* patterns.

One selects some image file they wish to reproduce in congestion patterns on a space-time diagram. The image is thresholded by color intensity to produce a bitmap of regions of desired congestion (X's) and free-flow (O's). Then a *congestion-on-demand* objective (Equation (4.16)) is constructed from the bitmap and scalarized using the α balance parameter to produce the a_i^k coefficients. A metering policy minimizing the objective is then computed using the optimal control method in Section 4.2.1. Given sufficient control of the network and optimization time, the resulting space-time diagram from the metering policy will resemble the input image file.



```
O X O O O O O
X X X O O O O
O O O O O X O
X X X X X X X
X X X X X X X
```

```
-.6 0.4 -.6 ......
-.6 -.6 -.6 ......
0.4 0.4 0.4 ......
-.6 -.6 -.6 ......
-.6 -.6 -.6 ......
```

Figure 4.8: Flow-chart for converting an arbitrary image to a *congestion-on-demand* goal. "Converting" an objective of the form in Equation 4.16 allows an attacker to compute metering rates that produce space-time diagrams resembling the original image.

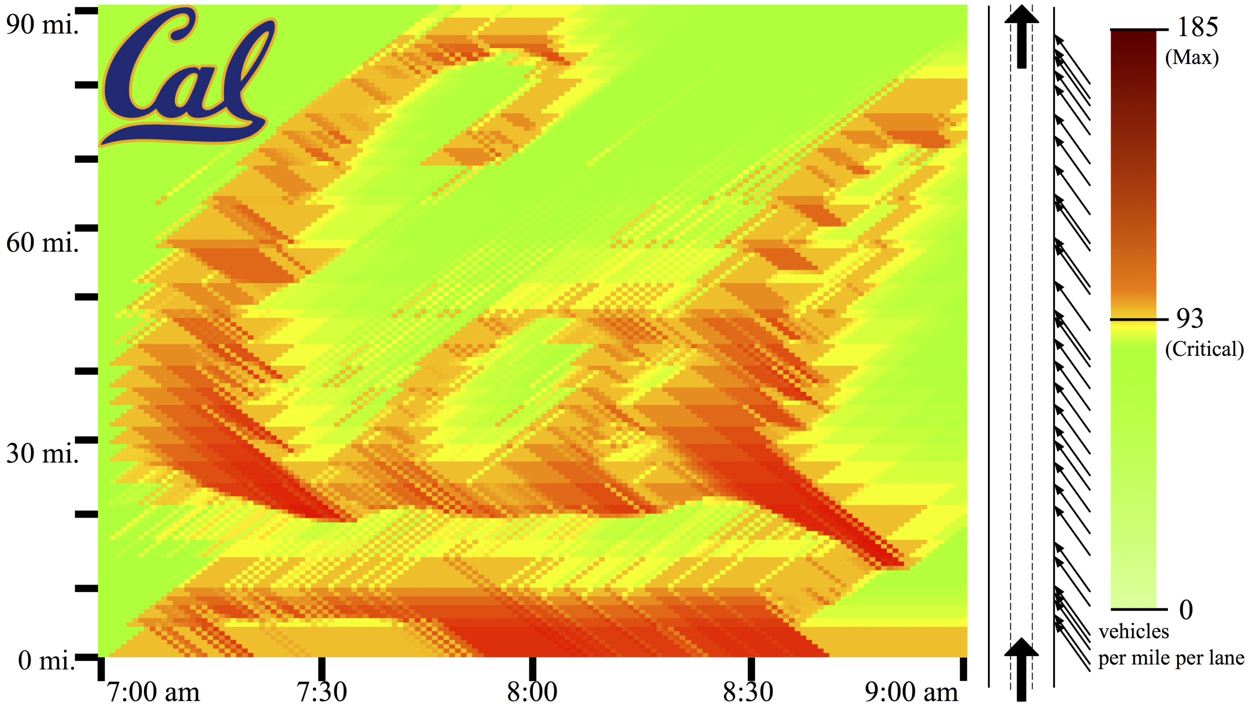


Figure 4.9: Space-time diagram obtained following a *congestion-on-demand* attack with a Cal logo as the objective function. The attack was simulated on a 90 miles and 33-onramp freeway, for a 2 hours simulation time and using coordinated ramp metering.

We give an example of the arbitrary *congestion-on-demand* attack in Figure 4.9, which produces a space-time diagram resembling the Cal logo. See [102] for a online video simulation of the *Cal attack*.

4.2.4 Catch-me-if-you-can Attack

We will now show that the use of the multi-objective optimization methods introduced in Section 4.2.2 can allow the design of more realistic and hard to define attacks. We will consider the example of a vehicle chase, presented in Section 4.1.2. Some vehicles are pursuing the driver along the freeway, while the driver wishes to escape. This objective is distinct from the *congestion-on-demand* attack, as our desired congestion pattern cannot immediately be imagined beforehand and is highly dependent upon the eventual path of the driver.

We translate the attack into a multi-objective problem (see Section 4.2.2). We can split this attack into four simpler and sometimes conflicting goals, each goal associated with an objective function to minimize:

1. The followers (everyone behind the driver) should travel along the freeway section as

slowly as possible — Minimizing f_1 will maximize the traffic density of all freeway sections behind the driver's trajectory.

2. In particular, those vehicles directly behind the driver should be impeded with increased priority — Minimizing f_2 will maximize the traffic density difference between the cells of the driver's trajectory and those cells immediately behind.

3. As to not arouse suspicion from monitoring traffic managers, most other travel times should be reduced — Minimizing f_3 will reduce the total travel time of all the vehicles on the freeway to avoid unnecessary congestion.

4. The driver should quickly exit the freeway — Minimizing f_4 will reduce the driver's travel time, to allow him to travel along the freeway as quickly as possible and escape his followers.

Constructing a trajectory f_2, f_3 and f_4 requires the trajectory of the driver, but reconstructing a vehicle's trajectory using a discretized, macroscopic traffic model is not obvious. We have chosen the following algorithm:

1. The driver's trajectory starts at $t = 0$ and in the first "spatial cell" of the freeway section.

2. The driver's current velocity is computed using the current cell's density.

3. The trajectory, assuming the current velocity, is projected to the next spatial cell.

4. If we are not at the end of the trajectory (in space or in time), we go back to step 2.

This algorithm only gives an approximation of the driver's trajectory, as some resolution is sacrificed in order to have a closed-form expression which permits computation of its partial derivatives.

We have four objective functions. In practice, presenting the results is clearer with only three functions, and we have chosen to keep only f_1, f_2 and f_3 in this chapter, as f_4 was not essential for producing interesting results. We will use the linear scalarization technique presented in Section 4.2.2, and chose three coefficients $a_1, a_2, a_3 \in \mathbb{R}_+$, so that $\sum_{i=1}^{3} a_i = 1$. The objective function we want to optimize is then the following:

$$J(\mathbf{u}, \rho) = \sum_{i=1}^{3} a_i \, f_i(\mathbf{u}, \rho) \tag{4.20}$$

Implementation

Graphical Representation The space-time diagram in Figure 4.10, for a 21 miles freeway with 6 adjacent onramps and a 20 minutes simulation time, is an example output of the

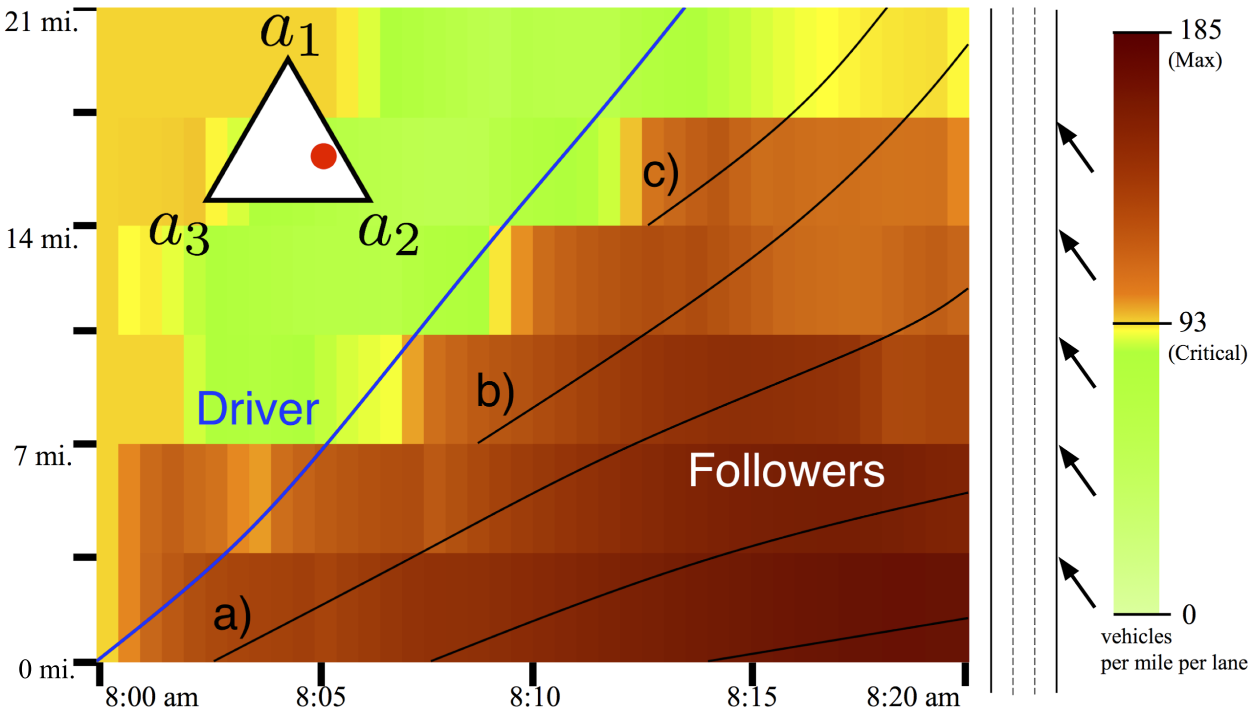
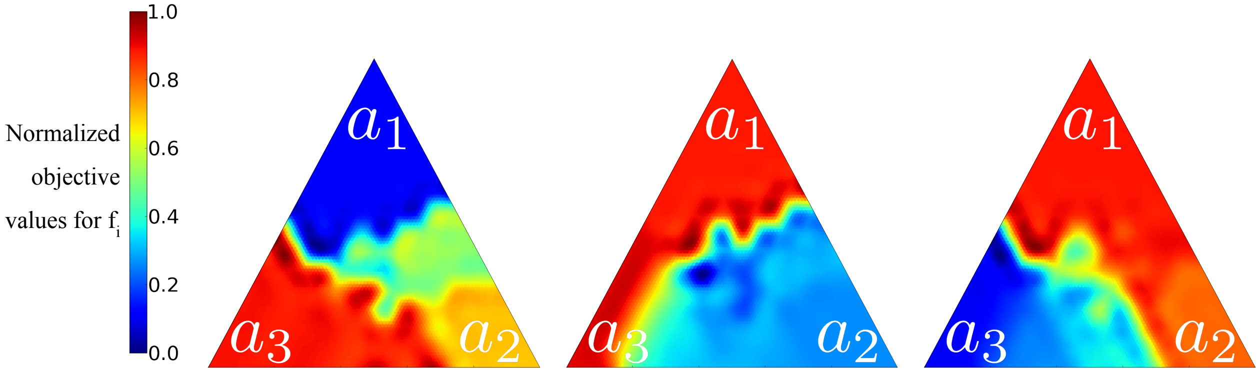


Figure 4.10: Space-time diagram with a ternary graph representing the a_1, a_2, a_3 coefficients (here 30%, 55% and 15% respectively) used for the scalarization process in the catch-me-if-you-can example. The trajectory of the driver (blue line) appears to always gain distance in relation to pursuants further upstream (black lines). Best viewed in color.

optimal control scalarization method. Such plots are useful for the DM to discern between "good" and "bad" simulations produced from metering rates. The driver's trajectory is represented in blue, while the trajectory of three pursuants (a, b, c) are depicted losing ground on the driver.

Ternary Graph The triangle in Figure 4.10 is a visualization of the chosen set of coefficients a_i. The red dot represents the weighted average of the three corners of an equilateral triangle: the closer the red circle is to the a_i corner, the closer a_i is to 1. This is called a *ternary graph*. The top edge will always be a_1, and the right and left a_2 and a_3 respectively. In this example, we can see that the dominant coefficients are a_1 and a_2. As a consequence, we have an significant congestion behind the driver, forming immediately behind him.

A posteriori Method - Grid Exploration Our approach for the a posteriori method is to automatically "explore the triangle of coefficients" to help the *Decision Maker* find a preferred coefficient solution or region of solutions. Figure 4.11 presents the result of the a posteriori method. We plot the values of each objective function for the optimal solution associated with all sets of a_i coefficients. The lowest values of each f_i are always reached with



(b) Scaled values of f_1 (c) Scaled values of f_2 (d) Scaled values of f_3

Figure 4.11: A grid exploration over the ternary graph. An optimization was conducted for a grid of coefficients regularly spaced on the ternary graph. The resulting scalarized objective is decomposed into the constituent objectives (normalized between 0 an 1) and plotted on separate summary ternary graphs.

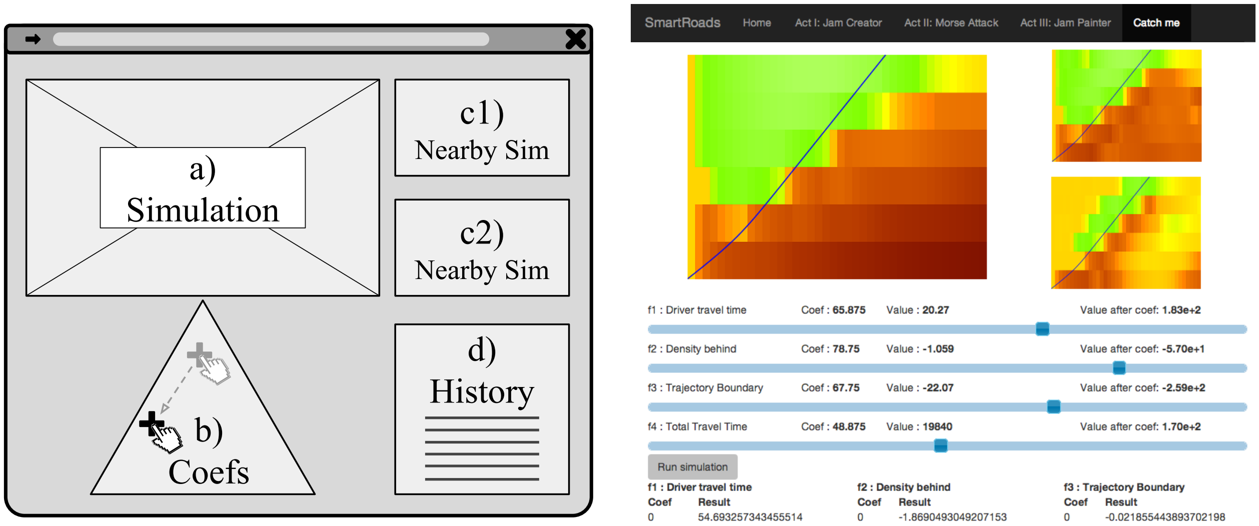
the highest values of a_i (where f_i has been normalized to take values between 0 and 1; see Section 4.2.2). Any non-monotonicity in the graphs are attributed to early terminations of the optimizer's gradient descent or convergence to sub-optimal local minima. The conflicting nature of the objectives is apparent. Figure 4.11b shows that f_1 is penalized more by high a_3 values than by high a_2 values, i.e. lowering the total travel time at the expense of congesting the region behind the driver.

