Noname manuscript No.
(will be inserted by the editor)

Adjoint-based optimization on a network of discretized scalar conservation law PDEs with applications to coordinated ramp metering

Jack Reilly[*] · **Walid Krichene**[*] · **Maria Laura Delle Monache**[†] · **Samitha Samaranayake**[*] · **Paola Goatin**[†] · **Alexandre M. Bayen**[*]

Submitted: June 14, 2014

Abstract The adjoint method provides a computationally efficient means of calculating the gradient for applications in constrained optimization. In this article, we consider a network of scalar conservation laws with general topology, whose behavior is modified by a set of control parameters in order to minimize a given objective function. After discretizing the corresponding partial differential equation models via the Godunov scheme, we detail the computation of the gradient of the discretized system with respect to the control parameters and show that the complexity of its computation scales linearly with the number of discrete state variables for networks of small vertex degree. The method is applied to solve the problem of coordinated ramp metering on freeway networks. Numerical simulations on the I15 freeway in California demonstrate an improvement in performance and running time compared to existing methods.

Keywords control of discretized PDEs, network of hyperbolic conservation laws, adjoint based optimization, transportation engineering, ramp metering

1 Introduction

In this paper, we propose a descrete adjoint approach to compute optimal ramp-metering stategies 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 gas pipeline flow [**Rothfarb1970**], supply chain [**Brunnermeier1999**], water channels [**Rabbani2010**, **litrico2009boundary**], or freeway traffic evolution [**garavello2006traffic**, **work2010traffic**, **frazzoli2002real**]. Optimization and control of these networks is an active field of research [**Gugat2005**, **Bayen2006**, **Kotsialos2004**]. More generally, numerous techniques exist for the control of conservation laws, such as, for example, backstepping [**Coron2013**, **Glass2007**], Lyapunov-based methods [**Coron2013**], and optimal control methods [**Jacquet2006**, **Blanchard2013**, **Keller2013**]. In particular, a common approach is to compute the gradient of the cost functional via the *adjoint method* [*some refs??*]. 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, see [**Ulbrich2002Sensitivity**, **Ulbrich2003AdjointBased**].

Extensive study has also been conducted on the choice of the method for effectively computing the gradient. In particular, the continuous adjoint method [**Jacquet2005**, **Gugat2005**, **Moin1994**, **Reuther1996**] 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 [**Giles2000**, **Gugat2005**, **Kotsialos2004**] 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 [**Muller2005**, **Giering1998**].

[*]University of California, Berkeley - California, USA
E-mail: jackdreilly@berkeley.edu ·
[†]Inria Sophia Antipolis - Méditerranée, Sophia Antipolis, France

It is well known that the numerical treatment of the adjoint method imposes a careful choice of the scheme to avoid the introduction of numerical errors at discontinuities [Giles, Discrete adjoint approximations with shocks]. Rigorous convergence results for optimization problems have been provided for Lax-Friedichs type schemes [**Giles2010Convergencea**] and relaxation methods [**Banda2012Adjoint**] [Banda-Herty, COAP 2010]. The case of road networks in free flow conditions is addressed in. In our more general setting, the presence of juction conditions modeling the onramp and offramp dynamics imposes the use of a modified Godunov scheme that precisely takes into account the flows at the network nodes. 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 paper. Here we made the choice of the discrete adjoint approach, which derives the gradient directly from the discretized system, thus avoiding working directly with weak boundary conditions in the continuous system [**garavello2006traffic**, **work2010traffic**, **strub2006weak**].

There exist many applications of the adjoint method for control, optimization and estimation of physical systems in engineering. Shape optimization of aircraft [**Reuther1996**, **Giles1997**, **Moin1994**] 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 [**Raffard2008**]. 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 [**Castaings2006**, **Strub2009**] and freeway traffic state estimation [**Jacqueta**].

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 [**gomes2006optimal**] for optimal ramp metering, and in [**ziliaskopoulos2000linear**] for optimal route assignment which is exact when the relaxation gap can be shown to be zero. The ramp metering technique in [**Muralidharana**] 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.

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 [**Ramon2013**, **Frejo2011**] 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 [**Ben-Akiva2003**], most strategies are based off macroscopic state parameters, such as vehicle density and the density’s relation to speed [**richards1956shock**, **lighthill1955kinematic**, **daganzo1995cell**]. Reactive metering strategies [**Papageorgiou1991**, **Papamichail**, **Kachroo2003**] use feedback from freeway loop detectors to target a desired mainline density, while predictive metering strategies [**Frejo2011**, **Kotsialos2004**, **gomes2006optimal**, **Chen1997**] 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 [**Muralidharana**].

This article develops a framework for efficient control of discretized conservation law PDE networks using the adjoint method [**Giles2000**, **Pironneau1974**] via Godunov discretization [**godunov1959**], 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. The results are demonstrated by running a coordinated ramp metering strategy on a 19 mile freeway stretch in California faster than real-time, while giving traffic performance superior to that of state of the art practitioners tools.

The rest of the article is organized as follows. Section 2 gives an overview of scalar conservation law networks and their discretization via the Godunov method, while introducing the nonlinear, finite-horizon optimal control problem. Section 3 details the adjoint method derivation for this class of problems and shows how it can be used to compute the gradient in linear time in the number of discrete state and control variables. Section 4 shows how the adjoint method can be applied to the problem of optimal coordinated ramp metering, with numerical results on a real freeway network in California shown in Section 5. Finally, some concluding remarks are given in Section 6.

2 Preliminaries

2.1 Conservation Law PDEs

In this paper we focus on scalar hyperbolic conservation laws. In particular, we consider the non-linear transport equation of the form:

$$\partial_t \rho\left(t,x\right) + \partial_x f\left(\rho\left(t,x\right)\right) = 0 \quad (t,x) \in \mathbb{R}^+ \times \mathbb{R} \tag{1}$$

where $\rho = \rho(t,x) \in \mathbb{R}^+$ is the scalar conserved quantity and $f : \mathbb{R}^+ \to \mathbb{R}^+$ is the flux function. Throughout the article we suppose that f is a stricly concave function.
The Cauchy problem to solve is then

$$\begin{cases} \partial_t \rho + \partial_x f(\rho) = 0, \; (t,x) \in \mathbb{R}^+ \times \mathbb{R}, \\ \rho(0,x) = \bar{\rho}(x), \qquad x \in \mathbb{R} \end{cases} \tag{2}$$

where $\bar{\rho}(x)$ is the initial condition. It can be shown that there exists a unique weak entropy solution for the Cauchy problem (2) as described in Definition 21.

Definition 21 *A function* $\rho \in \mathcal{C}^0(\mathbb{R}^+; \mathbf{L}^1_{loc} \cap \mathbf{BV})$ *is an admissible solution to* (2) *if* ρ *satisfies the Kružhkov entropy condition* [**Kruzhkov**] *on* $(\mathbb{R}^+ \times \mathbb{R})$, *i.e.,for every* $k \in \mathbb{R}$ *and for all* $\varphi \in \mathcal{C}^1_c(\mathbb{R}^2; \mathbb{R}^+)$,

$$\begin{aligned} &\textstyle\int_{\mathbb{R}^+} \int_{\mathbb{R}} (|\rho - k| \partial_t \varphi + \operatorname{sgn}(\rho - k)(f(\rho) - f(k)) \partial_x \varphi) dx dt \\ &\qquad + \textstyle\int_{\mathbb{R}} |\bar{\rho} - k| \varphi(0,x) dx \geq 0. \end{aligned} \tag{3}$$

For further details regarding the theory of hyperbolic conservation laws we refer the reader to [**garavello2006traffic**, **Evans1998**].

Definition 22 *Riemann Problem.*

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

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

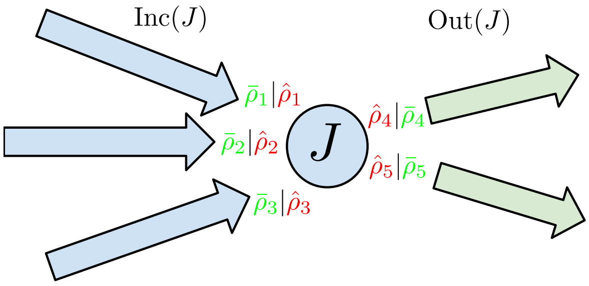
We denote the corresponding self-similar entropy weak solutions by $W_R\left(\frac{x}{t}; \rho_-, \rho_+\right)$.

2.2 Network of PDEs

A network is defined as a set of N links $\mathcal{I} = \{1, \ldots, N\}$, 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 $\operatorname{Inc}(J) = \left(i^1_J, \ldots, i^{n_J}_J\right) \subset \mathcal{I}$ and the set of m_J outgoing links $\operatorname{Out}(J) = \left(i^{n_J+1}_J, \ldots, i^{n_J+m_J}_J\right) \subset \mathcal{I}$. Each link $i \in \mathcal{I}$ has an associated upstream junction $J^{\mathrm{U}}_i \in \mathcal{J}$ and downstream junction $J^{\mathrm{D}}_i \in \mathcal{J}$, and has an associated spatial domain $(0, L_i)$ over which the evolution of the state on link i, $\rho_i\left(t,x\right)$, solves the Cauchy problem:

$$\begin{cases} \left(\rho_i\right)_t + f\left(\rho_i\right)_x &= 0 \\ \rho_i\left(0,x\right) &= \bar{\rho}_i\left(x\right) \end{cases} \tag{4}$$

where $\bar{\rho}_i \in BV \cap L^1_{\mathrm{loc}}\left(L_i; \mathbb{R}\right)$ is the initial condition on link i. For simplicity of notation, this section considers a single junction $J \in \mathcal{J}$ with $\operatorname{Inc}(J) = (1, \ldots, n)$ and $\operatorname{Out}(J) = (n+1, \ldots, n+m)$.



