Highway ramp metering via the adjoint method

Author

September 11, 2012

1 Introduction

2 Introduction to using the adjoint method in highway control

2.1 General highway control problem

Consider the following general optimization problem:

$$\begin{array}{ll} \text{minimize}_{u \in \mathcal{U}} & J(x, u) \\ \text{subject to} & H(x, u) = 0 \end{array} \tag{1}$$

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

$$\begin{aligned} J : \mathcal{X} \times \mathcal{U} &\to \mathbb{R} \\ (x, u) &\mapsto J(x, u) \end{aligned} \tag{2}$$

is the objective function and

$$\begin{aligned} H : \mathcal{X} \times \mathcal{U} &\to \mathbb{R}^{n_H} \\ (x, u) &\mapsto H(x, u) \end{aligned} \tag{3}$$

are the system constraints, where $x \in \mathcal{X}$ is the state vector of the system, and $u \in \mathcal{U}$ is a control vector.

In the traffic setting, $J(x, u)$ is typically the total travel time (TTT) or a combination of TTT and the total travel distance (TTD). More on this later. The system constraints $H(x, u)$ include the constraints that determine the dynamics of traffic flow, and the initial conditions. The control constraints can be encoded in the set $\mathcal{U}$ of admissible controls. In this work, the traffic dynamics will be based on a Godunov discretization of the LWR PDE with a triangular fundamental diagram and the control constraints will the ramp metering rate at each on-ramp.

The system constraints that arise from the Godunov discretization are non-linear. One approach to solving this problem is to relax the non-linear constraints and solve the relaxed problem. In certain cases, it can be shown that the resulting relaxation gap is zero. This is not the case in the ramp metering problem that we consider. Therefore, we propose using the adjoint method to compute the gradient and solve this non-linear optimization problem using gradient descent.

2.2 Overview of how to use the adjoint method [?]

The adjoint method is a technique to compute the gradient $\nabla_u J(x, u) = \frac{dJ}{du}$ of the objective function without fully computing $\nabla_u x = \frac{dx}{du}$. The gradient is then used to do gradient-decent based optimization.

2.2.1 A Lagrangian approach

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

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

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

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

In particular, if λ satisfies the adjoint equation

$$\frac{\partial J}{\partial x} + \lambda^T\frac{\partial H}{\partial x} = 0 \tag{4}$$

(TODO: justify existence of a solution λ)

The gradient becomes

$$\nabla_u L(x,u) = \frac{\partial J}{\partial u} + \lambda^T\frac{\partial H}{\partial u} \tag{5}$$

2.2.2 A second derivation

Since the Lagrangian coincides with the objective function on the feasible set, the gradient of the Lagrangian on the feasible set is given by

$$\nabla_u L(x,u,\lambda) = \nabla_u J(x,u) = \frac{\partial J}{\partial u} + \frac{\partial J}{\partial x}\frac{dx}{du}$$

From the system constraints, $\frac{dx}{du}$ satisfies the following condition

$$\begin{aligned}\frac{dH}{du} = 0 &\Leftrightarrow \frac{\partial H}{\partial x}\frac{dx}{du} + \frac{\partial H}{\partial u} = 0 \\ &\Leftrightarrow \frac{\partial H}{\partial x}\frac{dx}{du} = -\frac{\partial H}{\partial u}\end{aligned}$$

therefore for all $\lambda \in \mathbb{R}^{n_H}$

$$\lambda^T\frac{\partial H}{\partial x}\frac{dx}{du} = -\lambda^T\frac{\partial H}{\partial u}$$

in particular, for λ solution to the adjoint equation

$$\frac{\partial J}{\partial x} + \lambda^T\frac{\partial H}{\partial x} = 0$$

we have

$$\frac{\partial J}{\partial x}\frac{dx}{du} = -\lambda^T\frac{\partial H}{\partial x}\frac{dx}{du} = \lambda^T\frac{\partial H}{\partial u}$$

and the gradient becomes simply

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

2.3 Continuous approach and discrete approach

3 A continuous model of traffic on a highway with ramp flow control

3.1 Ramp model

3.2 Junction solver

4 Highway ramp metering using the adjoint method on the discretized system

4.1 Nomenclature

Constants

$\triangle t, \triangle x$	Time, space discretization size
v_i	Free flow speed on cell i
w_i	Congestion wave speed on cell i
ρ_i^{jam}	Jam density on cell i
F_i	Max flow leaving mainline of cell i
P_i	Priority factor for cell i

Inputs

ρ_i^0	Initial density on cell i
l_i^0	Initial queue length for onramp entering cell i
$D_i\left(k\right)$	Input flow on cell i, time step k
$\beta_i\left(k\right)$	Offramp split ratio on cell i, time step k