The a posteriori method provides the DM with a global overview of the Pareto front, enabling him to immediately locate a desired solution, or at least identify interesting starting points in the Pareto front. For example, Figure 4.11 gives an indication that the center regions of the triangles have large variations and should be explored further.

Interactive Method A web application[2] (diagram in Figure 4.12) was developed to allow a full exploration of the interactive method. The DM first selects his desired coefficients (a_i) by clicking on the appropriate spot within triangle b). Then, after a scalarization using the particular coefficients and an optimization of the resultant objective, the interface plots the space-time diagram of the resulting simulation in window a), along with the driver's trajectory. Any other vehicle's trajectory can be visualized by clicking at the starting point of the desired trajectory. To enhance the exploration process, the interactive program also chooses two random (but nearby) sets of coefficients and plots their simulation in c1) and c2).

Figure 4.13a shows an overview of the results obtained while using the interactive interface. The first column shows simulations for the corners of the ternary graph, i.e. only one objective is active at a time. The results are intuitive in that optimizing f_1 (Figure 4.13a.1)

[2] Interactive web application demo available at [102].



f1 : Driver travel time Coef : **65.875** Value : **20.27** Value after coef: **1.83e+2**

f2 : Density behind Coef : **78.75** Value : **-1.059** Value after coef: **-5.70e+1**

f3 : Trajectory Boundary Coef : **67.75** Value : **-22.07** Value after coef: **-2.59e+2**

f4 : Total Travel Time Coef : **48.875** Value : **19840** Value after coef: **1.70e+2**

f1 : Driver travel time

Coef	**Result**
0	54.693257343455514

f2 : Density behind

Coef	**Result**
0	-1.8690493049207153

f3 : Trajectory Boundary

Coef	**Result**
0	-0.021855443893702198

(a) Diagram of web application functionality.

(b) Actual web application [102] created for the purpose of this article. The triangle is replaced by 4 sliders, to match the 4 objective functions. Web application [102] is available online for the reader's convenience.

Figure 4.12: Interface of the interactive optimization system used to solve the multi-objective optimization problem to produce the attacks presented in the chapter.

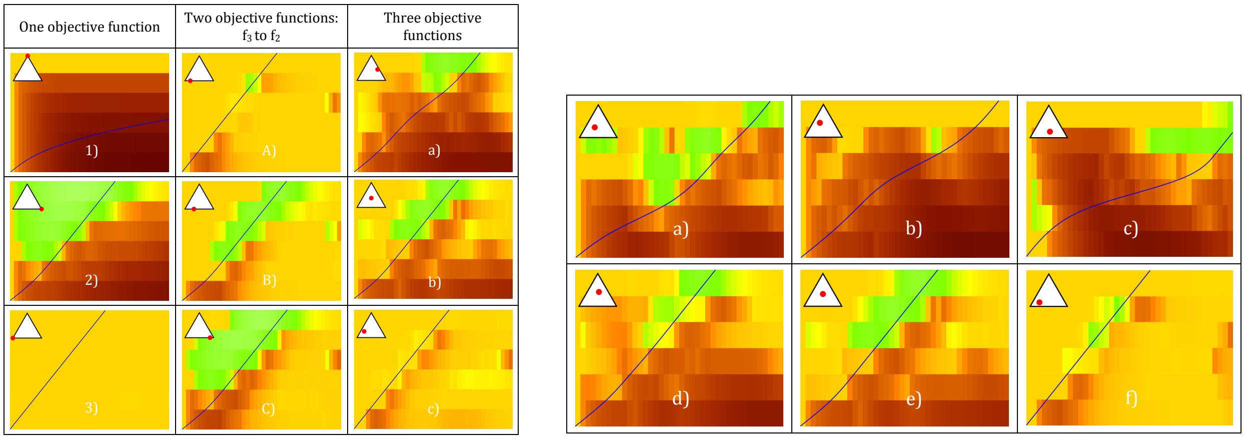
produces congestion everywhere behind the driver, optimizing f_2 (Figure 4.13a.2) creates a distinct increase in congestion behind the driver, and optimizing f_3 (Figure 4.13a.3) maintains critical density everywhere, equivalent to maximizing throughput at maximum freeway speeds.

The second column (Figures 4.13a.A-C) shows an interactive shift from favoring f_3 (minimize travel times) to favoring f_2 (trajectory boundary congestion). The shift progressively limits congestion formation, and intelligently removes more congestion *ahead* of the driver, as to not decrease the delay of pursuant vehicles.

The last column of Figure 4.13a demonstrates how the interactive process allows for fine-tuning of the balance of the objectives. Figure 4.13a.a appears to be overly congested within the driver's trajectory. An interactive progression towards lower total travel times concludes with a desirable congestion boundary in Figure 4.13a.c.

Figure 4.13b shows a few more examples of the coefficient space exploration. It shows result that all seem correct while being very different: this is the definition of the Pareto front. This is why interacting with the DM is necessary to chose the “best” attack.

In particular, examples a), b) and c) of Figure 4.13b were given by the random exploration tool of the interface of Figure 4.12. Their common idea is to hide a clearer path for the driver into a big congestion that covers almost the entire freeway. They are examples of



(a) (b)

Figure 4.13: Summary of *catch-me-if-you-can* simulations generated via the interactive method are shown in Figure 4.13a. Column 1 shows optimizations over individual objectives. Column 2 shows a transition from favoring f_3 to favoring f_2. Column 3 shows a progression across all three objectives. More interesting solutions of the multi-objective problem found via the use of the interface are shown in Figure 4.12.

situations that fulfill the objective (the vehicle escapes, the followers are slowed down), but that would not have been find easily without the interactive process.

4.2.5 Summary of Results

This chapter presents an overview of freeway traffic control systems and their vulnerability to physical and cyber-attacks. The impact of an attack is understood via the response of the control system, with direct attacks on the metering lights being potentially more effective than indirect attacks on the sensing infrastructure. Coordinated ramp metering attacks, being the highest level compromise, are extensively analyzed using methods from the fields of optimal control and multi-objective optimization. The mathematical approach to coordinated attacks on the freeway is explicitly derived for ramp metering applications. Detailed numerical simulations of coordinated ramp metering attacks were conducted to demonstrate the hazards of such compromises and the utility of optimal control tools in not only the hands of traffic managers, but also of adversaries.

Chapter 5

Optimization-based Framework for Rerouting a Subset of Users with Mixed Lagrangian-Eulerian Demand

5.1 Introduction

5.1.1 Traffic assignment: selfish routing vs. social routing

The problem of traffic assignment handles users' route and departure time decisions and how individual behaviors impact the performance of the underlying traffic network. If all user decide in a self-optimizing manner, then the resulting network state is a *user equilibrium* (e.g. [126]). If every user acts in a manner that is beneficial to societal goals, it is said to be a *system* or *social optimum.* Socially optimal schemes are studied under the assumption that a central agency controls *all* the users, while on the other extreme, user equilibrium is is a good model to describe selfish behavior *in the absence* of a central agency. A complete characterization user equilibrium model requires complete information of the origin-destination demands on the network. This information is often too expensive to obtain. Specifically, origin-destination information may only be available for a fraction of users because collecting such information requires participation/consent of the travelers and technological capability of the central agency. [76] give a variational inequality approach to solving user equilibrium, while [91] presents an optimal control framework, and both methods require full information of origin-destination demands on the network.

These technologies can be broadly categorized into two categories. First, there are recommendation systems, such as variable message signs that suggest particular routes based on estimated travel times or general dissemination of information to better inform users of network conditions. Second, there are direct control systems that restrict behavior of users via ramp metering or detours. [51] discusses the effectiveness of ramp metering as a means of achieving a social optimum. These direct control mechanisms are generally applied at a specific point and time and do not distinguish between users who have different routes or destinations. The effectiveness of such active control schemes usually depends on complete origin-destination demands. An exception is that boundary flow demands may be sufficient for evacuation-type problems (e.g. [130]).

5.1.2 Using mobile phones to control routes of individual users

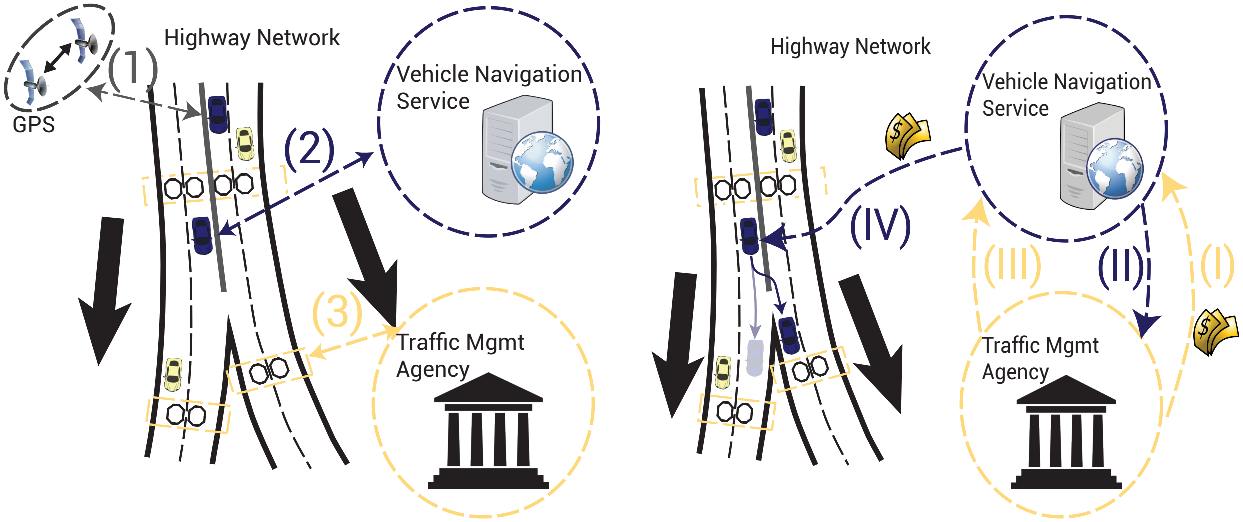
With the emergence of GPS-enabled cell phones and their widespread adoption in populated city areas, a third category of control has become possible: one that communicates directly with users and permits a central agency or a private entity to engage individual users to shift their travel choices. Such a high granularity of control would allow specific origin-destinations or routes to be targeted by the control scheme and could even be customized to the route preferences of the individual users.

Vehicle navigation services that collect, aggregate, and process information from a large number of GPS-equipped mobile devices have become increasingly popular. Such services include Waze, Google Maps, and other such mobile applications. While these services are popular for their utility to individual drivers, the service providers are also able to collect information on behavior of the fraction of users that are equipped with these devices. Once the data has been anonymized to protect the privacy of individual users, the origin-destination information could be interpreted as a subset of the total demand on a network. Additionally, route guidance decisions could be made to benefit their user-base as a whole, rather than on an individual level. [89] discusses the inherent inefficiencies of selfish routing versus the social optimum.

Individually-applied control schemes have many advantages, but a limitation is that the user-base of a particular vehicle navigation service would only constitute a subset of the total users of the network. A significant number of users of the network may not have access to or prefer not to use a GPS-enabled device. Also, a complete understanding of the origin-destination demands on a network by a single entity would still be difficult or very expensive to obtain.

5.1.3 Combining route-based demands with link-level flow information

While collecting route information on individual users suffers from limited penetration, existing traffic monitoring systems, such as loop detectors or cameras, are able to capture all vehicle flows for particular locations on networks. These stationary systems are often monitored by public, traffic management agencies, that are interested in the welfare of all users on the network. It is apparent that the two methods for capturing traffic information are complementary: GPS-based methods have limited penetration but more detailed origin-destination information, while stationary sensors have full penetration of flow, but cannot give route-level information on the demands. Figure 5.1a depicts the vehicle navigation service collecting aggregate GPS data from their "dark" users on the network **(1)**, while the traffic management agency collects flow count data, accounting for *all* users ("dark" and



(a) (b)

Figure 5.1: Route guidance system architecture for using both route-based and link-level flow information. **(a)** An illustration of Lagrangian and Eulerian traffic information collection systems. **(1)** Path-based information is collected via GPS-equipped vehicles, from either onboard route-guidance systems or cell phones. **(2)** Route-based information is sent back to a vehicle navigation service that aggregates traffic information from many GPS-equipped users. **(3)** Eulerian-based loop detectors collect flow counts and send the information to a traffic management agency. **(b)** An illustration of proposed interaction the traffic management agency, vehicle navigation services, and network users. **(I)** Contracting of the vehicle navigation services by the traffic management agency. This may involve monetary compensation or tolling. **(II)** Anonymized Lagrangian information (owned by the vehicle navigation agency) is transferred to the traffic management agency. **(III)** The traffic management service provides route guidance to the vehicle navigation service to improve overall traffic conditions. **(IV)** The vehicle navigation service provides individual network users with alternate route suggestions, with potential incentivization. Users may be guided to switch from their previously preferred (nominal) route.

"light") **(2)**, from the loop detectors embedded in the road. Collectively, the route-based flow data could be used as a source of *re-routable* traffic flow, while the link-based flow data could be used to better estimate the expected travel times that all users will experience before and after re-routing a subset of users. This chapter proposes a method for using both information types to improve traffic conditions across the network. For now, we motivate our work with a scenario in which such a technique could be employed.

Figure 5.1b depicts the scenario in which a traffic management agency partners with a vehicle navigation service to take advantage of their different information sources. Initially, a contractual phase may take place **(I)**, where the public agency compensates the vehicle navigation service for access to their data **(II)**. The information from the vehicle navigation service would be aggregated and anonymized, in order to protect the privacy of the individual users of the service. Then, the traffic management agency would input the route-based demand data and the stationary, link-based loop detector data into the algorithm we have developed (Sections 5.2 and 5.3). The algorithm outputs a new set of aggregated routing suggestions, which are then sent to the vehicle navigation service **(III)**. Finally, the service relays the routing suggestions to their users **(IV)**.