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

Fig. 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, $(\bar{\rho}_1, \ldots, \bar{\rho}_5)$.

Remark 1 *There is redundancy in the labeling of the junctions, if link i is directly upstream of link j, then we have $J_i^D = J_j^U$. See Fig. 2.*

While the dynamics on each link $\rho_i(t,x)$ is determined by (4), the dynamics at junctions still needs to be defined.

Definition 23 *Riemann problem at junctions.*
A Riemann problem at J is a Cauchy problem corresponding to an initial datum $(\bar{\rho}_1, \ldots, \bar{\rho}_{n+m}) \in \mathbb{R}^{n+m}$ which is constant on each link i.

Definition 24 *A Riemann solver is a map that assigns a solution to each Riemann initial data. For each junction J it is a function*

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

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

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

$$\begin{cases} (\rho_i)_t + f(\rho_i)_x & = 0 \\ \rho_i(0,x) & = \begin{cases} \bar{\rho}_i & x < 0 \\ \hat{\rho}_i & x \geq 0, \end{cases} \end{cases} \tag{5}$$

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

$$\sum_{i \in \mathrm{Inc}(J)} f(\hat{\rho}_i) = \sum_{j \in \mathrm{Out}(J)} f(\hat{\rho}_j).$$

  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 [**garavello2006traffic**].

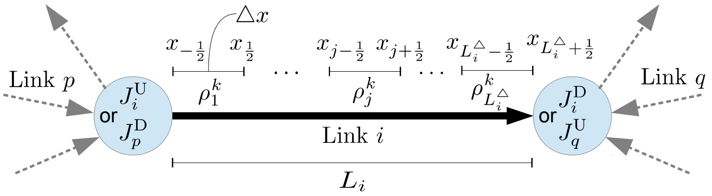

$\triangle x$
$x_{-\frac{1}{2}}$
$x_{\frac{1}{2}}$
$x_{j-\frac{1}{2}}$
$x_{j+\frac{1}{2}}$
$x_{L_i^{\triangle}-\frac{1}{2}}$
$x_{L_i^{\triangle}+\frac{1}{2}}$
Link p
J_i^{U} or J_p^{D}
ρ_1^k
ρ_j^k
$\rho_{L_i^{\triangle}}^k$
Link i
J_i^{D} or J_q^{U}
Link q
L_i

Fig. 2: Space discretization for a link $i \in \mathcal{I}$. Step size is uniform Δx, with discrete value ρ_j^k representing the state between x^{j-1} and x^j.

2.3 Godunov Discretization

In order to find approximate solutions we use the classical Godunov scheme [**godunov1959**]. 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

$$\rho_j^k = \frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \rho(t^k, x)dx. \tag{6}$$

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:

$$\rho_j^{k+1} = \rho_j^k - \frac{\Delta t}{\Delta x}(g^G(\rho_j^k, \rho_{j+1}^k) - g^G(\rho_{j-1}^k, \rho_j^k)), \tag{7}$$

where g^G is the Godunov numerical flux given by

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

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:

$$\rho_{L_i^{\Delta}}^{k+1} = \rho_{L_i^{\Delta}}^k - \frac{\Delta t}{\Delta x}(f(\hat{\rho}_{L_i^{\Delta}}^k) - g^G(\rho_{L_i^{\Delta}-1}^k, \rho_{L_i^{\Delta}}^k)), \qquad i \in \{1, \ldots, n\}$$

where L_i^{Δ} are the number of cells for link i (see Fig. 2) and $\hat{\rho}_i^k$ is the solution of the Riemann solver $RS\left(\rho_1^k, \ldots, \rho_{n+m}^k\right)$ for link i at the junction. The same can be done for the outgoing links:

$$\rho_1^{k+1} = \rho_1^k - \frac{\Delta t}{\Delta x}(g^G(\rho_1^k, \rho_2^k) - f(\hat{\rho}_1^k)), \qquad i \in \{n+1, \ldots, n+m\}$$

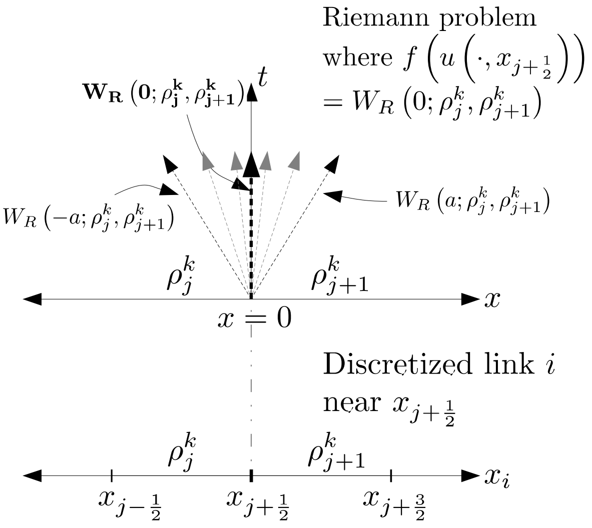

Riemann problem
where $f\left(u\left(\cdot, x_{j+\frac{1}{2}}\right)\right)$
$= W_R\left(0; \rho_j^k, \rho_{j+1}^k\right)$
$\mathbf{W_R}\left(\mathbf{0}; \rho_{\mathbf{j}}^{\mathbf{k}}, \rho_{\mathbf{j+1}}^{\mathbf{k}}\right)$
t
$W_R\left(-a; \rho_j^k, \rho_{j+1}^k\right)$
$W_R\left(a; \rho_j^k, \rho_{j+1}^k\right)$
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
$x_{j-\frac{1}{2}}$
$x_{j+\frac{1}{2}}$
$x_{j+\frac{3}{2}}$

Fig. 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. $W_R\left(0; \rho_j^k, \rho_{j+1}^k\right)$), gives the flux solution to the discretized problem in the bottom diagram.

Remark 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 $i \in \mathcal{I}$ at a time-step $k \in \{0, \ldots, T-1\}$ is represented by the single discrete value ρ_i^k.*

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 $k \in \{0, \ldots, T-1\}$ be represented as $\boldsymbol{\rho}_J^k := \left(\rho^k_{i_J^1}, \ldots, \rho^k_{i_J^{n_J+m_J}}\right)$. Similarly, we let the solution of a Riemann solver be represented as $\hat{\boldsymbol{\rho}}_J := RS\left(\boldsymbol{\rho}_J\right)$. Then, for a link $i \in \mathcal{I}$ with upstream and downstream junctions, J_i^{U} and J_i^{D}, and time-step $k \in \{0, \ldots, T-1\}$, the update step becomes:

$$\begin{aligned}\rho_i^{k+1} &= \rho_i^k - \frac{\Delta t}{\Delta x}\left(f\left(\left(RS\left(\boldsymbol{\rho}^k_{J_i^{\mathrm{D}}}\right)\right)_i\right) - f\left(\left(RS\left(\boldsymbol{\rho}^k_{J_i^{\mathrm{U}}}\right)\right)_i\right)\right)\\ &= \rho_i^k - \frac{\Delta t}{\Delta x}\left(f\left(\left(\hat{\boldsymbol{\rho}}_{J_i^{\mathrm{D}}}\right)_i\right) - f\left(\left(\hat{\boldsymbol{\rho}}_{J_i^{\mathrm{U}}}\right)_i\right)\right)\end{aligned} \tag{8}$$

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, $\hat{\boldsymbol{\rho}}_J$, and, instead, directly calculate the flux solution from the Riemann data. We denote this direct computation by g_J^G, the Godunov flux solution at a junction:

$$\begin{aligned} g_J^G : \mathbb{R}^{n_J+m_J} &\to \mathbb{R}^{n_J+m_J}\\ \boldsymbol{\rho}_J \quad &\mapsto f\left(RS\left(\boldsymbol{\rho}_J\right)\right) = \left(f\left(\hat{\rho}_1\right), \ldots, f\left(\hat{\rho}_{n+m}\right)\right). \end{aligned} \tag{9}$$

This gives a simplified expressions for the update step:

$$\rho_i^{k+1} = \rho_i^k - \frac{\Delta t}{\Delta x}\left(\left(g^G_{J_i^{\mathrm{D}}}\left(\boldsymbol{\rho}^k_{J_i^{\mathrm{D}}}\right)\right)_i - \left(g^G_{J_i^{\mathrm{U}}}\left(\boldsymbol{\rho}^k_{J_i^{\mathrm{U}}}\right)\right)_i\right). \tag{10}$$