For this last point of communication between the agency and the user, there are two notes. First, it is likely that a fraction of users will be suggested routes with larger travel times than to which they have become accustomed. In order to incentivize the user to accept the route suggestion, the traffic management agency may require the vehicle navigation service to compensate these users, enforceable by the initial contractual agreement between the two organizations. [77] describes an experiment which utilized incentives to move commuters' departure times to less congested times. Alternatively, one can consider that socially optimized routing policies may decrease the travel time for all users *on average*. Then, if all users get assigned desirable routes some days and less desirable routes other days (in order to reduce congestion on desirable routes), then every user could expect to have an improved average travel time. Such an argument could potentially remove the need for monetary compensation or other types of incentivizes. The second note is that we have assumed, given enough incentive from the vehicle navigation service, a user will *always* comply with the suggested route. We do not discuss the method of incentives in this chapter, but note that the assumption can be relaxed by limiting the amount of re-routable flow.

Due to the decoupled nature of the system described in Figure 5.1, we can generalize the scenario to include multiple vehicle navigation services (Figure 5.2). Without sharing information between services, more route-based flow information can be used as input into the algorithm, thus providing more complete information on the origin-destination preferences of the users and collecting a larger pool of re-routable users, while maintaining the privacy of the services which wish to provide socially optimal routes to users.

5.1.4 Accounting for untracked users' response

There are a number of reasons why a user would participate in the socially optimized routing guidance program described above. As already stated, they could be incentivized

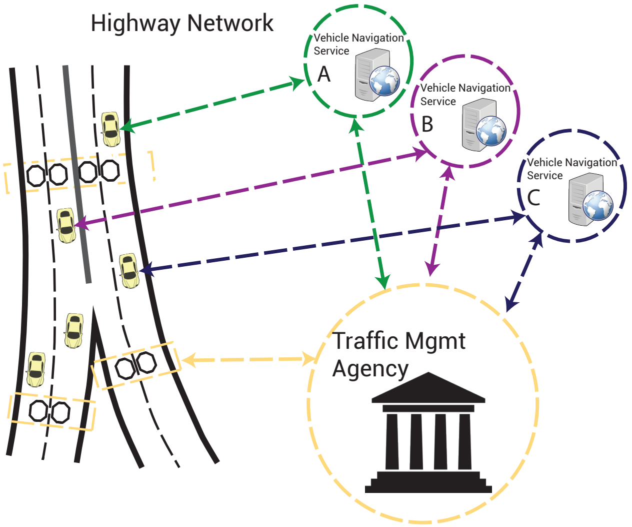

Highway Network
Vehicle Navigation Service
A
Vehicle Navigation Service
B
Vehicle Navigation Service
C
Traffic Mgmt Agency

Figure 5.2: Scenario with multiple vehicle navigation services. Individual service's data is aggregated by the central agency and not shared with other agencies. The routing strategies are calculated by the traffic management agency, and the suggested routes are partitioned between the different user groups.

through monetary compensation. They could also simply be altruistic, and willing to sacrifice personal optimality for the greater good. What is unknown is the behavioral aspects of those users of the network whose route cannot be tracked. How can one predict the response of these untracked (which we refer to as noncooperative later) network users to the routing schemes being implemented for the tracked users?

A standard approach, described as a Stackelberg game (e.g. [109, 70]), assumes that the users outside the control of the central agency will respond with a user equilibrium assignment. Since the origin-destination demands of the untracked users is unknown, solving for a user equilibrium is not possible.

In order to address this lack of information on preferences of untracked users, we develop an alternative model of behavior. Related to the concept of bounded rationality in [56, 57], we assume that the untracked users lack the full information of the state of the network, and cannot make fully rational decisions on their optimal route. Alternatively, the untracked users could possess some *inertia* towards switching routes, and will be content with their previously chosen (nominal) routes, as long as the experienced travel time on the route does not change "too much". This concept of inertia can be practically motivated by considering that some users may appreciate the scenic beauty of a particular suboptimal route, or others have a favorite caf along another route. Thus, in order to reasonably assume that the untracked users will not switch their routes, the routing suggestions provided by the algorithm are guaranteed to not significantly deteriorate the quality of existing routes, beyond an a priori specified bound.

A bounded rationality argument in the context of drivers' route selections was made in [57], where drivers only seek utility gains outside of a certain threshold. [56] give some empirical evidence of bounded rationality on road networks. Our model differs from these because our model lacks origin-destination information on the noncooperative users, and to make this distinction, we refer to our model as *bounded tolerance* model.

5.1.5 Contributions and overview

There is relatively little work done on how partial control schemes can be practically implemented on flow networks. Additionally, inconsistent estimations of traffic between GPS-based data and link-level data can complicate the analysis of the problem. In this chapter, we present a single methodology for accommodating both origin-destination based and link-level flow information for a general, multi-origin, multi-destination, static network (parameters are unchanging with time), while guaranteeing that the two sources of data are consistent with mass balance across junctions (Section 5.2). Furthermore, we present a behavioral model on the untracked users based on the concept of bounded rationality (Section 5.3). This bounded rationality model permits one to cope without origin-destination demands for all users on the network, while still addressing the behavioral aspects of self-routing users.

As our main contribution, we demonstrate how the models presented in this chapter lead to an elegant, optimization-based solution to the socially optimal routing strategy problem (Section 5.4). The optimization problem is proven to be convex for a specific instance of horizontal queues that model highway traffic and extended to a general class of vertical queues . As a corollary, we show that for the discretized LWR network model, the social optimum can be solved exactly for both the purely Eulerian flow and the purely Lagrangian flow cases.

The generality of our method is given by applying the framework to a multiple-destination network with horizontal queues and investigating how changes in the tolerance model impact the routing advice (Section 5.5). The chapter finishes with a conclusion and discussion of the practical importance of the framework and models developed here-within (Section 5.6).

Nomenclature

$\mathcal{L}$	Set of links.
$\mathcal{O} \subset \mathcal{L}$	Set of origins (sources).
$\mathcal{D} \subset \mathcal{L}$	Set of destinations (sinks).
$\mathcal{J}$	Set of junctions.
$\mathcal{R}$	Set of routes.
$r \in \mathcal{R}$	Sequence of contiguous links $\left(r_1, \ldots, r_{\lvert r \rvert}\right) : r_i \in \mathcal{L}$
$\Gamma_j \subset \mathcal{L}$	Set of incoming links for junction $j \in \mathcal{J}$.
$\Gamma_j^{-1} \subset \mathcal{L}$	Set of outgoing links for junction $j \in \mathcal{J}$.
$\mathbf{r}_j \subset \mathcal{R}$	Set of routes passing through junction $j \in \mathcal{J}$.
$f_l, \bar{f}_l$	Flow (resp. nominal flow) on link $l \in \mathcal{L}$.

f_r	Total flow on route $r \in \mathcal{R}$
f_r^{c}	Cooperative flow on route $r \in \mathcal{R}$.
$\rho_l, \bar{\rho}_l$	Density (resp. nominal density) on link $l \in \mathcal{L}$
$\mathbf{f}^{\mathrm{c}}$	Assignment of cooperative flows across all routes $\in \mathcal{R}$.
$\bar{f}_l^{\mathrm{nc}}$	Nominal noncooperative flow on link $l \in \mathcal{L}$.
$\bar{f}_{o,d}$	OD flow demand of cooperative (Lagrangian) users from origin $o \in \mathcal{O}$ to destination $d \in \mathcal{D}$.
$\ell_l, \bar{\ell}_l$	Latency (resp. nominal latency) on link $l \in \mathcal{L}$.
α	Tolerance scale factor.

5.2 Modeling partial cooperation with Lagrangian-Eulerian demands

We present the general setting of the routing problem considered, as summarized in Figure 5.3. Consider a setting in which a subset of users are equipped with GPS-enabled devices and are connected to a central coordinator through a *Routing interface* (e.g. a mobile phone application). We refer to this subset as *cooperative* users. First, the cooperative users provide their desired routes to the coordinator through the routing interface. This allows the coordinator to have individual route information, i.e. *Lagrangian information* for the cooperative users. Second, the loop-detectors (or other sensors capable of measuring aggregate link-level flows) provide Eulerian information. We refer to the historical estimates of Lagrangian and Eulerian information as the *nominal* state of the network.

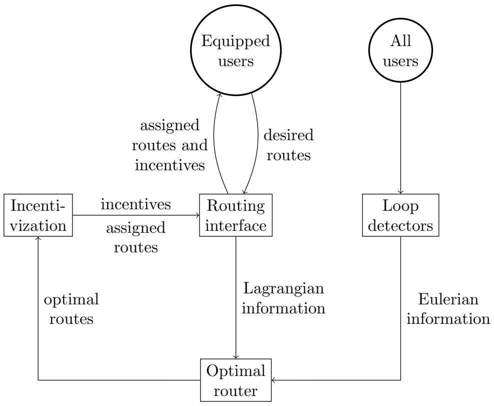


Figure 5.3: Data-flow diagram

Given the nominal Eulerian flow measurements for the entire network and the nominal Lagrangian information for the equipped vehicles, the central coordinator determines the optimal route assignment for the equipped vehicles (Section 5.2.3). This optimization problem is represented by the *optimal router* block. Since only the cooperative users follow the optimal route assignments provided by the central coordinator, we will refer to this problem as a *partial cooperation* problem.

The next step is an *incentivization* step: given the target optimal routes, and possibly additional constraints (such as a total available budget) a second problem (not discussed in this chapter) determines an incentive for each equipped vehicle and the corresponding target route. The incentivization problem is outside of the scope of the present chapter. More information on how to solve incentivization and traffic demand management can be found in [74].The assigned routes and the corresponding incentives are then offered to the equipped drivers, who can either accept or refuse the offer. The subset of vehicles that do accept the offer (thus taking the route assigned by the central coordinator) are called *cooperating vehicles*. In the present chapter, we focus our attention on the optimal route assignment with information on mixed Lagrangian-Eulerian demands.

Considering the route optimization goals stated above, we give a declaration of the problem statement to direct the model development of the proceeding sections.

> **Problem statement:** Find a mathematical framework for flow networks which can encompass:
>
> - Two different types of demand information: Lagrangian information, which is specified by the route traversed by the flow, and Eulerian information, which is specified by the flow-count across a link.
> - Socially optimal routing strategies which can encompass both information types, given their limitations:
>   - Lagrangian information is only known for the cooperative flow, which can be rerouted from its nominal route to improve network conditions.
>   - Only Eulerian information is known for the noncooperative flow, which is assumed to maintain its nominal state.

5.2.1 Network model

Using standard network notation, the network model is defined by the tuple, $(\mathcal{L}, \mathcal{J})$, where $\mathcal{L}$ is the set of links, and $\mathcal{J}$ is the set of junctions. A junction $j \in \mathcal{J}$ has a set of incoming links $\Gamma_j \subseteq \mathcal{L}$ and outgoing links $\Gamma_j^{-1} \subseteq \mathcal{L}$. An origin $o \in \mathcal{O} \subseteq \mathcal{L}$ is a link with no upstream junction. A destination $d \in \mathcal{D} \subseteq \mathcal{L}$ is a link with no downstream junction. A route $r = \left(r_1, \ldots, r_{|r|}\right) \in \mathcal{R}$ is a set of adjacent links where $r_1 \in \mathcal{O}$, $r_{|r|} \in \mathcal{D}$, and $\forall i \in (1, \ldots, |r| - 1)$, $\exists j_i^r : r_i \in \Gamma_{j_i^r}, r_{i+1} \in \Gamma_{j_i^r}^{-1}$.

5.2.2 Cooperative demand vs. total demand

Let the network in Section 5.2.1 contain flows f_l on every link $l \in \mathcal{L}$. Furthermore, we assume that the network is in steady state, i.e. all state on the network is stationary with respect to time (e.g. flows). We further differentiate two types of demands: Lagrangian and Eulerian.

- We assume that the cooperative users have provided their desired origin and destination. Therefore, for every origin-destination pair $(o, d) \in \mathcal{O} \times \mathcal{D}$, there is a nominal flow demand $\bar{f}_{o,d}$ from the cooperative users, where the bar notation refers to nominal state values. Since this type of demand concerns the routes taken by the flow, we describe this type of demand as *Lagrangian* demand.

- For the noncooperative users, i.e., the users who do not (or choose not to) interact with the routing interface, we do not assume knowledge of Lagrangian demand. Thus, we assume that the only the aggregate link-level flows are available via loop detectors. This aggregate level information does not include OD and route information, and is therefore defined as *Eulerian* demand.

To recover the nominal Eulerian demand of the noncooperative vehicles, we further assume that the nominally used routes of the cooperative vehicles are known. For each link $l \in \mathcal{L}$, we specify a nominal total link flow $\bar{f}_l$, and for each route $r \in \mathcal{R}$, we can specify a nominal route flow for cooperative vehicles, $\bar{f}^{\mathrm{c}}_r$. Then, the nominal noncooperative Eulerian demand, $\bar{f}^{\mathrm{nc}}_l$, is obtained for each link $l \in \mathcal{L}$ by subtracting cooperative flow from the total link flow:

$$\bar{f}^{\mathrm{nc}}_l = \bar{f}_l - \sum_{r|l \in r} \bar{f}^{\mathrm{c}}_r \tag{5.1}$$

For the remainder of the chapter we use the noncooperative link flows $(\bar{f}^{\mathrm{nc}}_l, l \in \mathcal{L})$ as the input data for nominal flow, but it is understood that this data is derived from the more practically measurable *total* nominal flow values $(\bar{f}_l, l \in \mathcal{L})$ and the cooperative nominal route flows $(\bar{f}^{\mathrm{c}}_r, r \in \mathcal{R})$ via Equation (5.1).

Since we have subtracted off the flow of nominal cooperative flow to obtain the noncooperative flow, we study properties of the network flow when the rerouted cooperative flow is added back into the network. We introduce the decision variable: f^{c}_r, the amount of cooperative flow assigned to route $r \in \mathcal{R}$. To enforce that the entire flow across a link is accounted for and same origin-destination demands of the cooperative users are satisfied, we have the following constraints:

$$\sum_{r|o,d \in r} f^{\mathrm{c}}_r = \bar{f}_{o,d} \qquad \forall o \in \mathcal{O}, d \in \mathcal{D} \tag{5.2}$$

$$f_l = \sum_{r|l \in r} f^{\mathrm{c}}_r + \bar{f}^{\mathrm{nc}}_l \quad \forall l \in \mathcal{L} \tag{5.3}$$

where $\bar{f}_{o,d} = \sum_{r|o,d \in r} \bar{f}^{\mathrm{c}}_r$ is cooperative flow between origin o and destination d.

A requirement of the Eulerian flow is that noncooperative flow must be conserved across junctions. If the flow across a link $l \in \mathcal{L}$ is f_l, then the following must hold:

$$\sum_{l \in \Gamma_j} f_l = \sum_{l \in \Gamma^{-1}_j} f_l \quad \forall j \in \mathcal{J} \tag{5.4}$$

Since we partitioned flow on each link into two classes (cooperative and noncooperative), flow conservation must hold across both classes independently. We will see shortly that flow conservation across the cooperative class will be guaranteed by the condition that all cooperative flow must be assigned to a route. Then, for the noncooperative class of nominal flow, we must always have the condition that flow conservation holds across junctions. Since this is a condition on the input to the problem we only state it here once and assume the condition for the rest of the chapter.

Model Consistency Condition: For every junction $j \in \mathcal{J}$, we assume: $\sum_{l\in\Gamma_j} \bar{f}_l^{\mathrm{nc}} = \sum_{l\in\Gamma_j^{-1}} \bar{f}_l^{\mathrm{nc}}$.

Equations (5.2)-(5.4) define a route-allocation policy $\mathbf{f}^{\mathrm{c}} = \{f_r^{\mathrm{c}} : r \in \mathcal{R}\}$ for all cooperative users that satisfies all demand requirements. There are three main requirements that we have from the set of constraints: non-compliant (Eulerian) demand is satisfied, compliant (Lagrangian) demand is satisfied, and mass balance across junctions is satisfied. The first two are obvious from the above constraints, while the third one needs proof.

Proposition 5.1. *For a feasible* $\mathbf{f}^{\mathrm{c}}$ *to the set of Equations (5.2) and (5.3),* $\forall j \in \mathcal{J}$, $\sum_{l\in\Gamma_j} f_l = \sum_{l\in\Gamma_j^{-1}} f_l$.

Proof. From the model consistency condition above, we only need to prove the following statement:

$$\sum_{l\in\Gamma_j}\sum_{r|l\in r} f_r^{\mathrm{c}} = \sum_{l\in\Gamma_j^{-1}}\sum_{r|l\in r} f_r^{\mathrm{c}}$$

Let $\mathbf{r}_j^{\mathrm{in}}$ be the routes that pass through links in the incoming links of junction j. Let $\mathbf{r}_j^{\mathrm{out}}$ be the same for outgoing links. Then $\mathbf{r}_j^{\mathrm{in}} = \{r \in \mathcal{R} : \Gamma_j \cap r \neq \emptyset\}$. We also know that by the definition of a route, any route that passes through an incoming link of a junction (not a source or sink) must pass through an outgoing link, and therefore $\mathbf{r}_j^{\mathrm{in}} \subseteq \mathbf{r}_j^{\mathrm{out}}$. A similar argument can be made to show $\mathbf{r}_j^{\mathrm{out}} \subseteq \mathbf{r}_j^{\mathrm{in}}$. This shows that $\mathbf{r}_j^{\mathrm{in}} = \mathbf{r}_j^{\mathrm{out}}$. Then,

$$\sum_{l\in\Gamma_j}\sum_{r|l\in r} f_r^{\mathrm{c}} = \sum_{\mathbf{r}_j^{\mathrm{in}}} f_r^{\mathrm{c}} = \sum_{\mathbf{r}_j^{\mathrm{out}}} f_r^{\mathrm{c}} = \sum_{l\in\Gamma_j^{-1}}\sum_{r|l\in r} f_r^{\mathrm{c}}$$

□

5.2.3 Reducing total latency by rerouting cooperative users

We now formulate the problem of minimizing total latency (or equivalently, total travel time) with route assignments of cooperative vehicles as the decision variable. There are two classes of latency functions studied in the literature: first [87, 109, 90], where the link latency is assumed to be the function of flow in the link, and second [120, 76, 23, 75, 106], where density is assumed to affect link latencies. We generically introduce latency as a value ℓ_l associated with the state and properties of a link $l \in \mathcal{L}$ and discuss the different models of latency as it pertains to different flow models in Section 5.4. We can therefore express the total latency on a link as the flow times the latency, or $f_l\ell_l$.

We can now express a general form of the Lagrangian-Eulerian flow, route assignment problem in a standard optimization program formulation:

$$\begin{array}{rll} \underset{f_l | l \in \mathcal{L}, f^{\mathrm{c}} | r \in \mathcal{R}}{\text{minimize}} & \sum_{l \in \mathcal{L}} f_l \ell_l & \\ \text{subject to:} & & \\ f_l = & \sum_{r | l \in r} f_r^{\mathrm{c}} + \bar{f}_l^{\mathrm{nc}} & \forall l \in \mathcal{L} \\ \sum_{r | o,d \in r} f_r^{\mathrm{c}} = & \bar{f}_{o,d} & \forall o \in \mathcal{O}, d \in \mathcal{D} \end{array} \tag{5.5}$$

where the objective represents the total latency in the network, the first constraint relates cooperative route flows and noncooperative link flows to total link flows, and the last constraint states that the compliant route flows must be partitioned in a way that satisfies the nominal origin-destination demand.

5.3 Accounting for response of noncooperative demand via bounded tolerance

Recall from our discussion in Section 5.2, that due to imperfect perceptions of the travel times, the travelers forming the non-cooperative demand may be assumed to be boundedly rational, and may not change their nominal paths. This assumption is especially valid when the travel times of non-cooperative users do not significantly change when the cooperative users are routed in a socially optimal manner. In contrast, if the cooperative vehicles cause an excessive increase in latency on some routes for noncooperative vehicles, then this assumption may not be realistic (see [4, 109, 71]). For this reason, we have to enhance the model for rerouting cooperative vehicles under stricter conditions.

5.3.1 Bounded tolerance

A traditional approach to predicting vehicle route choice comes from the field of traffic assignment (e.g. [126, 76, 91]) and Nash equilibria in game theory (e.g. [110, 90]), often described as *user equilibrium* in the context of traffic assignment and introduced in [126]. The congestion games literature considers Stackelberg games, which are used to analyze how selfish users respond to a centrally-controlled subset of users in a principled way (see [70, 109]). Our approach in this chapter is simpler. The reasonability of our approach can be argued using a bounded tolerance assumption on the part of noncooperative vehicles. We replace the assumption of stationarity with a stronger assumption that stationarity is only achieved if all routes on the network do not have a latency increase greater than a certain amount, proportional to the nominal latency experienced before rerouting the cooperative vehicles. Tolerance is assumed in the sense that if latencies on a route do not noticeably increase, then noncooperative vehicles do not seek better paths. However, the tolerance to increase delay is still limited in the sense that as latency increases on a route, noncooperative vehicles will eventually switch. The term *tolerance* is used to address the fact that the bounded rationality models assumed in [57, 56] allow decisions to be made by the noncooperative users, while our bounded tolerance model specifies how much perturbation of the nominal is allowed, before the assumption the assumption is no longer valid that noncooperative users do not change routes.

5.3.2 Modeling bounded tolerance

The discussion in Section 5.3.1 dictates that one must have knowledge about a nominal network state with which to compare final network state conditions. Therefore, we introduce as input, the nominal latency $\bar{\ell}_r$ for every route $r \in \mathcal{R}$. We then select as a model of bounded tolerance the condition that the final latency on a route may not be a factor $(1+\alpha)$ greater than the nominal latency $\bar{\ell}_r$, where $\alpha \in \mathbb{R}_+$ can be seen as the *tolerance scale factor.* The latency on a route can be computed by summing the latencies experienced on all links in $r = \left\{r_1, \dots, r_{|r|}\right\}$:

$$\ell_r = \sum_{l \in r} \ell_l$$

We can now express the bounded tolerance condition constraints:

$$\sum_{l \in r} \ell_l \quad \leq (1+\alpha)\bar{\ell}_r \quad \forall r \in \mathcal{R} \tag{5.6}$$

Adding this constraint to Problem 5.5 completes the *partial cooperation, bounded tolerance* problem. Section 5.4 describes how this problem applies to different flow models and latency models. We now introduce another tolerance model and describe the class of problems to which it can be applied.

5.3.3 Comparative tolerance

The model for bounded tolerance described above places a limit on the increase of latency on a particular route. An alternative approach would be to limit the increase in utility that alternative routes gain over a particular route. In other words, the model developed in Section 5.3.2 assumes that a particular route flow would be complacent on its original route as long as its own latency does not increase too much, *while not considering the possibility that the utilities of alternative routes may have increased significantly.* To address this limitation, we introduce a *comparative tolerance* model and discuss the underlying modeling assumptions.

We first assume for a given route $r \in \mathcal{R}$ with origin $o_r \in \mathcal{O}$ and destination $d_r \in \mathcal{D}$, and assuming a tolerance scale factor $\alpha = 0$ (no tolerance to delays induced by cooperative flows), that the allowable difference between the route's final latency and the final latency of all other routes sharing the same origin and destination is the largest difference in nominal latencies between itself and all other routes, or 0 if the considered route has the smallest nominal latency. Then, if the scale factor α is greater than 0, this allowable difference is increased by $\alpha\bar{\ell}_r$. Mathematically, we can express this condition as follows:

$$\begin{aligned} \ell_r - \ell_{\bar{r}} \leq \quad & \max\big(0, \max_{\tilde{r}|o,d\in\tilde{r},\tilde{r}\neq r}\big(\bar{\ell}_r - \bar{\ell}_{\tilde{r}}\big)\big) + \alpha\bar{\ell}_r \quad && \forall \bar{r} : o,d \in \bar{r}, \bar{r} \neq r \\ & && \forall r : o,d \in r, \forall o \in \mathcal{O}, \forall d \in \mathcal{D} \end{aligned}$$

While this formulation more accurately captures the concept of traveler behavior under improved comparative information about alternative routes, it introduces many more constraints than the bounded tolerance formulation in Section 5.3.2. Additionally, since the LHS of the constraint is a less-than inequality that contains the subtraction of two functions of decision variables, common assumptions on the latency functions will typically lead to a non-convex constraint.

One assumption that will guarantee convexity of the above constraints is if all links have affine latency functions. It can be seen by considering that the LHS is the summation of link latencies along a particular route, and the RHS is a constant that can be computed a priori. In Section 5.5.1 we will give a numerical example of a simple network with linear latency functions, comparing the output of our model assuming first simple bounded tolerance, and then considering comparative tolerance.

5.4 Formulating bounded tolerance as a convex optimization problem

The preceding sections discussed a generic model for route-based flow optimization on a flow network with mixed Lagrangian-Eulerian demands, without identifying any specific flow model. In this section, we discuss two types of flow models, horizontal queues and vertical queues. We show for each case how the modeling assumptions can be made into convex constraints, enabling one to solve the partial cooperation, bounded tolerance model as a convex optimization problem. We begin our discussion showing how vertical queues fit cleanly within our model (Section 5.4.1). However, modeling horizontal queues (e.g. highway networks) requires some additional theoretical setup. What has worked for modeling internet, supply chains, etc. does not work for highway networks, as they are nonlinear systems with non-convex constraints that depend on the density of the links, rather than the flows. Its discussion constitutes the bulk of the section (Section 5.4.2).

5.4.1 Vertical queues

Several types of networks, such as communication networks or machine queues (e.g. [109]), can model link latencies as a function of the aggregated flow on the link. To contrast with the model discussed in Section 5.4.2, we refer to such networks as *vertical queues*. In this section, we show an example of how the concepts of partial cooperation and bounded tolerance can be modeled as a convex optimization program for a specific class of vertical queues, and give a brief discussion on how the results extend to a more general class of vertical queues.

Example: M/M/1 queueing model

A common way to model latencies for communication networks is the M/M/1 queue (e.g. [4]), which assumes Poisson arrivals and exponential service times. On a link $l \in \mathcal{L}$, the average latency as a function of the rate of Poisson arrivals (the flow) f is given by the equation:

$$\ell_l(f) = \frac{\beta_l}{\mu_l - f} \tag{5.7}$$

where β_l is the occupation rate and μ_l is the processing rate. The flow on a link must be less than the processing rate for the system to be stable. The function $\ell_l(\cdot)$ is convex in $[0, \mu_l)$. We can now substitute Equation (5.7) into (5.5) and (5.6) to obtain the following program:

$$\begin{array}{lrll}
\underset{f_l | l \in \mathcal{L}, f_r | r \in \mathcal{R}}{\text{minimize}} & & \sum_{l \in \mathcal{L}} \frac{\beta_l f_l}{\mu_l - f_l} & \\
\text{subject to:} & & & \\
& \sum_{l \in r} \dfrac{\beta_l}{\mu_l - f_l} \leq & \bar{\ell}_r (1 + \alpha) & \forall r \in \mathcal{R} \\
& f_l = & \sum_{r | l \in r} f_r^{\text{c}} + \bar{f}_l^{\text{nc}} & \forall l \in \mathcal{L} \\
& \displaystyle\sum_{r | o,d \in r} f_r^{\text{c}} = & \bar{f}_{o,d} & \forall o \in \mathcal{O}, d \in \mathcal{D}
\end{array} \tag{5.8}$$

where again, α is given and corresponds to the maximal threshold tolerable by users if Lagrangian (cooperative) demand perturbs the nominal flow. This program is convex and can be solved by standard convex solvers (with some algebraic manipulations for disciplined convex programming solvers). Indeed, the objective is the summation of convex functions, and the first constraint is a convex inequality *(less-than* inequality with a summation of convex functions on the LHS).

Class of convex vertical queues

This section shows that if all link latencies are convex, increasing functions of flow (e.g. following the modeling assumptions of [109]), then the partial cooperation, bounded tolerance problem is guaranteed to be convex.

From the discussion in Section 5.4.1, we can generalize the class of latency functions for vertical queues, which lead to a convex formulation. In Equation (5.5), only the objective contains latency terms, and in Equation (5.6), the LHS of the inequality contains latency terms. Therefore, we need to verify the convexity of the objective and the bounded tolerance constraints.

A well known-result of convex analysis is that the summation of convex functions preserves convexity. Therefore, the convexity of the bounded tolerance constraint is guaranteed if the latency function on every link is convex. Additionally, for a link $l \in \mathcal{L}$, the total link latency, $f_l \ell_l(f_l)$, is convex from the assumptions on ℓ_l, and therefore the sum of all total link latencies (which is the objective) is guaranteed to be convex.

5.4.2 Horizontal queues

A standard assumption in transportation networks is that latencies are not determined uniquely by the flow on the link, but rather how densely populated the link is. Such latency models are referred to as *horizontal queue*s, as the congestion on a link occupies physical space which may propagate in the horizontal direction. In this section, we show how the partial cooperation and bounded tolerance models can be extended to horizontal queues. First we develop the relationship between link flow and link density, the resulting latency model, and how a convex optimization problem can be formulated for networks with horizontal queues.