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

Algorithm 1 `Riemann solver update procedure`

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

for junction $J \in \mathcal{J}$:
        # Apply Riemann solver to $J$
        $\hat{\boldsymbol{\rho}}_J^k = RS\left(\boldsymbol{\rho}_J^k\right)$
for link $i \in \mathcal{I}$:
        # update density on link $i$ with junction fluxes
        $\rho_i^{k+1} = \rho_i^k - \frac{\Delta t}{\Delta x}\left(f\left(\left(\hat{\boldsymbol{\rho}}_{J_i^{\mathtt{D}}}^k\right)_i\right) - f\left(\left(\hat{\boldsymbol{\rho}}_{J_i^{\mathtt{U}}}^k\right)_i\right)\right)$
```

Algorithm 1 takes as input the state at a time-step k for all links $\left(\rho_i^k : i \in \mathcal{I}\right)$ and returns the state advanced by one time-step $\left(\rho_i^{k+1} : i \in \mathcal{I}\right)$. The algorithm first iterates over all junctions J, calculating all the boundary conditions, $\hat{\boldsymbol{\rho}}_J^k$. Then, the algorithm iterates over all links $i \in \mathcal{I}$ to compute the updated state ρ_i^{k+1} using the previously computed boundary conditions, as in 8.

Algorithm 2 `Godunov junction flux update procedure`

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

for link $i \in \mathcal{I}$:
        # update density on link $i$ with direct Godonuv fluxes
        $\rho_i^{k+1} = \rho_i^k - \frac{\Delta t}{\Delta x}\left(\left(g_{J_i^{\mathtt{D}}}^G\left(\boldsymbol{\rho}_{J_i^{\mathtt{D}}}^k\right)\right)_i - \left(g_{J_i^{\mathtt{U}}}^G\left(\boldsymbol{\rho}_{J_i^{\mathtt{U}}}^k\right)\right)_i\right)$
```

Algorithm 2 is similar to Algorithm 1, except that the boundary conditions $\hat{\boldsymbol{\rho}}_J^k$ are not explicitly computed, but rather the Godunov flux solution is used to update the state, as in 10. 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.

2.4 State, Control, and Governing Equations

The rest of the article focuses on controlling systems of the form in Equation (10) 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 Algorithm 2 T time-steps forward, i.e. we wish to determine the discrete state values ρ_i^k for all links $i \in \mathcal{I}$ 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 $\left(u_{j_J^1}^k, \ldots, u_{j_J^{M_J}}^k\right) \in \mathbb{R}^{M_J}$ influence the solution of the Riemann solver. Similar to the notation developed for state variables, for control variables, we define $\mathbf{u}_J^k := \left(u_{j_J^1}^k, \ldots, u_{j_J^{M_J}}^k\right)$ 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 $\boldsymbol{\rho}_J^k$, and the current control parameters $\mathbf{u}_J^k$. Then using the same notation as before, we express the Riemann solver as:

$$\begin{aligned} RS_J : \mathbb{R}^{n_J+m_J} \times \mathbb{R}^{M_J} &\to \mathbb{R}^{n_J+m_J} \\ \left(\boldsymbol{\rho}_J^k, \mathbf{u}_J^k\right) &\mapsto RS_J\left(\boldsymbol{\rho}_J^k, \mathbf{u}_J^k\right) = \hat{\boldsymbol{\rho}}_J^k. \end{aligned}$$

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

For each state variable ρ_i^k, write the corresponding update equation h_i^k:

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

This takes the following form:

$$h_i^0(\boldsymbol{\rho}, \mathbf{u}) = \rho_i^0 - \bar{\rho}_i = 0 \tag{11}$$

$$\begin{aligned} h_i^k(\boldsymbol{\rho}, \mathbf{u}) = \rho_i^k - \rho_i^{k-1} &+ \frac{\Delta t}{L_i} f\left(RS_{J_i^{\mathrm{D}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{D}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{D}}}^{k-1}\right)\right)_i \\ &- \frac{\Delta t}{L_i} f\left(RS_{J_i^{\mathrm{U}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{U}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{U}}}^{k-1}\right)\right)_i = 0 \; \forall k \in \{2, \ldots, T-1\}, \end{aligned} \tag{12}$$

or in terms of the Godunov junction flux:

$$\begin{aligned} h_i^k(\boldsymbol{\rho}, \mathbf{u}) = \rho_i^k - \rho_i^{k-1} &+ \frac{\Delta t}{\Delta x}\left(g_{J_i^{\mathrm{D}}}^G\left(\boldsymbol{\rho}_{J_i^{\mathrm{D}}}^{k}, \mathbf{u}_{J_i^{\mathrm{D}}}^{k-1}\right)\right)_i \\ &- \frac{\Delta t}{\Delta x}\left(g_{J_i^{\mathrm{U}}}^G\left(\boldsymbol{\rho}_{J_i^{\mathrm{U}}}^{k}, \mathbf{u}_{J_i^{\mathrm{U}}}^{k-1}\right)\right)_i \end{aligned} \tag{13}$$

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

3 Adjoint Based Flow Optimization

3.1 Optimal Control Problem Formulation

In addition to our governing equations $H(\boldsymbol{\rho}, \mathbf{u}) = 0$, we also introduce a cost function C, which we assume to be in C^2:

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

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 $\mathbf{u}$, while constraining the state of the system to satisfy the governing equations $H(\boldsymbol{\rho}, \mathbf{u}) = 0$, which is, again, the concatenated version of (12) or (13). We summarize this with the following optimization problem:

$$\begin{aligned} \min_{\mathbf{u}} \quad & C(\boldsymbol{\rho}, \mathbf{u}) \\ \text{subject to: } & H(\boldsymbol{\rho}, \mathbf{u}) = 0 \end{aligned} \tag{14}$$

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

3.2 Calculating the Gradient

We wish to use gradient information in order to find control values $\mathbf{u}^*$ that give locally optimal costs $C^* = C\left(\boldsymbol{\rho}\left(\mathbf{u}^*\right), \mathbf{u}^*\right)$. Since there may exist many local minima for this optimization problem (14) (which is non-convex in general), gradient methods do not guarantee global optimality of $\mathbf{u}^*$. Still, nonlinear optimization methods such as interior point optimization utilize gradient information to improve performance [**Andreas2005**].

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

$$d_{\mathbf{u}} C\left(\boldsymbol{\rho}', \mathbf{u}'\right) = \left.\frac{\partial C(\boldsymbol{\rho}, \mathbf{u})}{\partial \boldsymbol{\rho}}\right|_{\boldsymbol{\rho}', \mathbf{u}'} \frac{d\boldsymbol{\rho}}{d\mathbf{u}} + \left.\frac{\partial C(\boldsymbol{\rho}, \mathbf{u})}{\partial \mathbf{u}}\right|_{\boldsymbol{\rho}', \mathbf{u}'}. \tag{15}$$

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

$$d_{\mathbf{u}} H\left(\boldsymbol{\rho}', \mathbf{u}'\right) = \left.\frac{\partial H(\boldsymbol{\rho}, \mathbf{u})}{\partial \boldsymbol{\rho}}\right|_{\boldsymbol{\rho}', \mathbf{u}'} \frac{d\boldsymbol{\rho}}{d\mathbf{u}} + \left.\frac{\partial H(\boldsymbol{\rho}, \mathbf{u})}{\partial \mathbf{u}}\right|_{\boldsymbol{\rho}', \mathbf{u}'} = 0. \tag{16}$$

The partial derivative terms, $H_{\boldsymbol{\rho}} \in \mathbb{R}^{NT \times NT}$, $H_{\mathbf{u}} \in \mathbb{R}^{NT \times MT}$, $C_{\boldsymbol{\rho}} \in \mathbb{R}^{NT}$, and $C_{\mathbf{u}} \in \mathbb{R}^{MT}$, can all be evaluated (more details provided in Section 3.3) and then treated as constant matrices. Thus, in order to evaluate $d_{\mathbf{u}} C\left(\boldsymbol{\rho}', \mathbf{u}'\right) \in \mathbb{R}^{MT}$, we must solve a coupled system of matrix equations.

Note 1 In (16), $H_{\boldsymbol{\rho}}$ and $H_{\mathbf{u}}$ might not necessarily be defined, either because f itself is not smooth (note that we took f to be C^2 to avoid this problem), or because g^G is not smooth. The derivations below are valid when the partials $H_{\boldsymbol{\rho}}$ and $H_{\mathbf{u}}$ can indeed be taken. There are several settings in which the conditions for differentiability are satisfied, see in particular [**Gugat2005**, **Flasskamp2012**].

Forward system. If we solve for $\frac{d\boldsymbol{\rho}}{d\mathbf{u}} \in \mathbb{R}^{NT \times MT}$ in (16), which we call the *forward system*:

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

then we can substitute the solved value for $\frac{d\boldsymbol{\rho}}{d\mathbf{u}}$ into (15) to obtain the full expression for the gradient. Section 3.3 below gives details on the invertibility of $H_{\boldsymbol{\rho}}$, guaranteeing a solution for $\frac{d\boldsymbol{\rho}}{d\mathbf{u}}$.