Link model

Constraining flow by link densities For each horizontal queuing link $l \in \mathcal{L}$, in addition to having a link flow f_l, a horizontal queue link also has a density of vehicles ρ_l, expressing the number of vehicles occupying a link divided by the length L_l. Relating the density of a link to its flow, each link also has a trapezoidal fundamental diagram specified by three parameters: free-flow velocity v_l, congestion velocity w_l, and max flow $f_l^{\max}$. From these parameters, one can compute the critical density ρ_l^{c} and jam density $\rho_l^{\max}$. Given that we are assuming the network is in equilibrium, then outflow must equal inflow for each link (see [51] for a detailed analysis of horizontal queue equilibria). Therefore, only need to consider the single variable f_l when analyzing flow on a link, rather than considering both the inflow and outflow of a link. We express the ρ_l (as traditionally assumed by the LWR equation [23, 24, 75, 106]), we have two coupled variables f_l and ρ_l, with the following constraints:

$$f_l \leq v_l \rho_l \tag{5.9}$$

$$f_l \leq w_l \ (\rho_l^{\max} - \rho_l) \tag{5.10}$$

$$0 \leq f_l \leq f_l^{\max} \tag{5.11}$$

where (5.9) restricts the outflow of link l, (5.10) restricts the inflow, and (5.11) is a physical capacity of the link. These constraints are a relaxation of the fundamental diagram, initially introduced by [50].

Latencies For a link $l \in \mathcal{L}$, the latency ℓ_l is obtained by multiplying the length and velocity of the link, where the velocity of the link is a function of both flow and density. With a notational change (the latency now depending on two variables), the latency function is given by:

$$\ell_l(f, \rho) = \frac{L_l \rho}{f}$$

and the total latency $f_l \ell_l(f_l, \rho_l)$ on a link is given simply by the number of vehicles on a link, or $L_l \rho$. Note that a nominal link latency must be determined from both a nominal flow and nominal density, requiring more information than the point queue model, which only needs nominal link flows.

Relaxation of Junction Model

In order to guarantee uniqueness of solutions of junction problems in LWR networks, it is common to assume that the sum of flows across junctions is maximal, while respecting the prescribed turning ratios. [24] describes a junction model for 2-to-1 merges and 1-to-2 diverges tailored to the CTM model, while [42] describe a more general junction model allowing n-to-m merges for the continuous LWR network model, which includes the Daganzo model as a special case with triangular fundamental diagrams and limited merge/diverge types. We refer to the flow-maximizing junction models as the *unrelaxed* junction model, and the flow-density relationship in Section 5.4.2 as the *relaxed* junction model as it does not include a flow maximization condition.

One technical reason why the relaxed model is used is that a flow-maximization condition would lead to a non-convex problem formulation. Another argument that can be made is that for certain junction types, some *split-ratio vector* or *priority vector* (see [20]) may exist that would lead to the flow solution given by the optimization problem. Therefore, since this problem has no fixed split-ratios, it can be considered a free variable and the optimization problem has discovered one of the many possible solutions to some junction. This argument has limits, as there is no such free parameter for 1-to-1 junction types, for instance.

There have been methods proposed for dealing with the implicit “car holding” issue introduced from the relaxation, such as adding penalty terms in the objective (see [130]), but we do not consider these in our analysis.

Optimization program

For the horizontal queues network, we can now express the total latency minimization problem expressed in Section (5.3.2):

$$
\begin{array}{rll}
\underset{f_l,\rho_l|l\in\mathcal{L},f_r|r\in\mathcal{R}}{\text{minimize}} & \sum_{l\in\mathcal{L}} L_l\rho_l & \\
\text{subject to:} & & \\
\sum_{l\in r}\dfrac{L_l\rho_l}{f_l} \le & (1+\alpha)\bar{\ell}_r & \forall r\in\mathcal{R} \\
f_l \le & v_l\rho_l & \forall l\in\mathcal{L} \\
f_l \le & w_l(\rho_l^{\max}-\rho_l) & \forall l\in\mathcal{L} \\
0 \le & f_l \le f_l^{\max} & \forall l\in\mathcal{L} \\
f_l = & \sum_{r|l\in r} f_r^{\mathrm{c}} + \bar{f}_l^{\mathrm{nc}} & \forall l\in\mathcal{L} \\
\displaystyle\sum_{r|o,d\in r} f_r^{\mathrm{c}} = & \bar{f}_{o,d} & \forall o\in\mathcal{O}, d\in\mathcal{D}
\end{array}
\tag{5.12}
$$

This formulation is not convex, specifically the bounded tolerance constraint is not convex. There is a superseding problem with the formulation, that the bounded tolerance constraints and outflow constraints ($f_l \le w_l(\rho_l^{\max} - \rho_l)$) are guaranteed to be non-binding. Several of the constraints can be shown to be non-binding by observing that a solution must satisfy $\rho_l = \frac{f_l}{v_l}$. We prove that next.

Lemma 5.1. *If the solution* $\mathbf{f}^{\mathrm{c}*}, \rho^*$ *is optimal for Problem 5.12, then* $\forall l \in \mathcal{L}$*:* $\rho_l^* = \frac{f_l^*}{v_l}$

Proof. Assume $\exists \rho_l > \frac{f_l^*}{v_l}$. Reducing ρ_l to $\frac{f_l^*}{v_l}$ only decreases the LHS of the first constraint in Problem (5.12). The second constraint becomes an equality by construction. The RHS of the third constraint increases. Since the flow terms are not changed, we see that the feasibility of the problem is maintained. Additionally, the objective strictly decreases, thus proving that a solution with such a ρ_l is sub-optimal. □

We can now simplify Problem 5.12 by substituting in the value of ρ from Lemma 5.1, and using the following notational change for the parameters L_l and $a_l = \frac{L_l}{v_l}$:

$$
\begin{array}{rl}
\underset{f_l|l\in\mathcal{L},f_r^{\mathrm{c}}|r\in\mathcal{R}}{\min} & \sum_{l\in\mathcal{L}} a_l f_l \\
\text{subject to:} & \\
\sum_{l\in r} L_l v_l \le (1+\alpha)\bar{\ell}_r & \forall r\in\mathcal{R} \\
f_l \le v_l\left(\frac{f_l}{v_l}\right) & \forall l\in\mathcal{L} \\
f_l \le w_l\left(\rho_l^{\max} - \frac{f_l}{v_l}\right) & \forall l\in\mathcal{L} \\
0 \le f_l \le f_l^{\max} & \forall l\in\mathcal{L} \\
f_l = \sum_{r|l\in r} f_r^{\mathrm{c}} + \bar{f}_l^{\mathrm{nc}} & \forall l\in\mathcal{L} \\
\sum_{r|o,d\in r} f_r^{\mathrm{c}} = \bar{f}_{o,d} & \forall o\in\mathcal{O}, d\in\mathcal{D}
\end{array}
\tag{5.13}
$$

We can now detect non-binding constraints easily. The first constraint in Problem (5.13) must be satisfied because the LHS is the free-flow travel time of the route and is minimal, while the RHS must be greater or equal to free-flow (keeping in mind $\alpha \geq 0$). The second constraint is always an equality, by Lemma 5.1. The third constraint is guaranteed from the assumption of a trapezoidal fundamental diagram. The simplified problem is now:

$$\begin{aligned} \min_{f_l \mid l \in \mathcal{L}, f_r^{\mathrm{c}} \mid r \in \mathcal{R}} \quad & \textstyle\sum_{l \in \mathcal{L}} a_l f_l \\ \text{subject to:} \quad & \\ & 0 \leq f_l \leq f_l^{\max} \quad \forall l \in \mathcal{L} \\ & f_l = \textstyle\sum_{r \mid l \in r} f_r^{\mathrm{c}} + \bar{f}_l^{\mathrm{nc}} \quad \forall l \in \mathcal{L} \\ & \textstyle\sum_{r \mid o,d \in r} f_r^{\mathrm{c}} = \bar{f}_{o,d} \quad \forall o \in \mathcal{O}, d \in \mathcal{D} \end{aligned} \tag{5.14}$$

The above problem is now in a linear program formulation. We now show that the concept of noncooperative flow can be replaced by a capacity reduction on all the links. Let us rework some of the expressions in terms of the cooperative and noncooperative vehicles:

$$\begin{aligned} \min_{f_r^{\mathrm{c}} \mid r \in \mathcal{R}} \quad & \textstyle\sum_{l \in \mathcal{L}} a_l \bar{f}_l^{\mathrm{nc}} + \sum_{l \in \mathcal{L}} a_l \sum_{r \mid l \in r} f_r^{\mathrm{c}} \\ \text{subject to:} \quad & \\ & -\bar{f}_l^{\mathrm{nc}} \leq 0 \leq \textstyle\sum_{r \mid l \in r} f_r^{\mathrm{c}} \leq f_l^{\max} - \bar{f}_l^{\mathrm{nc}} \quad \forall l \in \mathcal{L} \\ & \textstyle\sum_{r \mid o,d \in r} f_r^{\mathrm{c}} = \bar{f}_{o,d} \quad \forall o \in \mathcal{O}, d \in \mathcal{D} \end{aligned} \tag{5.15}$$

We can now simplify further. The first term in the objective is constant, since f_r^{nc} is not a decision variable. Then, the second constraint can be simplified by introducing a reduced capacity constant, $\bar{f}_l^{\max} = f_l^{\max} - \bar{f}_l^{\mathrm{nc}}$. If we drop the *cooperative* pretense from the decision variable, then we have reduced the problem to a modified capacity, constant latency, Lagrangian system optimal problem, which is simplified and linear:

$$\begin{aligned} \min_{f_r \mid r \in \mathcal{R}} \quad & \textstyle\sum_{l \in \mathcal{L}} a_l \sum_{r \mid l \in r} f_r \\ \text{subject to:} \quad & \\ & 0 \leq \textstyle\sum_{r \mid l \in r} f_r \leq \bar{f}_l^{\max} \quad \forall r \in \mathcal{R} \\ & \textstyle\sum_{r \mid o,d \in r} f_r = \bar{f}_{o,d} \quad \forall o \in \mathcal{O}, d \in \mathcal{D} \end{aligned} \tag{5.16}$$

Lemma 5.2. *Let* $\mathbf{f}^* = \{f_r^* : r \in \mathcal{R}\}$ *be a solution to Problem* (5.16)*. Then*

$$\begin{aligned} \mathbf{f}^{\mathrm{c}\prime} &= \mathbf{f}^* \\ \rho_l' &= \frac{\bar{f}_l^{\mathrm{nc}} + \sum_{r \mid l \in r} f_r^*}{v_l} \end{aligned}$$

is a solution to Problem (5.12)*.*

Proof. Using Lemma 5.1, the equality $f_l = \bar{f}_l^{\mathrm{nc}} + \sum_{r|l\in r} f_r^{\mathrm{c}}$, and the variable name substitution made in Problem 5.16, the result follows immediately. □

Corollary 5.1. *An optimal solution to Problem* (5.12) *is a feasible solution of the unrelaxed junction model in Section 5.4.2.*

Proof. Since the flow on every link is in free flow ($f_l = v_l\rho_l, \forall l \in \mathcal{L}$), the supply $\sum_{l\in\Gamma_j} v_l\rho_l$ at every junction $j \in \mathcal{J}$ is equal to the flow across the junction $\sum_{l\in\Gamma_j} f_l$, and is therefore maximal. □

This corollary shows that solving for the static social optimum on networks with horizontal queues does not encounter the non-convexity issues typically associated with the CTM constraints in dynamic traffic problems. For instance, [130] uses the relaxed junction model (which allows "car-holding") that we present in Section 5.4.2 to solve the single destination social optimum problem as a linear program, and [50] use a relaxed junction model to solve an optimal ramp metering problem as a linear program (with a zero-relaxation gap under certain conditions).

Commonly considered problems in traffic assignment such as social optimum for purely Lagrangian flow ($\bar{f}_l^{\mathrm{nc}} = 0, \forall l \in \mathcal{L}$) and purely Eulerian flow ($f_r^{\mathrm{c}} = 0, \forall r \in \mathcal{R}$) serve as special cases of Corollary 5.1 and therefore an optimal solution can be found for both problems for the unrelaxed junction model by solving the linear program in Problem (5.16).

Limiting deviations in density

There are limitations in the expressiveness of the current horizontal queues model under total latency minimization. To circumvent these issues, this section proposes the addition of constraints that restrict the allowable densities to be within the locality of the nominal densities that are used to compute nominal latencies.

The purpose of these constraints is to prevent the optimization program from setting all links to be in the free-flow state, which has the negative effect of over-simplifying the model developed here-within (Section (5.4.2)). Instead, total latencies across the network can be minimized *while considering likely congestion patterns.* To motivate the usefulness of such a model, one can make a physical argument that rerouting may only cause bounded deviations in the density, and that congestion may not be cleared due to rerouting because of additional issues such as weaving or the physical road conditions.

To restrict the densities to only take certain values, we require that each link $l \in \mathcal{L}$, includes an upper and lower density bound, $\rho_l^{\uparrow}$ and $\rho_l^{\downarrow}$ respectively. We append to the program in Equation (5.12), the set of constraints bounding the allowable densities:

$$\rho_l^{\downarrow} \leq \rho_l \leq \rho_l^{\uparrow} \quad \forall l \in \mathcal{L}$$

In Section 5.5, we show an example of a network with horizontal queues with density bounds that has the bounded tolerance constraint as a tight constraint. This demonstrates that bounding the allowable densities can capture the characteristics of bounded tolerance for networks with horizontal queues.

5.4.3 Algorithm for data preconditioning

If input data into our problem is taken from a physical network with inherent sources of noise, it is likely that there will be a number of conditions that will cause the raw data to be incompatible with the problem constraints, thus making the problem infeasible. For instance, a link's estimated density may not lie within the fundamental diagram constraints in Section 5.4.2, or there may not be exact mass balance across junctions. If the estimates from the stationary sensors are reasonable, then these constraint violations will not be severe, but even small deviations will render the optimization program infeasible. Therefore the input data must be filtered to be preconditioned to meet the requirements. We propose an optimization program formulation.

The constraints that concern noncooperative flow are the following:

$$
\begin{aligned}
f_l &\leq v_l \rho_l & \forall l \in \mathcal{L} \qquad (5.17)\\
f_l &\leq w_l(\rho_l^{\max} - \rho_l) & \forall l \in \mathcal{L} \qquad (5.18)\\
0 \leq f_l &\leq f_l^{\max} & \forall l \in \mathcal{L} \qquad (5.19)\\
f_l &= \textstyle\sum_{r|l\in r} f_r & \forall l \in \mathcal{L} \qquad (5.20)
\end{aligned}
$$

These constraints are all convex (indeed, linear) in the decision variables f_l, f_r. Then, let $\hat{f}_l, \hat{\rho}_l$ be the input flow and density respectively on link $l \in \hat{\mathcal{L}}$, where $\hat{\mathcal{L}} \subseteq \mathcal{L}$ are the links with input data available. Our objective will be to minimize some definition of distance from the input data to the selected data that violates none of the above constraints. If we select as the distance measurement the *n-norm*, $n \geq 1$, then we have the following convex optimization program for obtaining amenable input data:

$$
\begin{aligned}
\underset{f_l,\rho_l : l\in\mathcal{L}}{\text{minimize}} \quad & \textstyle\sum_{l\in\bar{\mathcal{L}}} \left\| \hat{f}_l - f_l \right\|_n + \|\hat{\rho}_l - \rho_l\|_n \\
\text{subject to:} \quad & \text{Constraints } (5.17) - (5.20)
\end{aligned}
$$

The result of the optimization problem is a set of route-based flows $\{f_r : r \in \mathcal{R}\}$. Finally, the route flows would then be partitioned into both cooperative and noncooperative flows, which then gets the data in a suitable format.