Adjoint system. Instead of evaluating $\frac{d\boldsymbol{\rho}}{d\mathbf{u}}$ 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):

$$H_{\boldsymbol{\rho}}^T \lambda = -C_{\boldsymbol{\rho}}^T \tag{17}$$

Then the expression for the gradient becomes:

$$d_{\mathbf{u}} C\left(\boldsymbol{\rho}', \mathbf{u}'\right) = \lambda^T H_{\mathbf{u}} + C_{\mathbf{u}} \tag{18}$$

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

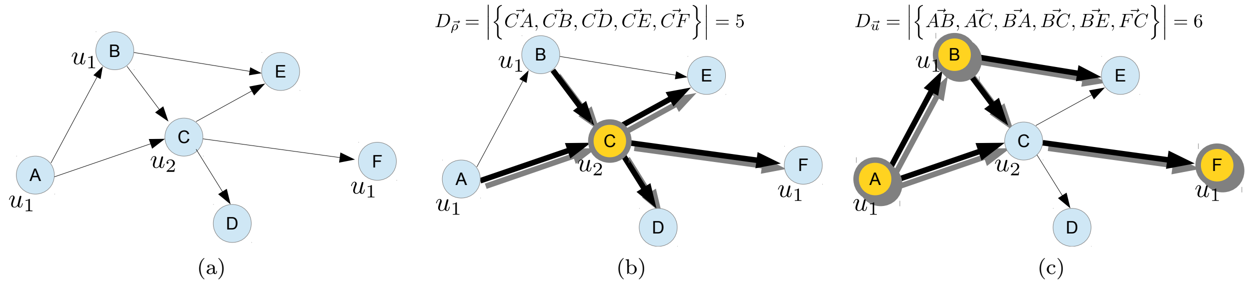
$$D_{\boldsymbol{\rho}} = \max_{J \in \mathcal{J}} \left(n_J + m_J\right), \tag{19}$$

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

$$D_{\mathbf{u}} = \max_{u \in \mathbf{u}} \sum_{J \in \mathcal{J} : u \in \mathbf{u}_J^k} \left(n_J + m_J\right). \tag{20}$$

Note that $\left\{u \in \mathbf{u}_J^k : J \in \mathcal{J}\right\}$ 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. $u \in \mathbf{u}_J^k$, then $\forall k$, $u \in \mathbf{u}_J^k$.

Using these definitions, we show later in Section 3.4 how the complexity of computing the gradient is reduced from $O(D_{\boldsymbol{\rho}} NMT^2)$ to $O(T\left(D_{\boldsymbol{\rho}} N + D_{\mathbf{u}} M\right))$ by considering the adjoint method over the forward method.



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

Fig. 4: Depiction of $D_{\boldsymbol{\rho}}$ and D_v for an arbitrary graph. Fig. 4a 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. 4b depicts the center junction having the largest number of connecting edges, thus giving $D_{\boldsymbol{\rho}} = 5$. Fig. 4c 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 $D_{\mathbf{u}} = 6$. Note that in Fig. 4c, the link going from junction A to junction B is counted twice: once as an outgoing link **AB** and once as in incoming link **BA**.

A graphical depiction of $D_{\boldsymbol{\rho}}$ and $D_{\mathbf{u}}$ are given in Fig. 4. 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 (2.4), a single control variable's influence over state variables will not grow with the size of the network. Since the $D_{\boldsymbol{\rho}}$ and $D_{\mathbf{u}}$ 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.

3.3 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 about the structure and sparsity of $H_{\boldsymbol{\rho}}$ and $H_{\mathbf{u}}$.

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 $H_{\boldsymbol{\rho}}$ and $H_{\mathbf{u}}$ matrices.

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

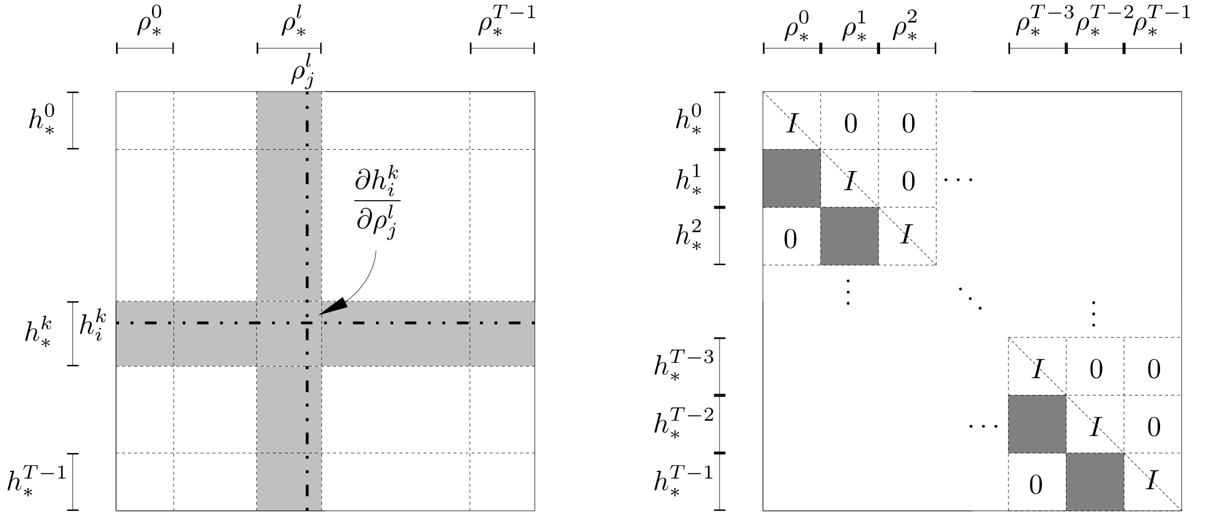
$$\begin{aligned}\frac{\partial h_i^k}{\partial \rho_j^l} = \frac{\partial \rho_i^k}{\partial \rho_j^l} - \frac{\partial \rho_i^{k-1}}{\partial \rho_j^l} &+ \frac{\Delta t}{L_i} f'\left(RS_{J_i^{\mathrm{D}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{D}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{D}}}^{k-1}\right)_i\right) \frac{\partial}{\partial \rho_j^l}\left(RS_{J_i^{\mathrm{D}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{D}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{D}}}^{k-1}\right)_i\right) \\ &- \frac{\Delta t}{L_i} f'\left(RS_{J_i^{\mathrm{U}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{U}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{U}}}^{k-1}\right)_i\right) \frac{\partial}{\partial \rho_j^l}\left(RS_{J_i^{\mathrm{U}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{U}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{U}}}^{k-1}\right)_i\right)\end{aligned} \tag{21}$$

or if we consider the composed Riemann flux solver g_J^G in (9):

$$\frac{\partial h_i^k}{\partial \rho_j^l} = \frac{\partial \rho_i^k}{\partial \rho_j^l} - \frac{\partial \rho_i^{k-1}}{\partial \rho_j^l} + \frac{\Delta t}{L_i}\left(\frac{\partial}{\partial \rho_j^l}\left(g_{J_i^{\mathrm{D}}}^G\left(\boldsymbol{\rho}_{J_i^{\mathrm{D}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{D}}}^{k-1}\right)\right)_i - \frac{\partial}{\partial \rho_j^l}\left(g_{J_i^{\mathrm{U}}}^G\left(\boldsymbol{\rho}_{J_i^{\mathrm{U}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{U}}}^{k-1}\right)\right)_i\right) \tag{22}$$

A diagram of the structure of the $H_{\boldsymbol{\rho}}$ matrix is given in Fig. (5a). Similarly for $H_{\mathbf{u}}$, we take a control parameter $u_j^l \in \mathbf{u}$, and derive the expression:

$$\begin{aligned}\frac{\partial h_i^k}{\partial u_j^l} = &+ \frac{\Delta t}{L_i} f'\left(RS_{J_i^{\mathrm{D}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{D}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{D}}}^{k-1}\right)_i\right) \frac{\partial}{\partial u_j^l}\left(RS_{J_i^{\mathrm{D}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{D}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{D}}}^{k-1}\right)_i\right) \\ &- \frac{\Delta t}{L_i} f'\left(RS_{J_i^{\mathrm{U}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{U}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{U}}}^{k-1}\right)_i\right) \frac{\partial}{\partial u_j^l}\left(RS_{J_i^{\mathrm{U}}}\left(\boldsymbol{\rho}_{J_i^{\mathrm{U}}}^{k-1}, \mathbf{u}_{J_i^{\mathrm{U}}}^{k-1}\right)_i\right)\end{aligned} \tag{23}$$



(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 $H_{\boldsymbol{\rho}}$ matrix. Besides the diagonal blocks, which are identity matrices, blocks where $l \neq k-1$ are zero.

Fig. 5: Structure of the $H_{\boldsymbol{\rho}}$ matrix.

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

$$\frac{\partial h_i^k}{\partial u_j^l} = \frac{\Delta t}{L_i}\left(\frac{\partial}{\partial u_j^l}\left(g^G_{J_i^{\mathrm{D}}}\left(\boldsymbol{\rho}^{k-1}_{J_i^{\mathrm{D}}}, \mathbf{u}^{k-1}_{J_i^{\mathrm{D}}}\right)\right)_i - \frac{\partial}{\partial u_j^l}\left(g^G_{J_i^{\mathrm{U}}}\left(\boldsymbol{\rho}^{k-1}_{J_i^{\mathrm{U}}}, \mathbf{u}^{k-1}_{J_i^{\mathrm{U}}}\right)\right)_i\right). \tag{24}$$

Analyzing (21), the only partial terms that are not trivial to compute are $\frac{\partial}{\partial \rho_j^l}\left(RS_{J_i^{\mathrm{D}}}\left(\boldsymbol{\rho}^{k-1}_{J_i^{\mathrm{D}}}, \mathbf{u}^{k-1}_{J_i^{\mathrm{D}}}\right)_i\right)$ and $\frac{\partial}{\partial \rho_j^l}\left(RS_{J_i^{\mathrm{U}}}\left(\boldsymbol{\rho}^{k-1}_{J_i^{\mathrm{U}}}, \mathbf{u}^{k-1}_{J_i^{\mathrm{U}}}\right)_i\right)$. Similarly for (23), the only nontrivial terms are $\frac{\partial}{\partial u_j^l}\left(RS_{J_i^{\mathrm{D}}}\left(\boldsymbol{\rho}^{k-1}_{J_i^{\mathrm{D}}}, \mathbf{u}^{k-1}_{J_i^{\mathrm{D}}}\right)_i\right)$ and $\frac{\partial}{\partial u_j^l}\left(RS_{J_i^{\mathrm{U}}}\left(\boldsymbol{\rho}^{k-1}_{J_i^{\mathrm{U}}}, \mathbf{u}^{k-1}_{J_i^{\mathrm{U}}}\right)_i\right)$. Once one obtains the solutions to these partial terms, then one can construct the full $H_{\boldsymbol{\rho}}$ and $H_{\mathbf{u}}$ matrices and use (17) and (18) 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 (4).

3.4 Complexity of Solving Gradient via Forward Method vs. Adjoint Method

This section demostrates the following proposition:

Proposition 31 The total complexity for the adjoint method on a scalar hyperbolic network of PDEs is $O(T\left(D_{\boldsymbol{\rho}}N + D_{\mathbf{u}}M\right))$.

We can show the lower-triangular structure and invertibility of $H_{\boldsymbol{\rho}}$ by examining (11) and (12). For $k \in \{1, \ldots, T-1\}$, we have that h_i^k is only a function of ρ_i^k and of the state variables from the previous time-step $k-1$. Thus, based on our ordering scheme in Section 2.4 of ordering variables by increasing time-step and ordering constraints by corresponding variable, we know that the diagonal terms of $H_{\boldsymbol{\rho}}$ 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 $H_{\boldsymbol{\rho}}$ 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 (12) 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

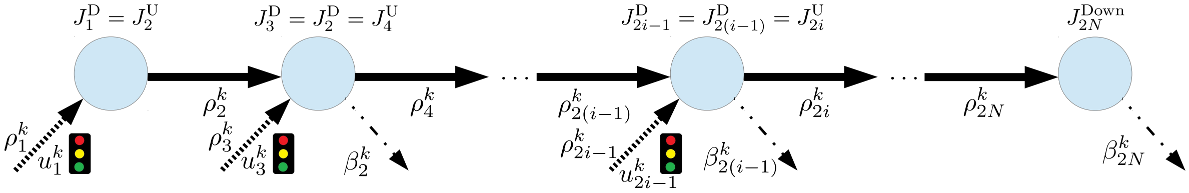


Fig. 6: Freeway network model. For a junction $J_{2i-1}^{\mathrm{D}} = J_{2(i-1)}^{\mathrm{D}} = J_{2i}^{\mathrm{U}}$ at time-step $k \in \{0, \ldots, T-1\}$, the upstream mainline density are represented by $\rho_{2(i-1)}^k$, the downstream mainline density by ρ_{2i}^k, the on-ramp density by ρ_{2i-1}^k, and the off-ramp split ratio by $\beta_{2(i-1)}^k$.

J_j^{D} and J_j^{U} respectively. To summarize, $H_{\boldsymbol{\rho}}$ matrices for systems described in Section 2.4 will be square, invertible, lower-triangular and each column will have a maximum cardinality equal to $D_{\boldsymbol{\rho}}$ in (19). The sparsity structure of $H_{\boldsymbol{\rho}}$ is depicted in Fig. 5b.

Using the same line of argument for the maximum cardinality of $H_{\boldsymbol{\rho}}$, we can bound the maximum cardinality of each column of $H_{\mathbf{u}}$. 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 $u_j^l \in \mathbf{u}_J^{l+1}$. These conditions give us the expression for $D_{\mathbf{u}}$ in (20), or the maximum cardinality over all columns in $H_{\mathbf{u}}$.

If we only consider the lower triangular form of $H_{\boldsymbol{\rho}}$, then the complexity of solving for the gradient using the forward system is $O((NT)^2\,MT)$, where the dominating term comes from solving (15), 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 (17) 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 (18), 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 $H_{\boldsymbol{\rho}}$ and $H_{\mathbf{u}}$ matrices, as detailed in Section 3.4. For the backward-substitution step, each entry in the λ vector is solved by *at most* $D_{\boldsymbol{\rho}}$ multiplications, and thus the complexity of solving (17) is reduced to $O(D_{\boldsymbol{\rho}}NT)$. 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 $D_{\mathbf{u}}$ multiplications, giving a complexity of $O(D_{\mathbf{u}}MT)$.

4 Applications to Optimal Coordinated Ramp Metering on Freeways

4.1 Formulation of the Network Model And Explicit Riemann Solver

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

The off-ramp directly downstream of link $2i, i \in \{1, \ldots, N\}$ has, at time-step k, a split ratio β_{2i}^k representing the ratio of cars which stay on the freeway over the total cars leaving the upstream mainline of junction J_{2i}^{D}. The model assumes that all flux from on-ramp $2i-1$ enters downstream mainline $2i$. Since J_2^{U} is the source of the network, it has no upstream mainline or off-ramp, and similarly J_{2N}^{D} has no downstream mainline or on-ramp ($\beta_{2N}^k = 0$). Each link $i \in \mathcal{I}$ has a discretized state value $\rho_i^k \in \mathbb{R}$ 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. 6. 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.

The vehicle flow dynamics on all links i (mainlines, on-ramps, and off-ramps) are modeled using the conservation law governing the density evolution (1), where ρ is the density state, and f is the flux function (or fundamental diagram) $f(\rho)$. In the context of traffic, this model is referred to as the Lighthill-Whitham-Richards (LWR) model [**lighthill1955kinematic**, **richards1956shock**]. The fundamental diagram f is typically assumed to be concave, and has a bounded domain $[0, \rho^{\max}]$ and a maximum flux value $F^{\max}$ attained at a critical density $\rho^c : f(\rho^c) = F^{\max}$. We assume that the fundamental diagram has a trapezoidal form as depicted in Fig. 7. For the remainder of the article, we instantiate the conservation law in (1) with the LWR equation as it applies to traffic flow modeling.

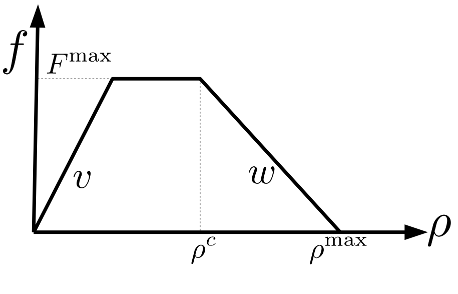


Fig. 7: Fundamental diagram (the name of the flux function in transportation literature) with free-flow speed v, congestion wave speed w, max flux $F^{\max}$, critical density ρ^c, and max density $\rho^{\max}$.

As control input, an on-ramp $2i-1 \in \mathcal{I}, i \in \{1,\ldots,N\}$ at time-step $k \in \{0,\ldots,T-1\}$ has a metering rate $u^k_{2i-1} \in [0,1]$ 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 (31).

For notational simplicity we define the set of densities of links incident to $J^{\mathrm{U}}_{2i} = J^{\mathrm{D}}_{2(i-1)}$ at time-step k as $\boldsymbol{\rho}^k_{J^{\mathrm{U}}_{2i}} = \left\{\rho^k_{2(i-1)}, \rho^k_{2i-1}, \rho^k_{2i}\right\}$. The off-ramp is considered to have infinite capacity, and thus has no bearing on the solution of junction problems. Initial conditions are handled as in (11), while for $k \in \{1,\ldots,T-1\}$, the mainline density ρ^k_{2i} using the Godunov scheme from (12) is given by:

$$\begin{aligned} h^k_{2i}(\boldsymbol{\rho},\mathbf{u}) &= \rho^k_{2i} - \rho^{k-1}_{2i} + \frac{\Delta t}{L_{2i}}\left(g^G_{J^{\mathrm{D}}_{2i}}\left(\boldsymbol{\rho}^{k-1}_{J^{\mathrm{D}}_{2i}}, u^{k-1}_{2i+1}\right)\right)_{2i} \qquad (25)\\ &\quad - \frac{\Delta t}{L_{2i}}\left(g^G_{J^{\mathrm{U}}_{2i}}\left(\boldsymbol{\rho}^{k-1}_{J^{\mathrm{U}}_{2i}}, u^{k-1}_{2i-1}\right)\right)_{2i} \\ &= \rho^k_{2i} - \rho^{k-1}_{2i} + \frac{\Delta t}{L_{2i}}\left(g^{k-1}_{2i,\mathrm{D}} - g^{k-1}_{2i,\mathrm{U}}\right) = 0 \qquad (26)\end{aligned}$$