While the formulation presented specifically discusses horizontal queue constraints, the same methods can be extended to other problems with convex constraints, such as M/M/1 queues presented in Section 5.4.1.

Name	$a\ \left(\frac{s^2}{\text{units}}\right)$	$b\ (s)$	$f^{\max}\ \left(\frac{\text{units}}{s}\right)$
source	1	0	1
sink	1	0	1
left	1	0	1
right	0.5	0.5	1

(a)

Type	Description	$f\ \left(\frac{\text{cars}}{\text{s}}\right)$
O-D (Lagrangian)	source-sink	0.8
Link (Eulerian)	source	0.2
Link (Eulerian)	sink	0.1
Link (Eulerian)	left	0.1
Link (Eulerian)	right	0.2

(b)

Table 5.2: Summary of illustrative network properties. **5.2a**: Link-level input parameters. **5.2b:** Network-level input demands

5.5 Numerical results

We demonstrate the highly practical nature of our work by applying the model to two different problems. We focus on a multi-destination network of horizontal queues. This problem demonstrates the generality of our method to the multi-commodity case and the ability to solve real-world transportation planning problems on a regional level. First, we demonstrate the simplicity of the model on a small network of vertical queues, to which we apply both models of tolerance and compare the benefits gained by re-routing.

5.5.1 Linear Latency

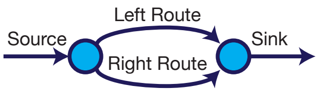


Figure 5.4: Network diagram

Figure 5.4 depicts an illustrative, two parallel routes network. Flow enters at the source and exits at the sink and can travel along either the "left" route or the "right" route. Each link $l \in \mathcal{L}$ has a linear latency function $\ell_l(f) = a_l f + b_l$. The link properties given in Figure 5.2a show that the left link has a lower *zero-flow* latency than the right link, but has a higher marginal cost per unit flow. As the left route becomes more congested, its latency will eventually increase until the right route has equal utility. From a user equilibrium viewpoint, as the network is loaded with additional flow, the latencies across the two routes will remain the same. But from a social optimum viewpoint, additional flow will always be routed to the right route since it will always have a lower marginal cost than the left route.

As described in Figure 5.2b, we assume the network is loaded with both cooperative and noncooperative flow. There is a total of 0.2 units-per-second of noncooperative flow, with 0.1 units-per-second of flow on both the left and right link. In addition, there is 0.8 units-per-second of flow on the network, which we assume is initially distributed amongst the left and right routes in a manner that achieves user equilibrium.

We now show how our route optimization framework can be applied to this network to optimally route the cooperative flow. Results are given over a range of bounded rationality

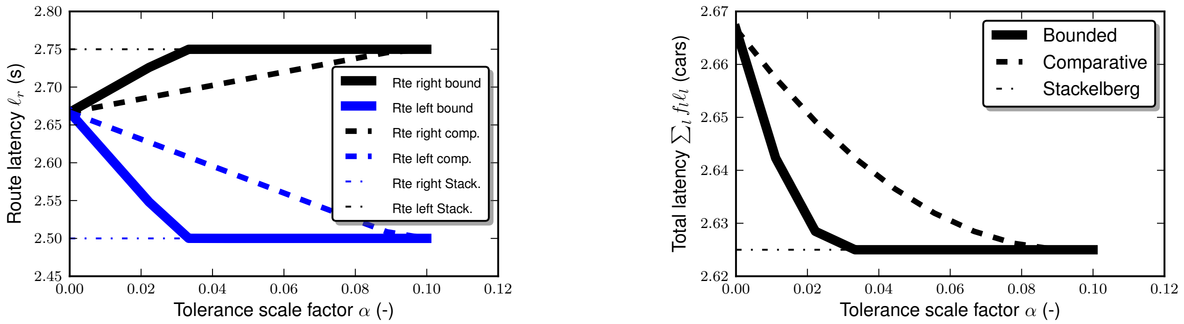


Figure 5.5: Comparison of simple bounded tolerance and comparative tolerance. **5.5a:** Route latencies. Comparative tolerance allows smaller deviations in route latencies than bounded tolerance. **5.5b:** Total latencies. The total latencies decrease more slowly with the comparative tolerance model versus the bounded tolerance model. The total route flows approach the Stackelberg equilibrium as the tolerance scale factor goes to infinite.

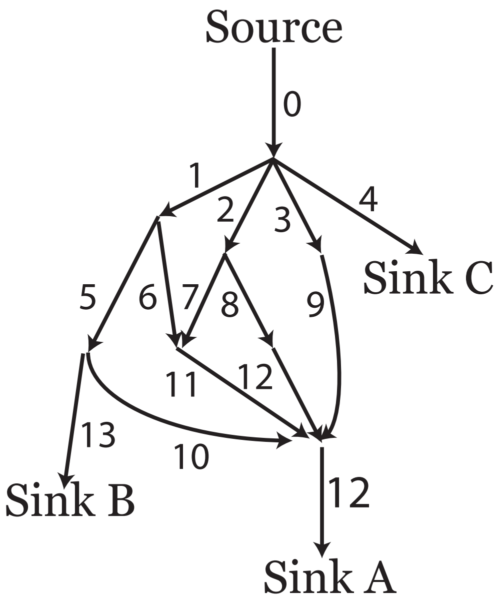
scale factor to show the sensitivity of our results to the scale factor. Since the latency functions are linear, both the standard bounded rationality model and comparative bounded rationality model are solved using well-developed and highly efficient convex optimization tools (see [112]). In addition to comparing the two models against each other, we compare them both against the Stackelberg game solution of the routing problem. The Stackelberg game solution gives a minimum social cost with the assumption that the noncooperative flow will be routed in a user equilibrium manner. Stackelberg analysis is only possible in the case when origin-destination demands can be uniquely determined for all users of the network, which holds for our simple network. This does not hold in general, and this case will studied subsequently (Section 5.5.2).

Figure 5.5 summarizes the numerical results on the simple network. The route latencies as a function of the bounded rationality scale factor are shown in Figure 5.5a, while total latencies are shown in Figure 5.5b. As expected, as the bounded rationality scale factor increases, so do the benefits of re-routing. Additionally, the comparative bounded rationality model improves at a slower rate than the standard bounded rationality model. This is due to the fact that the comparative model permits the right route to be "aware" of the latency improvements on the left route, while the standard model only limits deviations in route latencies in comparison to a route's individual *nominal* latency and ignores the improvements on the left route.

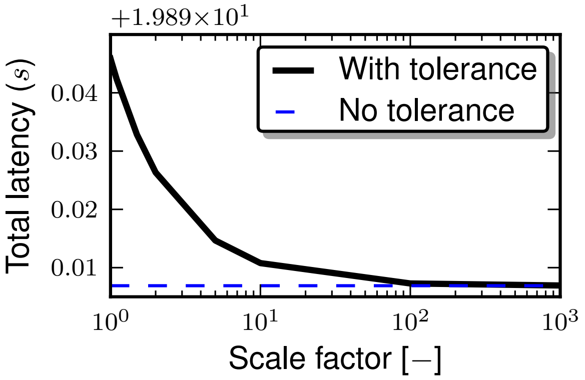
The results tell us that as the scale factor increases, the model converges to the Stackelberg solution. It may appear counter-intuitive that the model with inherently no tolerance factor could perform better than the tolerance models. The explanation is that the tolerance models are overly-conservative due to the assumption that no noncooperative flow changes routes, and the routing strategy will not drastically improve one route over another route.

On the other hand, the Stackelberg solution shifts all noncooperative flow to the left route and the cooperative flow accommodates this shift in a socially optimal manner. Since all noncooperative flow is on a single route and improves upon its nominal latency, discrepancies in route latencies are no longer a behavioral issue, allowing the Stackelberg to be as liberal as necessary with latency increases on the right route.

5.5.2 Horizontal queueing network



(a) Multiple-destination network with horizontal queues. There are many overlapping routes between *Source* and *Sink A*, while *Sink B* and *Sink C* are origin-destinations which have demands on the same network as *Sink A* demands.



(b) Total latency on network of horizontal queues as a function of tolerance scale.

As discussed in Section 5.4.2, given the nonlinear dynamics of horizontal queueing cells, modeling horizontal queues is in general a more difficult process than vertical queues. It is also important to consider a more general network than the compact one in Section 5.5.1, one with multiple destinations, and therefore multiple Lagrangian demand types. In this section, we model a mid-sized multi-destination network of horizontal queues within the partial cooperation, bounded tolerance framework. We follow with numerical results on how the routing strategies change with respect to the parameters of the tolerance model.

Name	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Length (m)	0.5	0.1	1.0	1.5	0.3	0.1	0.6	0.4	1.0	1.5	1.0	0.7	0.8	1.0	0.1
Nom. flow ($\frac{\text{units}}{s}$)	0.15	0.05	0.03	0.02	0.05	0.05	0	0.02	0.01	0.02	0	0.02	0.01	0.05	0.05
Nom. density ($\frac{\text{units}}{m}$)	0.15	0.05	0.03	4.99	0.05	0.05	0	0.02	4.99	4.99	0	0.02	0.01	0.05	0.05
State constraint	F	F	F	C	F	F	F	F	C	C	F	F	F	F	F

(a)

Sink Dest.	A	B	C
Flow ($\frac{\text{users}}{s}$)	0.5	0.5	0.5

(b)

Table 5.3: Multi-destination network properties. **5.3a:** Link properties, including nominal state. **5.3b:** demand input into network

Network properties and demands

Figure 5.6a shows a topological description of the network. Since only *Sink A* can be reached through multiple routes, the algorithm only decides how to partition the demand across the routes originating from *Source* and leading to *Sink A*. The algorithm will take as input some network and link level properties, as recorded in Table 5.3. The nominal state given in Table 5.3a shows that links 3, 8 and 9 were heavily congested, while the other links were close to free flow. Furthermore, as discussed in Section 5.4.2, the densities must have more constraints than just the fundamental diagram constraints (Equations (5.9)-(5.11)). Table 5.3a tells us that this problem assumes that links do not shift from their nominal state (links with a nominal free flow/congestion state must maintain this state). Table 5.3b tells us that there is 0.5 $\frac{\text{users}}{sec}$ demand between all origin-destination pairs on the network. For simplicity, we assume that all demand is cooperative as well to focus analysis.

Numerical results

The results of our numerical calculations are summarized in Figure 5.6b. As supported by the results for vertical queues in Section 5.5.1, the relief of network congestion is greater the more tolerance is assumed in the users. Additionally, it is noted that the network does not immediately push into free flow (social optimum), but rather decongests links to an amount dependent on the tolerance scale factor. This is a desirable behavior of the model, as it is not reasonable to assume that congestion can be completely avoided just through re-routing schemes. Lastly, we see the intuitive result that the bounded tolerance model will converge to the more familiar social optimum as the scale factor increases.

5.6 Summary of Results

We have presented a framework rerouting flow in an socially optimal way with mixed Lagrangian-Eulerian information. The cooperative flow has known nominal routes, while the noncooperative flow has known flow counts across links. In order to anticipate network conditions for all users after re-routing has been applied, the model combines the two types of information in a complementary way; by only allowing the cooperative flow to change routes, we have removed the necessity of having origin-destination demand information for all users on the network. Furthermore, by looking at the static flow problem, we can study practical networks with multiple origins and multiple destinations, where dynamic multi-commodity problems often suffer from intractability issues.

The framework also addresses the behavioral nature of the noncooperative users, which we call *bounded tolerance* and *comparative tolerance*, by only allowing perturbations of the nominal state of the network that boundedly impact the noncooperative flow in a negative manner. The tolerance model comes about as a response to the lack of origin-destination information that does not permit the game-theoretic Stackelberg game analysis, but does allow us to require only Eulerian information across the majority of the network. We show that the comparative tolerance model will in general limit network latency improvements more so than the bounded tolerance model, but since the comparative tolerance model allows individual routes to compare latencies with other routes, it is arguably a more accurate model of noncooperative flow behavior.

By taking a convex optimization-based approach, the framework is shown to efficiently solve many classes of network flow problems. The horizontal queue, highway network problem can be modeled as a convex optimization program, which permits one to study highway networks of practical size. The multi-destination network of horizontal queues gives an example of how the framework can be applied to multi-commodity type networks such as highways with multiple onramps and off-ramps. A live data feed of Lagrangian GPS sensors and Eulerian loop detectors, in conjunction with the data pre-conditioning algorithm, would enable the framework to run in an "online" sense, and provide automatic, daily routing advice for a traffic management agency during rush hour periods. We conclude that the partial cooperation, bounded tolerance model can allow a traffic management operator to make beneficial re-routing decisions with much less origin-destination demand input required.

Chapter 6

Conclusion, Future Work, and Vision

To conclude the work presented in this thesis, we summarize the main contributions, detail a number of extensions for future study, and provide a broader context and purpose in which the work was developed.

Summary of Contributions

- **A continuous and discrete model for freeway onramp metering and optimal control applications.** In Chapter 2, we covered preliminary networked conservation law theory including the PDE formulation, Riemann solvers and Godunov discretization. From these tools, we derived a new model for linear freeway stretches, where the mainline flow is modeled as networked horizontal queues of vehicle density, while the onramps are modeled as ODE's of vertical queues of vehicle counts. The ODE allows for strong boundary conditions to be guaranteed at the onramp boundaries, and thus guaranteeing all applied boundary flow enters the system. We discussed the suitability of such a model to optimal control applications, where strong boundary conditions are desirable.

- **Adjoint-based finite horizon optimal control framework for networked conservation laws.** In Chapters 3.1 and 3.2, we gave an overview of the discrete adjoint method and its general applicability to optimal control problems. We derived a specific discrete adjoint formulation for networked conservation laws generalized about the specific Riemann solver being used in the application. Such a formulation allows for a study of the sparsity structure of such networked systems. We showed that such a sparsity structure permits one to compute gradients of optimal control objectives on such systems with complexity linear in the size of the network and linear in the time horizon. We demonstrated the effectiveness and robustness of an adjoint-based MPC ramp metering controller on a model of the I15 South freeway, where our method was able to improve upon standard practicioners' techniques.

- **Distributed optimization over subsystems with shared state.** In Chapter 3.3, we derived a decentralized and asynchronous control algorithm over sub-systems which share not only "free" control variables, but "dependent" state variables. We first presented a general treatment of the problem and solution method following the A-ADMM algorithm [127]. Then we showed how networked conservation laws, such as freeway networks, fit the assumptions of the presented problem, and how its specific sparsity pattern permits one to solve optimal control problems in parallel by splitting the network into subnetworks, with communication requirements that scale linearly with the network size. Furthermore, we showed how the adjoint method can be applied to the subnetwork subproblems for systems with nonconvex dynamics. We then implemented a provably optimal, decentralized ramp metering and variable-speed-limit controller and demonstrated its improved running time with increasing subnetwork splits and its performance improvement over simpler decentralized approaches.