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

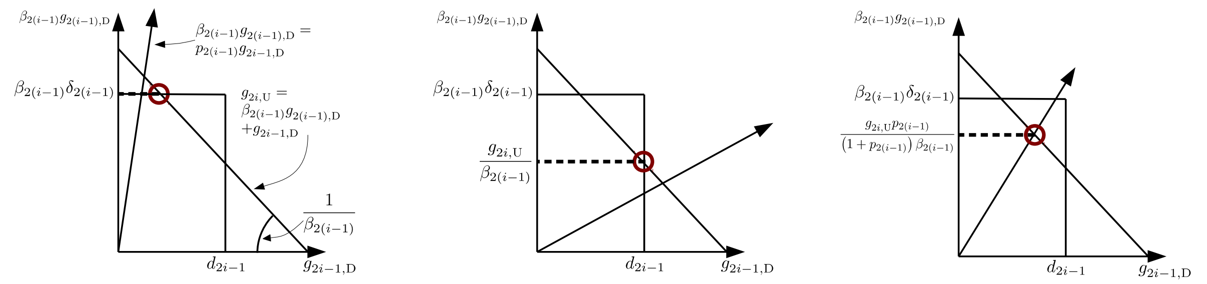
We also make the assumption that on-ramps have infinite capacity and a free-flow velocity $v_{2i-1} = \frac{L_{2i-1}}{\Delta t}$ to prevent the ramp congestion from blocking demand from ever entering the network. Since the on-ramp has no physical length, the length is chosen arbitrarily and the "virtual" velocity chosen above is chosen to replicate the dynamics in [**Monache2013**]. We can then simplify the on-ramp update equation to be:

$$\begin{aligned} h^k_{2i-1}(\boldsymbol{\rho},\mathbf{u}) &= \rho^k_{2i-1} - \rho^{k-1}_{2i-1} - \frac{\Delta t}{L_{2i-1}}\left(\left(g^G_{J^{\mathrm{U}}_{2i}}\left(\boldsymbol{\rho}^{k-1}_{J^{\mathrm{U}}_{2i}}, u^{k-1}_{2i-1}\right)\right)_{2i-1} - D^{k-1}_{2i-1}\right) \qquad (27)\\ &= \rho^k_{2i-1} - \rho^{k-1}_{2i-1} - \frac{\Delta t}{L_{2i-1}}\left(g^{k-1}_{2i-1,\mathrm{D}} - D^{k-1}_{2i-1}\right) = 0 \qquad (28)\end{aligned}$$

where D^{k-1}_{2i-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. Details on weak versus strong boundary conditions can be found in [**Monache2013**, **strub2006weak**, **work2010traffic**].

The on-ramp model in (27) differs from [**Monache2013**] in that we model the on-ramp as a discretized PDE with an infinite critical density, while [**Monache2013**] 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 article.

Riemann solver. For the ramp metering problem, there are many potential Riemann solvers that satisfy the properties required in Section 2.2. Following the model of [**Monache2013**], for each junction J^{U}_{2i}, we add two modeling decisions:


$\beta_{2(i-1)}g_{2(i-1),\mathrm{D}}$
$\beta_{2(i-1)}g_{2(i-1),\mathrm{D}} = p_{2(i-1)}g_{2i-1,\mathrm{D}}$
$\beta_{2(i-1)}\delta_{2(i-1)}$
$g_{2i,\mathrm{U}} = \beta_{2(i-1)}g_{2(i-1),\mathrm{D}} + g_{2i-1,\mathrm{D}}$
$\frac{1}{\beta_{2(i-1)}}$
d_{2i-1}
$g_{2i-1,\mathrm{D}}$
$\beta_{2(i-1)}g_{2(i-1),\mathrm{D}}$
$\beta_{2(i-1)}\delta_{2(i-1)}$
$\frac{g_{2i,\mathrm{U}}}{\beta_{2(i-1)}}$
d_{2i-1}
$g_{2i-1,\mathrm{D}}$
$\beta_{2(i-1)}g_{2(i-1),\mathrm{D}}$
$\beta_{2(i-1)}\delta_{2(i-1)}$
$\frac{g_{2i,\mathrm{U}}p_{2(i-1)}}{\left(1+p_{2(i-1)}\right)\beta_{2(i-1)}}$
d_{2i-1}
$g_{2i-1,\mathrm{D}}$

(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.

Fig. 8: Godunov junction flux solution for ramp metering model at junction J^{U}_{2i}. The rectangular region represents the feasible flux values for $\beta_{2(i-1)}g_{2(i-1),\mathrm{D}}$ and $g_{2i-1,\mathrm{D}}$ as determined by the upstream demand, while the line with slope $\frac{1}{\beta_{2(i-1)}}$ represents feasible flux values as determined by mass balance. The $\beta_{2(i-1)}g_{2(i-1),\mathrm{D}}$ term accounts for only the flux out of link $2\,(i-1)$ 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 $\beta_{2(i-1)}g_{2(i-1),\mathrm{D}} = p_{2(i-1)}g_{2i-1,\mathrm{D}}$.

1. The flux solution maximizes the outgoing mainline flux $g^k_{2i,\mathrm{U}}$.
2. Subject to (1), the flux solution attempts to satisfy $g^k_{2(i-1),\mathrm{D}} = p_{2(i-1)}g^k_{2i-1,\mathrm{D}}$, where $p_{2(i-1)} \in \mathbb{R}_+$ is a merging parameter for junction $J^{\mathrm{D}}_{2(i-1)}$. Since (1) allows multiple flux solutions at the junction, (2) is necessary to obtain a unique solution.

This leads to the following system of equations that gives the flux solution of the Riemann solver at time-step $k \in \{1, \ldots, T-1\}$ and junction J^{U}_{2i} for $i \in \{1, \ldots, N\}$:

$$\delta^k_{2(i-1)} = \min\left(v_{2(i-1)}\rho^k_{2(i-1)}, F^{\max}_{2(i-1)}\right) \tag{29}$$

$$\sigma^k_{2i} = \min\left(w_{2i}\left(\rho^{\max}_{2i} - \rho^k_{2i}\right), F^{\max}_{2i}\right) \tag{30}$$

$$d^k_{2i-1} = u^k_{2i-1}\min\left(\frac{L_{2i-1}}{\Delta t}\rho^k_{2i-1}, F^{\max}_{2i-1}\right) \tag{31}$$

$$g^k_{2i,\mathrm{U}} = \min\left(\beta^k_{2(i-1)}\delta^k_{2(i-1)} + d^k_{2i-1}, \sigma^k_{2i}\right) \tag{32}$$

$$g^k_{2(i-1),\mathrm{D}} = \begin{cases} \delta^k_{2(i-1)} & \frac{p_{2(i-1)}g^k_{2i,\mathrm{U}}}{\beta^k_{2(i-1)}\left(1+p_{2(i-1)}\right)} \geq \delta^k_{2(i-1)} \text{[Case 1]} \\ \frac{g^k_{2i,\mathrm{U}} - d^k_{2i-1}}{\beta^k_{2(i-1)}} & \frac{g^k_{2i,\mathrm{U}}}{1+p_{2(i-1)}} \geq d^k_{2i-1} \qquad \text{[Case 2]} \\ \frac{p_{2(i-1)}g^k_{2i,\mathrm{U}}}{\left(1+p_{2(i-1)}\right)\beta^k_{2(i-1)}} & \text{otherwise} \qquad\qquad \text{[Case 3]} \end{cases} \tag{33}$$

$$g^k_{2i-1,\mathrm{D}} = g^k_{2i,\mathrm{U}} - \beta^k_{2(i-1)}g^k_{2(i-1),\mathrm{D}} \tag{34}$$

where, for notational simplicity, at the edges of of the range for i, any undefined state values (e.g. ρ^k_0) are assumed to be zero by convention. Equations (29) and (31) determine the maximum flux that can exit link $2(i-1)$ and link $2i-1$ respectively. Equation (30) gives the maximum flux allowed into link $2i$. The actual flux into link $2i$, shown in (32), is given as the minimum of the "demand" from upstream links and "supply" of the downstream link. See [**Monache2013**] for more details on the model for this equation. The flux out of link $2(i-1)$ is split into three cases in (33). The solutions are depicted in Fig. 8, which demonstrates how the flux solution depends upon the respective demands and the merging parameter $p_{2(i-1)}$. Finally, Equation (34) gives the flux out of the on-ramp $2i-1$, which is the difference between the flux into link $2i$ and the flux out of link $2\,(i-1)$ the remains on the mainline.

For $k = 0$, the update equation is given by a pre-specified initial condition, as in (11). 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 $\beta^k_{2(i-1)} g^k_{2(i-1),\mathrm{D}}$.

4.2 Formulation of the Optimal Control Problem

Optimal coordinated ramp-metering. Including the initial conditions as specified in (11) with (25) and (27) gives a complete description of the system $H(\boldsymbol{\rho}, \mathbf{u}) = 0$, $\boldsymbol{\rho} \in \mathbb{R}^{2N}$, $\mathbf{u} \in \mathbb{R}$, where:

$$\begin{aligned} \boldsymbol{\rho}_{2Nk+i} &:= \rho^k_i \;\; 1 \le i \le 2N, 0 \le k \le T-1 \\ \mathbf{u}_{Nk+i} &:= u^k_{2i} \;\; 1 \le i \le N, 0 \le k \le T-1 \end{aligned}$$

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

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

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

$$\begin{aligned} \min_{\mathbf{u}} \;\; & C(\boldsymbol{\rho}, \mathbf{u}) \\ \text{subject to:} \;\; & H(\boldsymbol{\rho}, \mathbf{u}) \;\; = 0 \\ & 0 \le u \le 1 \; \forall u \in \mathbf{u} \end{aligned} \tag{35}$$

Since the adjoint method in Section 3 only deals with equality constraints, we add barrier penalties to the cost function [**Boyd2010**, **Bayen2006**]:

$$\tilde{C}(\boldsymbol{\rho}, \mathbf{u}, \epsilon) = C(\boldsymbol{\rho}, \mathbf{u}) - \epsilon \sum_{u \in \mathbf{u}} \log\left((1-u)(u-0)\right). \tag{36}$$

As $\epsilon \in \mathbb{R}^+$ tends to zero, the solution to (36) will approach the solution to the original problem (35). Thus we can solve (35) by iteratively solving the augmented problem:

$$\begin{aligned} \min_{\mathbf{u}} \; & \tilde{C}(\boldsymbol{\rho}, \mathbf{u}, \epsilon) \\ \text{subject to:} \;\; & H(\boldsymbol{\rho}, \mathbf{u}) \;\; = 0 \end{aligned} \tag{37}$$

with decreasing values of ϵ. As a result, $\tilde{C}$ will approach C as the number of iterations increases.

Applying the adjoint method. To use the adjoint method as described in Section 3, we need to compute the partial derivative matrices $H_{\boldsymbol{\rho}}$, $H_{\mathbf{u}}$, $\tilde{C}_{\boldsymbol{\rho}}$ and $\tilde{C}_{\mathbf{u}}$. Computing the partial derivatives with respect to the cost function is straight forward:

$$\begin{aligned} \frac{\partial \tilde{C}}{\partial \rho^k_i} &= \qquad \Delta t L_i \qquad 1 \le i \le 2N, 0 \le k \le T-1 \\ \frac{\partial \tilde{C}}{\partial u^k_{2i}} &= \epsilon \left(\tfrac{1}{1-u^k_{2i}} - \tfrac{1}{u^k_{2i}} \right) \; 1 \le i \le N, 0 \le k \le T-1 \end{aligned}$$

To compute the partial derivatives of H, we follow the procedure in Section 3.2. For an upstream junction $J^{\mathrm{U}}_{2i} \in \mathcal{J}$ and time-step $k \in \{1, \ldots, T-1\}$, we only need to compute the partial derivatives of the flux solver $g^G_{J^{\mathrm{U}}_{2i}} \left(\boldsymbol{\rho}^k_{J^{\mathrm{U}}_{2i}}, u^k_{2i-1} \right)$ with respect to the adjacent state variables $\boldsymbol{\rho}^k_{J_i}$ and ramp metering control u^k_i. We calculate the partial derivatives of the functions in (29)-(34) with respect to either a state or control variable$s \in \boldsymbol{\rho} \cup \mathbf{u}$:

$$
\begin{aligned}
\frac{\partial \delta^k_{2(i-1)}}{\partial s} &= \begin{cases} v_{2(i-1)} & s = \rho^k_{2(i-1)}, v_i \rho^k_{2(i-1)} \leq F^{\max}_{2(i-1)} \\ 0 & \text{otherwise} \end{cases} \\
\frac{\partial \sigma^k_{2i}}{\partial s} &= \begin{cases} -w_{2i} & s = \rho^k_{2i}, w_{2i} \left(\rho^{\max}_{2i} - \rho^k_{2i}\right) \leq F^{\max}_{2i} \\ 0 & \text{otherwise} \end{cases} \\
\frac{\partial d}{\partial s} &= \begin{cases} u^k_{2i-1} & s = \rho^k_{2i-1}, \rho^k_{2i-1} \leq F^{\max}_{2i-1} \\ \min\left(\rho^k_{2i-1}, F^{\max}_{2i-1}\right) & s = u^k_{2i-1} \\ 0 & \text{otherwise} \end{cases} \\
\frac{\partial}{\partial s} g^k_{2i,\mathrm{U}} &= \begin{cases} \beta^k_{2(i-1)} \frac{\partial \delta^k_{2(i-1)}}{\partial s} + \frac{\partial d^k_{2(i-1)}}{\partial s} & \beta^k_{2(i-1)} \delta^k_{2(i-1)} + d^k_{2i-1} \leq \sigma^k_{2i} \\ \frac{\partial \sigma^k_{2i}}{\partial s} & \text{otherwise} \end{cases} \\
\frac{\partial}{\partial s} g_{2(i-1),\mathrm{D}} &= \begin{cases} \frac{\partial \delta^k_{2(i-1)}}{\partial s} & \frac{g^k_{2i,\mathrm{U}} p_{2(i-1)}}{1+p_{2(i-1)}} \geq \frac{\delta^k_{2(i-1)}}{\beta^k_{2(i-1)}} \\ \frac{1}{\beta^k_{2(i-1)}} \left(\frac{\partial}{\partial s} g^k_{2i,\mathrm{U}} - \frac{\partial d^k_{2i-1}}{\partial s} \right) & \frac{g^k_{2i,\mathrm{U}}}{1+p_{2(i-1)}} \geq d^k_{2(i-1)} \\ \frac{p_{2(i-1)}}{\beta^k_{2(i-1)}\left(1+p_{2(i-1)}\right)} \frac{\partial}{\partial s} g^k_{2i,\mathrm{U}} & \text{otherwise} \end{cases} \\
\frac{\partial}{\partial s} g_{2i-1,\mathrm{D}} &= \frac{\partial}{\partial s} g^k_{2i,\mathrm{U}} - \beta^k_{2(i-1)} \frac{\partial}{\partial s} g_{2(i-1),\mathrm{D}}
\end{aligned}
$$

These expressions fully quantify the partial derivative values needed in (22) and (24). Thus we can construct the $H_{\boldsymbol{\rho}}$ and $H_{\mathbf{u}}$ matrices. With these matrices and $C_{\boldsymbol{\rho}}$ and $C_{\mathbf{u}}$, we can solve for the adjoint variable $\lambda \in \mathbb{R}^{2NT}$ in (17) and substitute its value into (18) to obtain the gradient of the cost function C with respect to the control parameter $\mathbf{u}$.

5 Numerical Results for Model Predictive Control Implementations

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 [**Andreas2005**], 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 [**Papageorgiou1991**]: 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 2 To demonstrate the reduced running time associated with the adjoint approach, we also implemented a gradient descent using a finite differences approach similar to [**Frejo2011**, **Ramon2013**], 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. 9. 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.

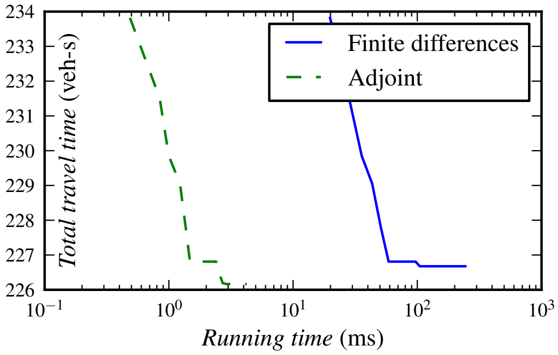


Fig. 9: Running time of ramp metering algorithm using IpOpt with and without gradient information. Network consists of 4 links and 6 time-steps with synthetic boundary flux data. The method using gradient information via the adjoint method converged well before the completion of the *first* step of the finite differences descent method.

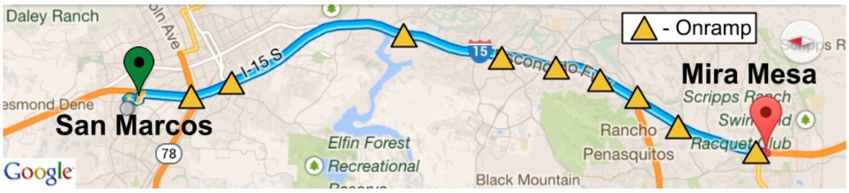


Fig. 10: Model of section of I15 South in San Diego, California. The freeway section spanning 19.4 miles was split into 125 links with 9 on-ramps.

5.1 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. 10.

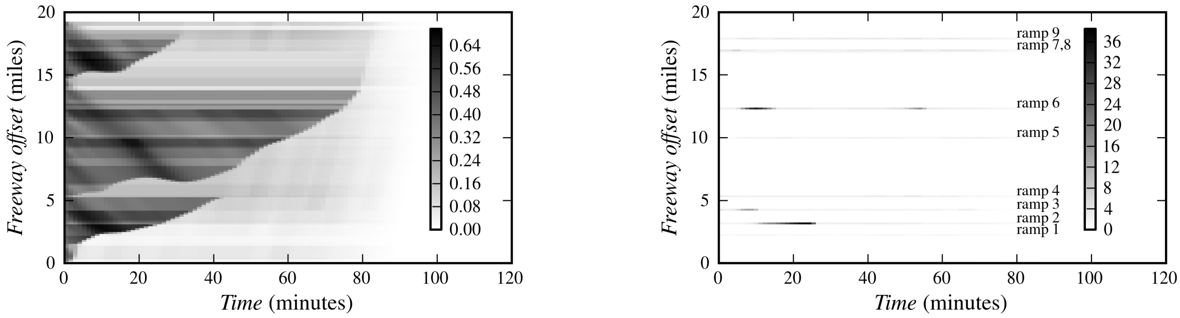
Link length data was obtained using the Scenario Editor software developed as part of the *Connected Corridors* project, a collaboration between UC Berkeley and PATH research institute in Berkeley, California. Fundamental diagram parameters, split ratios, and boundary data were also obtained using calibration techniques developed by Connected Corridors. 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 [**Chen2003**] 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. 11 under the described boundary conditions.