- **Security analysis of freeway control systems.** In Chapter 4, we investigated the security and potential compromise points of freeway control systems, including by the physical and virtual aspects of control, sensing and communication. We distinguished between direct attacks, where the actuation is directly compromised, and indirect attacks, where sensing infrastructure is compromised in a manner which induces a desired outcome from the actuation. For coordinated ramp metering attacks, we constructed a high-level framework, based on optimal control and multi-objective optimization, which enables an attacker to accomplish precise and intricate objectives using only metering lights as the control. A number of attack simulations were conducted on macroscopic freeway models which demonstrate the level of precisions possible from a coordinated ramp metering attack.

- **Rerouting strategies using mixed Lagrangian-Eulerian information.** In Chapter 5, we presented a framework for static route suggestions to a subset of users on flow networks occupied by non-compliant, greedy users. The framework only requires route-based, Lagrangian information from the compliant users, and flow counts on links from all users (Eulerian information). After applying a *bounded-tolerance* model for the non-compliant drivers, we posed the optimal route suggestion problem as a convex optimization problem and give numerical examples applying the framework to both communication network dynamics and freeway dynamics.

Future Work During the course of conducting the above work, a number of avenues for further research were identified.

- **Adjoint-based model calibration.** While using the discrete adjoint framework to conduct *congestion-on-demand* attacks in Chapter 4, it was identified that one could consider *congestion-on-demand* objectives as model calibration, using onramp flow as the tuning model parameter. If one were to use more standard model parameters, such as the triangular fundamental diagram parameters and split ratios as the controllable, tuning parameters, then one can employ the discrete adjoint framework to minimize the model prediction error from some known sensor measurements by optimally adjusting the fundamental diagram parameters. A similar concept was presented in [60] for state estimation. As future work, one could study how such a calibration technique could be introduced into an MPC framework to allow for automatic, dynamic adjustment of freeway model parameters to account for unknowns such as weather or lane closures.

- **Sensitivity analysis of coordinated traffic subnetworks.** The strength of the A-ADMM approach to distributed freeway control presented in Chapter 3.3 comes from the transmission of not only boundary conditions to neighboring subnetworks, but objective value information via the Lagrangian dual variables. These dual variables also capture the *sensitivity* of the objective to a particular dynamical constraint violation. Since constraint violation is equivalent to consensus enforcement in A-ADMM, by studying the dual variable values being transmitted between different subnetworks, one observes the impact of their communication on the objective value. Such analyses could inform traffic control systems designers on which subnetworks one should invest in enabling coordination, and which subnetworks do not require such investments.

Vision Adapting to the broader trends of smarter cities and connected infrastructure, transportation agencies have become increasingly better at measuring, estimating and predicting their traffic patterns. We view understanding and measuring as the first step in a fully closed-loop management system, where control systems can leverage the improved estimation and prediction systems. The research contained in this thesis works towards a practical and effective implementation of the control and actuation part of such a future management system, and has done so from two distinct perspectives.

First, we have given numerous real-world applications and numerical examples to demonstrate the current-day feasibility of the proposed methods. The PDE-ODE model in Chapter 2.2.2 was developed since strong boundary conditions were necessary for vehicle demand conservation, thus enabling accurate modeling of the system response at the boundaries to varying control. Robustness validations were conducted in Chapter 3 to address the realistic scenario of noise pervading many dimensions of the control problems (initial conditions, boundary conditions, model parameters). In Chapter 4, controllability analysis was conducted on a realistic model of the I15 Freeway to demonstrate the vulnerability of real-world traffic systems to harmful attacks with ramp meters as the only control mechanism.

Second, the presented control framework was specifically designed for generalization to enable extensions to future control methods and new applications. The adjoint-based *model predictive control* framework in Chapter 1.3 extends readily to any continuous system which can be discretized using Godunov's method, while still having the same computational efficiency guarantees. Thus, if one prefers a more expressive offramp model for freeway-onramp traffic, or one wants to consider variable speed limits, or even a cost function which considers fairness, little specialization is needed to apply optimal control to the new model besides the straight-forward calculation of partial derivatives. The decentralized control theory in Chapter 3.3.3 can be seen as a “module” for permitting looser communication and higher scalability applied to the centralized adjoint framework, without any sacrifice in the expressiveness or extensibility of the centralized approach.

Thus, we view the current work as sitting squarely between immediate application and future extension. The specific examples presented for ramp metering, variable speed limits, and even rerouting in Chapter 5 and [113] serve as a “proof-of-concept”, and are readily implementable within any control system with some level of coordination and communication. But additionally, and potentially more-importantly, these examples sit upon a general framework which is agnostic to the specific networks, the current objectives, and even the phenomena being modeled. In light of the evidence of feasibility and versatility presented in this thesis, it is our hope that model-predictive control in traffic applications will continue to be studied and adopted in practice.

Bibliography

[1] Connected Corridors, http://connected-corridors.berkeley.edu/, 2013. 43

[2] UC Berkeley, Partners for Advanced Transportation Technology: Connected Corridors, 2014. 1

[3] AASHTO, ITE, and NEMA. Model 2070 Controller Standard Version 03. Technical report, 2012. 63

[4] Anil Aswani and Claire J Tomlin. Game-theoretic routing of GPS-assisted vehicles for energy efficiency. In *American Control Conference (ACC), 2011*, pages 3375–3380. IEEE, 2011. 93, 96

[5] Mapundi K Banda and Michael Herty. Adjoint imex-based schemes for control problems governed by hyperbolic conservation laws. *Computational Optimization and Applications*, 51(2):909–930, 2012. 25, 30

[6] Jaume Barceló, Esteve Codina, J Casas, J L Ferrer, and D Garcia. Microscopic traffic simulation: A tool for the design, analysis and evaluation of intelligent transport systems. *Journal of Intelligent and Robotic Systems*, 41(2-3):173–203, 2005. 61, 66, 74

[7] Alexandre M. Bayen, Robin L Raffard, and Claire J Tomlin. Adjoint-based control of a new eulerian network model of air traffic flow. *IEEE Transactions on Control Systems Technology*, 14(5):804–818, September 2006. 24, 44, 45

[8] Moshe Ben-Akiva, David Cuneo, Masroor Hasan, Mithilesh Jha, and Qi Yang. Evaluation of freeway control using a microscopic simulation laboratory. *Transportation Research Part C: Emerging Technologies*, 11(1):29–50, 2003. 26

[9] Louis Blanchard, Régis Duvigneau, Anh-Vu Vuong, and Bernd Simeon. Shape Gradient for Isogeometric Structural Design. *Journal of Optimization Theory and Applications*, pages 1–7, September 2013. 24

[10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010. 4, 44, 49

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, volume 25. Cambridge University Press, 2010. 34

[12] Alberto Bressan. *Hyperbolic systems of conservation laws*, volume 20 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2000. 8

[13] Alberto Bressan and Graziano Guerra. Shift-differentiability of the flow generated by

a conservation law. *Discrete Contin. Dynam. Systems*, 3(1):35–58, 1997. 24

[14] Smita Brunnermeier and Sheila A Martin. Interoperability cost analysis of the US automotive supply chain: Final report. Technical report, DIANE Publishing, 1999. 24

[15] Eduardo Camponogara and Lucas Barcelos De Oliveira. Distributed optimization for model predictive control of linear-dynamic networks. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(6):1331–1338, 2009. 4, 44, 45

[16] W. Castaings, D. Dartus, M. Honnorat, F.L. Dimet, Y. Loukili, and D. Monnier. Automatic differentiation: a tool for variational data assimilation and adjoint sensitivity analysis for flood modeling. In *Automatic Differentiation: Applications, Theory, and Implementations*, volume 50, pages 249–262. Springer, 2006. 25

[17] Rohan Chabukswar, Bruno Sinópoli, Gabor Karsai, Annarita Giani, Himanshu Neema, and Andrew Davis. Simulation of network attacks on SCADA systems. In *First Workshop on Secure Control Systems*, 2010. 66

[18] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. Freeway performance measurement system: mining loop detector data. *Transportation Research Record: Journal of the Transportation Research Board*, 1748(1):96–102, 2001. 39

[19] OJ Chen, AF Hotz, and ME Ben-Akiva. Development and evaluation of a dynamic ramp metering control model. Technical report, 1997. 26

[20] G.M. Coclite and Benedetto Piccoli. Traffic Flow on a Road Network. *Arxiv preprint math/0202146*, 2002. 98

[21] Codenomicon. The Heartbleed Bug, 2014. 60

[22] Jean-Michel Coron, Rafael Vazquez, Miroslav Krstic, and Georges Bastin. Local Exponential H 2 Stabilization of a 2 X 2 Quasilinear Hyperbolic System Using Backstepping. *SIAM Journal on Control and Optimization*, 51(3):2005–2035, 2013. 24

[23] C. F. Daganzo. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, 28(4):269–287, 1994. 92, 97

[24] C. F. Daganzo. The cell transmission model, part ii: network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995. 26, 45, 50, 54, 97, 98

[25] Ciro D'Apice, Simone Gottlich, Michael Herty, and Benedetto Piccoli. *Modeling, Simulation, and Optimization of Supply Chains: A Continuous Approach.* SIAM, 2010. 4, 25, 34

[26] Shideh Dashti, Jonathan D Bray, Jack Reilly, Steven Glaser, Alexandre Bayen, and Ervasti Mari. Evaluating the Reliability of Phones as Seismic Monitoring Instruments. *Earthquake Spectra*, 30(2):721–742, 2014. 3

[27] Maria Laura Delle Monache, Jack Reilly, Samitha Samaranayake, Walid Krichene, Paola Goatin, and Alexandre M. Bayen. A PDE-ODE model for a junction with ramp buffer. *SIAM Journal on Applied Mathematics*, 74(1):22–39, 2014. 4, 8, 19, 22, 45, 54

[28] Gunes Dervisoglu, Alexander Kurzhanskiy, Gabriel Gomes, and Roberto Horowitz. Macroscopic freeway model calibration with partially observed data, a case study. In

American Control Conference (ACC), 2014, pages 3096–3103. IEEE, 2014. 2, 3, 72

[29] David L Donoho, Arian Maleki, Inam Ur Rahman, Morteza Shahram, and Victoria Stodden. Reproducible research in computational harmonic analysis. *Computing in Science & Engineering*, 11(1):8–18, 2009. 72

[30] AC Duffy. An Introduction to Gradient Computation by the Discrete Adjoint Method. 2009. 4

[31] Stacy M Eisenman, Xiang Fei, Xuesong Zhou, and Hani S Mahmassani. Number and location of sensors for real-time network traffic estimation and prediction: Sensitivity analysis. *Transportation Research Record: Journal of the Transportation Research Board*, 1964(1):253–259, 2006. 2

[32] M Ervasti, S Dashti, J Reilly, J Bray, S Glaser, and Bay. iShake: Mobile Phones as Seismic Sensors ? User Study Findings. 3

[33] LC C Evans. *Partial differential equations*. Graduate studies in mathematics. American Mathematical Society, 1998. 8

[34] FHWA. Type 170 Traffic Signal Controller System - Microcomputer Based Intersection Controller. Technical report, Federal Highway Administration, 1978. 63

[35] Anthony V Fiacco and Garth P McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*, volume 4. Siam, 1990. 34

[36] Kathrin Flasskamp, Todd Murphey, Sina Ober-Blobaum, and Sina Ober-bl. Switching time optimization in discretized hybrid dynamical systems. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, number 4, pages 707–712. IEEE, 2012. 29

[37] Emilio Frazzoli, Munther A Dahleh, and Eric Feron. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, 25(1):116–129, 2002. 24

[38] J R D Frejo and E F Camacho. Feasible Cooperation Based Model Predictive Control for Freeway Traffic Systems. *Conference on Decision and Control*, 50(2):5965–5970, 2011. 3, 4, 26, 37, 44, 50, 52, 54, 57

[39] José Ramón Domínguez Frejo and Eduardo Fernández Camacho. Global Versus Local MPC Algorithms in Freeway Traffic Control With Ramp Metering and Variable Speed Limits. *Intelligent Transportation Systems, IEEE Transactions on*, 13(4):1556–1565, 2013. 26, 37

[40] Armin Fugenschuh, Michael Herty, Axel Klar, and Alexander Martin. Combinatorial and continuous models for the optimization of traffic flows on networks. *SIAM Journal on Optimization*, 16(4):1155–1176, 2006. 25

[41] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976. 4, 44

[42] M. Garavello and B. Piccoli. *Traffic flow on networks*, volume 1. American institute of mathematical sciences Springfield, MA, USA, 2006. 4, 8, 10, 17, 19, 24, 25, 54, 98

[43] Ralf Giering and Thomas Kaminski. Recipes for adjoint code construction. *ACM Transactions on Mathematical Software (TOMS)*, 24(4):437–474, 1998. 24

[44] Michael Giles and Niles Pierce. Adjoint equations in CFD : duality , boundary conditions and solution behaviour. *AIAA paper*, 97(1850):182–198, 1997. 25

[45] Michael B. Giles and Niles A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65(3-4):393–415, 2000. 4, 24, 26

[46] Mike Giles and Stefan Ulbrich. Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. part 2: Adjoint approximations and extensions. *SIAM Journal on Numerical Analysis*, 48(3):905–921, 2010. 25

[47] Pontus Giselsson, Minh Dang Doan, Tams Keviczky, Bart De Schutter, and Anders Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49(3):829–833, 2013. 44, 45

[48] O. Glass and S. Guerrero. On the uniform controllability of the Burgers equation. *SIAM Journal on Control and Optimization*, 46(4):1211–1238, January 2007. 24

[49] Sergei Konstantinovich Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959. 4, 11, 26

[50] G. Gomes and R. Horowitz. Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C: Emerging Technologies*, 14(4):244–262, 2006. ix, 4, 7, 25, 26, 97, 101

[51] G. Gomes, R. Horowitz, A. A. Kurzhanskiy, P. Varaiya, and J. Kwon. Behavior of the cell transmission model and effectiveness of ramp metering. *Transportation Research Part C: Emerging Technologies*, 16(4):485–513, August 2008. 83, 97

[52] Shelby Grad. Engineers who hacked into L.A. traffic signal computer, jamming streets, sentenced. *Los Angeles Times*, 2009. 60

[53] Martin Gugat. Contamination source determination in water distribution networks. *SIAM Journal on Applied Mathematics*, 72(6):1772–1791, 2012. 24

[54] Martin Gugat, Markus Dick, and Gnter Leugering. Gas flow in fan-shaped networks: classical solutions and feedback stabilization. *SIAM Journal on Control and Optimization*, 49(5):2101–2117, 2011. 24, 50

[55] Martin Gugat, Michael Herty, Axel Klar, and Günter Leugering. Optimal Control for Traffic Flow Networks. *Journal of Optimization Theory and Applications*, 126(3):589–616, September 2005. 4, 24, 25, 29, 30