5.2 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:

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

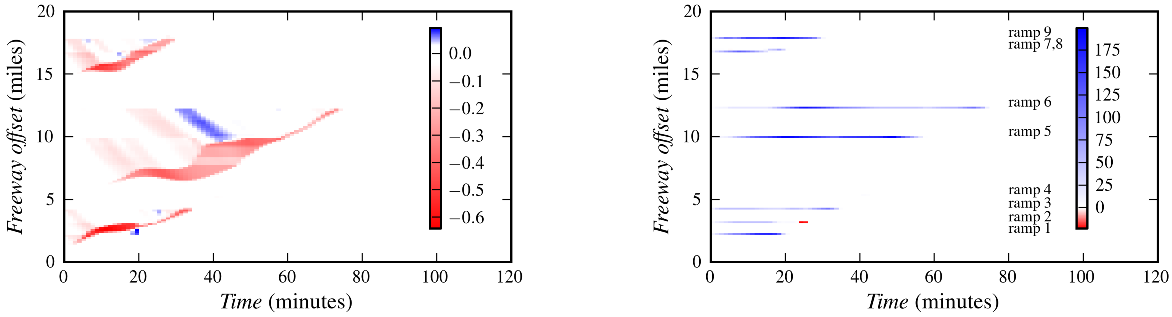
where $c_{\mathrm{c}}, c_{\mathrm{nc}} \in \mathbb{R}_+$ are the *congestion* resulting from the *control* and *no-control* scenarios, respectively. We use the metric for congestion as defined in [**Skabardonis2003**]; for a given section of road S and time



(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.

Fig. 11: 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.



(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.

Fig. 12: 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. *View in color.*

horizon T, the congestion is given as

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

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.

Results. Fig. 12 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. 12a and 12b 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. 13 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.

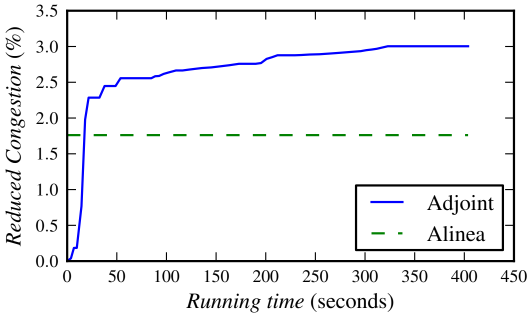


Fig. 13: Reduced congestion versus simulation time for freeway network. The results indicate that the algorithm can run with performance better than Alinea if given an update time of less than a minute.

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.

5.3 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:

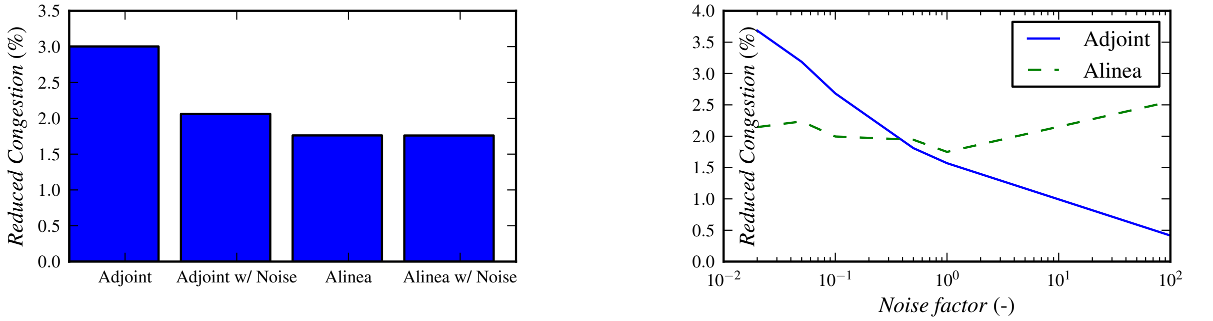
$$\bar{\rho} = \rho \cdot (1 + \sigma \cdot R)$$

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.
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. 14a. 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



(a) Reduced congestion.

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

Fig. 14: 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.

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. 14b. 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.

6 Conclusions

This article 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 [**CC**] system, a project by UC Berkeley and PATH for integrated corridor management, as a component of the traffic simulator module. Future work includes investigating decentralized, coordinated control schemes over physical networks via the adjoint method to allow traffic control strategies to scale to regional-scale networks.

7 Acknowledgments

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

Nomenclature

Variable	Space	Meaning
t	$\mathbb{R}_+$	time
x	$\mathbb{R}$	space
N	$\mathbb{N}$	Number of links
$\mathcal{J}$		Set of junctions
$\mathcal{I} = [1, N]$	$\mathbb{N}^N$	Set of links
L_i	$\mathbb{R}_+$	Length of link $i \in \mathcal{I}$
$\rho_i\left(t, x\right)$	$\mathbb{R}_+ \times]0, L_i[\to \mathbb{R}$	conserved quantity for link $i \in \mathcal{I}$ as function of x
$\bar{\rho}$	$BV \cap L^1_{\mathrm{loc}}$	continuous intialial condition
ρ_i^k	$\mathbb{R}$	discrete conserved quantity for link i at time-step k
$f(\rho)$	$f : \mathbb{R} \to \mathbb{R}$	flux function
$\bar{\rho}$	$\mathbb{R}$	Riemann data
ρ^-	$\mathbb{R}$	Left state of the Riemann data
ρ^+	$\mathbb{R}$	Right state of the Riemann data
$\bar{x}$	$\mathbb{R}$	Point of discontinuity in Riemann problem
W_R	$\mathbb{R}$	Self-similar solution of the Riemann problem
n_J	$\mathbb{N}$	Number of incoming links at a junction J
m_J	$\mathbb{N}$	Number of outgoing links at a junction J
$\mathrm{Inc}\left(J\right) = \left(i_J^1, \ldots, i_J^{n_J}\right) \subset \mathcal{I}$		Set of incoming links at a junction J
$\mathrm{Out}\left(J\right) = \left(i_J^{n_J+1}, \ldots, i_J^{n_J+m_J}\right) \subset \mathcal{I}$		Set of outgoing links at a junction J
$J_i^{\mathrm{U}} \in \mathcal{J}$		Upstream junction for the link $i \in \mathcal{I}$
$J_i^{\mathrm{D}} \in \mathcal{J}$		Downstream junction for the link $i \in \mathcal{I}$
RS	$\mathbb{R}^{m+n} \to \mathbb{R}^{m+n}$	Riemann Solver
$\hat{\rho}_i$	$\mathbb{R}^{m+n}$	Trace for a link i at the junction
$\varDelta t$	$\mathbb{R}$	Time grid size
$\varDelta x$	$\mathbb{R}$	Space grid size
$t^k = k\varDelta t$	$k \in \mathbb{N}$	Time grid points
$t^j = l\varDelta x$	$l \in \mathbb{Z}$	Space grid points
$\lambda^{\max}$	$\mathbb{R}$	Wave speed
$\boldsymbol{\rho}_J^k$	$\mathbb{R}^{m_J+n_J}$	state variables at a junction $J \in \mathcal{J}$ at a time-step k
T	$\mathbb{N}$	Number of time steps
$\hat{\boldsymbol{\rho}}_J$	$\mathbb{R}^{m_J+n_J}$	solution of RS at a junction $J \in \mathcal{J}$ at a time-step k
$\mathbf{u}_J^k$	$\mathbb{R}^{M_J}$	control variables at a junction $J \in \mathcal{J}$ at a time-step k
h_i^k	$\mathbb{R}^{NT} \times \mathbb{R}^{MT}$	update equation
C	$\mathbb{R}^{NT} \times \mathbb{R}^{N_{\mathbf{u}}T}$	cost function
λ	$\mathbb{R}^{NT}$	adjoint variable
$D_{\boldsymbol{\rho}}$	$\mathbb{N}$	maximum junction degree on the network
$D_{\mathbf{u}}$	$\mathbb{N}$	maximum number of constraints
β_{2i}^k	$[0, 1]$	off-ramp split ratio
D_{2i-1}^k		flux demand at the boundary of on-ramp $2i-1$
ϵ		barrier penalty coefficient
$\delta_{2(i-1)}$		demand on the link $2(i-1)$
d_{2i-1}^k		demand from on-ramp $2i-1$
σ_{2i}^k		supply on the link $2i$
v_i	$\mathbb{R}_+$	free flow speed for link i
w_i	[0,1]	congestion speed i
$p_{2(i-1)}$	[0,1]	merging parameter