[56] X. Guo and H.X. Liu. Bounded rationality and irreversible network change. *Transportation Research Part B: Methodological*, 45(10):1606–1618, December 2011. 87, 88, 94

[57] T. Hu and H.S. Mahmassani. Day-to-day evolution of network flows under real-time information and reactive signal control. *Transportation Research Part C: Emerging Technologies*, 5(1):51–69, 1997. 87, 88, 94

[58] Timothy Hunter, Aude Hofleitner, Jack Reilly, Walid Krichene, Jerome Thai, Anastasios Kouvelas, Pieter Abbeel, and Alexandre Bayen. Arriving on time : estimating travel time distributions on large-scale road networks. In *In preparation*, 2013. 2

[59] D Jacquet, CC de Wit, and D Koenig. Freeway traffic control and monitoring with distributed macroscopic models. *gipsa-lab.grenoble-inp.fr*. 2, 25

[60] Denis Jacquet, Carlos Canudas de Wit, and Damien Koenig. Optimal Ramp Metering Strategy with Extended LWR Model, Analysis and Computational Methods. In *Proceedings of the 16th IFAC World Congress*, 2005. 24, 110

[61] Denis Jacquet, Miroslav Krstic, and C.C. Canudas de Wit. Optimal control of scalar one-dimensional conservation laws. In *American Control Conference, 2006*, number 2, pages 6—-pp. IEEE, Ieee, 2006. 24

[62] Antony Jameson and Luigi Martinelli. *Aerodynamic shape optimization techniques based on control theory.* Springer, 2000. 24

[63] Yasser Jebbari, Walid Krichene, Jack D Reilly, and Alexandre M Bayen. Stackelberg thresholds on parallel networks with horizontal queues. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 268–274. IEEE, 2013. 7

[64] Tobias Jeske. Floating car data from smartphones: What google and waze know about you and how hackers can control traffic. *Proc. of the BlackHat Europe*, 2013. 60

[65] Zhanfeng Jia, Chao Chen, Ben Coifman, and Pravin Varaiya. The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 536–541. IEEE, 2001. 60, 72

[66] P Kachroo. *Feedback ramp metering in intelligent transportation systems.* Springer, 2003. 26

[67] Diana Keller. Optimal Control of a Nonlinear Stochastic Schrodinger Equation. *Journal of Optimization Theory and Applications*, September 2013. 24

[68] F. Kelly. The mathematics of traffic in networks. *The Princeton companion to mathematics*, 2008. 5

[69] Apostolos Kotsialos and Markos Papageorgiou. Nonlinear Optimal Control Applied to Coordinated Ramp Metering. *IEEE Transactions on Control Systems Technology*, 12(6):920–933, November 2004. 4, 7, 24, 26

[70] Walid Krichene, Jack Reilly, Saurabh Amin, and Alexandre Bayen. On the characterization and computation of Nash equilibria on parallel networks with horizontal queues. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 7119–7125. IEEE, Ieee, December 2012. 7, 87, 93

[71] Walid Krichene, Jack Reilly, Saurabh Amin, and Alexandre M. Bayen. Stackelberg Routing on Parallel Networks With Horizontal Queues. *Automatic Control, IEEE Transactions on*, 59(3):714–727, March 2014. 3, 7, 93

[72] SN Kružkov. First order quasilinear equations in several independent variables. *Sbornik: Mathematics*, 10(2):217—-243, 1970. 8

[73] J P Lebacque. The Godunov scheme and what it means for first order traffic flow models. In *Internaional symposium on transportation and traffic theory*, pages 647–677, 1996. 4, 17

[74] R. Leblanc and J. L. Walker. Which is the biggest carrot? Comparing non-traditional incentives for demand management. 2012. 89

[75] M.J. J Lighthill and G.B. B Whitham. On kinematic waves. II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A.*

Mathematical and Physical Sciences, 229(1178):317, 1955. 4, 15, 26, 92, 97

[76] H. K. Lo and W. Y. Szeto. A cell-based variational inequality formulation of the dynamic user optimal assignment problem. *Transportation Research Part B: Methodological*, 36(5):421–443, 2002. 83, 92, 93

[77] D. Merugu, B.S. Prabhakar, and N.S. Rama. An incentive mechanism for decongesting the roads: A pilot program in bangalore. *NetEcon, ACM Workshop on the Economics of Networked Systems*, 2009. 86

[78] Mark Allan Miller and Alexander Skabardonis. *San Diego I-15 Integrated Corridor Management (ICM) System: Stage II (Analysis, Modeling, and Simulation.* California PATH Program, Institute of Transportation Studies, University of California at Berkeley, 2010. 1, 60, 74

[79] Parviz Moin and Thomas Bewley. Feedback Control of Turbulence. *Applied Mechanics Reviews*, 47(6S):S3, 1994. 24, 25

[80] Joo F C Mota, Joo M F Xavier, Pedro M Q Aguiar, and Markus Pschel. Distributed admm for model predictive control and congestion control. In *CDC*, pages 5110–5115, 2012. 4, 44, 45

[81] J-D Müller and P Cusdin. On the performance of discrete adjoint CFD codes using automatic differentiation. *International journal for numerical methods in fluids*, 47(8-9):939–945, 2005. 24

[82] Ajith Muralidharan, Gunes Dervisoglu, and Roberto Horowitz. Freeway traffic flow simulation using the Link Node Cell transmission model. *2009 American Control Conference*, pages 2916–2921, 2009. 2, 3

[83] Ajith Muralidharan and Roberto Horowitz. Imputation of Ramp Flow Data for Freeway Traffic Simulation. *Transportation Research Record: Journal of the Transportation Research Board*, 2099(-1):58–64, January 2009. 72

[84] Ajith Muralidharan and Roberto Horowitz. Optimal control of freeway networks based on the link node cell transmission model. In *American Control Conference (ACC)*, number c, pages 5769–5774. IEEE, 2012. ix, 3, 7, 25, 26, 45, 54, 55, 56, 63

[85] Haim Nessyahu and Eitan Tadmor. Non-oscillatory central differencing for hyperbolic conservation laws. *Journal of computational physics*, 87(2):408–463, 1990. 25

[86] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983. 44

[87] Yu Marco Nie and Yang Liu. Existence of self-financing and Pareto-improving congestion pricing: Impact of value of time distribution. *Transportation Research Part A: Policy and Practice*, 44:39–51, 2010. 92

[88] Brendan O'Donoghue, Giorgos Stathopoulos, and Stephen P Boyd. A splitting method for optimal control. *IEEE Trans. Contr. Sys. Techn.*, 21(6):2432–2442, 2013. 3, 44

[89] C. Papadimitriou. Algorithms, games, and the internet. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing - STOC '01*, pages 749–753, New York, New York, USA, July 2001. ACM Press. 84

[90] C. Papadimitriou. Selfish Routers and the Price of Anarchy. *Science*, 2010. 92, 93

[91] M. Papageorgiou. Dynamic modeling, assignment, and route guidance in traffic net-

works. *Transportation Research Part B: Methodological*, 24(6):471–495, December 1990. 83, 93

[92] M Papageorgiou, J M Blosseville, and H Hadj-Salem. Macroscopic modelling of traffic flow on the Boulevard P{é}riph{é}rique in Paris. *Transportation Research Part B: Methodological*, 23(1):29–47, 1989. 7

[93] M Papageorgiou, H Hadj-Salem, and J.M. Blosseville. ALINEA: A local feedback control law for on-ramp metering. *Transportation Research Record*, 1320:58–64, 1991. 4, 7, 26, 36, 37, 54, 60

[94] Markos Papageorgiou, Habib Hadj-Salem, and F Middelham. ALINEA local ramp metering: Summary of field results. *Transportation Research Record: Journal of the Transportation Research Board*, 1603(1):90–98, 1997. 7

[95] Ioannis Papamichail, Markos Papageorgiou, Vincent Vong, and John Gaffney. Heuristic ramp-metering coordination strategy implemented at Monash freeway, Australia. *Transportation Research Record: Journal of the Transportation Research Board*, 2178(1):10–20, 2010. 7, 26, 36

[96] O Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974. 26

[97] Ye Pu, Melanie N. Zeilinger, and Colin N. Jones. Fast alternating minimization algorithm for model predictive control. In *IFAC World Congress*, page (To appear), 2014. 44

[98] Tarek S. Rabbani, Florent Di Meglio, Xavier Litrico, and Alexandre M. Bayen. Feed-Forward Control of Open Channel Flow Using Differential Flatness. *IEEE Transactions on Control Systems Technology*, 18(1):213–221, January 2010. 24

[99] Robin L Raffard, Keith Amonlirdviman, Jeffrey D Axelrod, and Claire J Tomlin. An adjoint-based parameter identification algorithm applied to planar cell polarity signaling. *Automatic Control, IEEE Transactions on*, 53(Special Issue):109–121, 2008. 24, 25

[100] Jack Reilly and Alexandre M. Bayen. Distributed Optimization for Shared State Systems: Applications to Decentralized Freeway Control via Subnetwork Splitting. *IEEE Transactions on Intelligent Transportation Systems (under review)*, 2014. 2, 4

[101] Jack Reilly, Shideh Dashti, Mari Ervasti, Jonathan D. Bray, Steven D. Glaser, and Alexandre M. Bayen. Mobile Phones as Seismologic Sensors: Automating Data Extraction for the iShake System. *IEEE Transactions on Automation Science and Engineering*, 10(2):242–251, April 2013. 3

[102] Jack Reilly and Sebastien Martin. SmartRoads Website, 2014. 5, 66, 67, 72, 75, 77, 80, 81

[103] Jack Reilly, Sebastien Martin, Mathias Payer, Dawn Song, and Alexandre M. Bayen. On Cybersecurity of Freeway Control Systems: Analysis of Coordinated Ramp Metering Attacks. *Transportation Research Part B - Methodological (under review)*, 2014. 5

[104] Jack Reilly, S Samaranayake, M L Delle Monache, Walid Krichene, Paola Goatin, and A.M. Bayen. Adjoint-based optimization on a network of discretized scalar conservation

law PDEs with applications to coordinated ramp metering. *Journal of Optimization Theory and Applications (under review)*, 2014. 2, 4, 7, 44, 45, 50, 52, 54, 60

[105] James Reuther, Antony Jameson, James Farmer, Luigi Martinelli, and David Saunders. *Aerodynamic Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation.* Research Institute for Advanced Computer Science, NASA Ames Research Center, 1996. 24, 25

[106] P.I. Richards. Shock waves on the highway. *Operations research*, 4(1):42–51, 1956. 4, 15, 26, 92, 97

[107] Mike Rosenberg. Underground copper wire heist causes San Jose freeway flood. *San Jose Mercury News*, 2014. 63

[108] B Rothfarb, H Frank, D M Rosenbaum, K Steiglitz, and D J Kleitman. Optimal design of offshore natural-gas pipeline systems. *Operations Research*, 18(6):992–1020, 1970. 24

[109] T. Roughgarden. Stackelberg scheduling strategies. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 104–113. ACM, 2001. 87, 92, 93, 95, 96

[110] T. Roughgarden and E. Tardos. How bad is selfish routing? *Journal of the ACM (JACM)*, 49(2):236–259, 2002. 3, 93

[111] Tim Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003. 7

[112] Tomas Tinoco De Rubira. Cvxpy, 2011. 104

[113] Samitha Samaranayake, Jack Reilly, Walid Krichene, Maria Laura Delle Monache, Paola Goatin, and Alexandre M. Bayen. Multi-commodity real-time dynamic traffic assignment with horizontal queuing. *Operations Research (to be submitted)*, 2014. 2, 4, 7, 111

[114] David Schrank, Bill Eisele, and Tim Lomax. TTIs 2012 urban mobility report. *Proceedings of the 2012 annual urban mobility report. Texas A&M Transportation Institute, Texas, USA*, 2012. 7

[115] Alexander Skabardonis, Pravin Varaiya, and KF Petty. Measuring recurrent and nonrecurrent traffic congestion. *Transportation Research Record*, 1856(03):118–124, 2003. 39

[116] Victoria Stodden. Enabling reproducible research: Licensing for scientific innovation. *Int'l J. Comm. L. & Pol'y*, 13:1–55, 2009. 72

[117] I S Strub and A M Bayen. Weak formulation of boundary conditions for scalar conservation laws: An application to highway traffic modelling. *International Journal of Robust and Nonlinear Control*, 16(16):733–748, 2006. 17, 25

[118] Issam S. Strub, Julie Percelay, Mark T. Stacey, and Alexandre M. Bayen. Inverse estimation of open boundary conditions in tidal channels. *Ocean Modelling*, 29(1):85–93, January 2009. 25

[119] Jonathan Sutton. Copper wire stolen from traffic signal, street lights in Oklahoma City. *The Oklahoman*, 2014. 63

[120] C. O. Tong and S. C. Wong. A predictive dynamic traffic assignment model in congested

capacity-constrained road networks. *Transportation Research Part B: Methodological*, 34(8):625–644, 2000. 92

[121] Nicholas Tufnell. Students hack Waze, send in army of traffic bots. *wired.co.uk*, 2014. 60

[122] Stefan Ulbrich. A sensitivity and adjoint calculus for discontinuous solutions of hyperbolic conservation laws with source terms. *SIAM journal on control and optimization*, 41(3):740–797, 2002. 24

[123] Stefan Ulbrich. Adjoint-based derivative computations for the optimal control of discontinuous solutions of hyperbolic conservation laws. *Systems & Control Letters*, 48(3):313–328, 2003. 24, 30

[124] Aswin N Venkat, Ian A Hiskens, James B Rawlings, and Stephen J Wright. Distributed mpc strategies with application to power system automatic generation control. *Control Systems Technology, IEEE Transactions on*, 16(6):1192–1206, 2008. 44

[125] Andreas Wachter and Lorenz T Biegler. *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.* 2005. 29, 37

[126] J.G. Wardrop. Some Theoretical Aspects of Road Traffic Research. *Proceedings of the Institution of Civil Engineers*, 1:325–378, 1952. 83, 93

[127] Ermin Wei and Asuman Ozdaglar. On the o(1/k) convergence of asynchronous distributed alternating direction method of multipliers. *arXiv preprint arXiv:1307.8254v1*, page 30, 2013. 44, 45, 48, 49, 109

[128] D. B. Work, S. Blandin, O. P. Tossavainen, B. Piccoli, and A. M. Bayen. A traffic model for velocity data assimilation. *Applied Mathematics Research eXpress*, 2010(1):1, 4 2010. 2, 3, 24, 25, 60, 62

[129] Kim Zetter. Hackers Can Mess With Traffic Lights to Jam Roads and Reroute Cars. *wired.com*, 2014. 60

[130] A. K. Ziliaskopoulos. A linear programming model for the single destination system optimum dynamic traffic assignment problem. *Transportation science*, 34(1):37, 2000. 5, 7, 25, 84, 98, 101