Variables

$f_i^{\text{in}}\left(k\right)$	Flow into mainline on cell i, time step k
$f_i^{\text{out}}\left(k\right)$	Flow out of mainline on cell i, time step k
$\rho_i\left(k\right)$	Density on cell i, time step k
$r_i\left(k\right)$	Flow from onramp entering cell i, time step k
$l_i\left(k\right)$	Queue length for onramp entering cell i, time step k
$u_i\left(k\right)$	Max flow from onramp entering cell i, time step k
$\sigma_i\left(k\right)$	Supply on cell i, time step k
$\delta_i\left(k\right)$	Demand on cell i, time step k
$d_i\left(k\right)$	Onramp demand entering cell i, time step k

4.2 System description

We consider a discretization of the continuous system described in Section 3, using the Godunov scheme. A diagram describing the system is given in fig. 1. Discrete time is indexed by $k \in \{0, \dots, T\}$. The mainline is divided into N cells, indexed by $i \in \{1, \dots, N\}$. The density on cell i at time step k is given by $\rho_i\left(k\right)$. The incoming (respectively outgoing) flux to cell i at time step k is given by $f_i^{\text{in}}\left(k\right)$ (respectively $f_i^{\text{out}}\left(k\right)$). We add a ghost cell at the entrance of the network, cell $i = 0$, to impose the boundary flow, or flow demand, given at time step k by $D_0\left(k\right)$. Each cell $i \in \{1, \dots, N-1\}$ is followed by a two-two junction (referred to as junction i), that connects the mainline to an on-ramp and an off-ramp (referred to as on-ramp i and off-ramp i). The flow demand from the off-ramp is determined by the control $u_i\left(k\right)$ (that is the maximum flux out of the ramp) and the car count $l_i\left(k\right)$ on the ramp. The junction flows are determined in the same way as in the continuous system. On-ramp i is subject to flux input (or flux demand) given by the sequence $\left(D_i\left(k\right)\right)_k$.

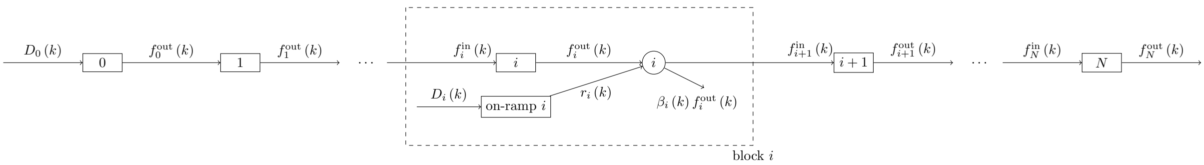


Figure 1: Flow variables and boundary flows in the system.

4.2.1 Choosing the correct space and time discretization

TODO: Add the CFL conditions and discuss how to select the proper cell size. Smaller cells will violate the CFL condition and larger cells will cause numerical inaccuracies when vehicles need more that one time step to pass through the cell in free flow.

4.3 System equations

Let x denote the state vector of the system and let u denote the vector of control variables, $u = (u(0), \dots, u(T))$, where at time k, $u(k)$ is given by

$$u(k) = (u_1(k), \dots, u_{N-1}(k))$$

The system equations are written formally in the form $H(x, u) = 0$. The discretized system can be described using eight types of constraints, given by $H^c_{k,i} = 0$ for $c \in \{1, \dots, 8\}$, where we index each equality constraint by time index k, and cell index i. We now give the system equations.

The mass conservation equations are given by

$$H^1_{k,i}: \quad \rho_i(k) = \rho_i(k-1) + \frac{\triangle t}{\triangle x}\left(f^{\text{in}}_i(k-1) - f^{\text{out}}_i(k-1)\right) \quad \forall i \in \{1, \dots, N-1\}, k \in \{1, \dots, T\} \tag{H1a}$$

$$H^1_{k,0}: \quad \rho_0(k) = \rho_0(k-1) + \frac{\triangle t}{\triangle x}\left(D_0(k-1) - f^{\text{out}}_0(k-1)\right) \quad \forall k \in \{1, \dots, T\} \tag{H1b}$$

$$H^1_{k,N}: \quad \rho_N(k) = \rho_N(k-1) + \frac{\triangle t}{\triangle x}\left(f^{\text{in}}_N(k-1) - \delta_N(k-1)\right) \quad \forall k \in \{1, \dots, T\} \tag{H1c}$$

and initial condition

$$H^0_{1,i}: \rho_i(0) = \rho^0_i \quad \forall i \in \{0, \dots, N\} \tag{I1}$$

The car count on ramp i is given by

$$H^2_{k,i}: \quad l_i(k) = l_i(k-1) + \triangle t\left(D_i(k-1) - r_i(k-1)\right) \quad \forall i \in \{1, \dots, N-1\}, k \in \{1, \dots, T\} \tag{H2}$$

$$H^2_{0,i}: \quad l_i(0) = l^0_i \quad \forall i \in \{1, \dots, N-1\} \tag{I2}$$

At junctions, the flows are given by the solver described in section 3.2. The flows can be determined by first computing the demand function (equation $H^3_{k,i}$) and the supply function (equation $H^4_{k,i}$) for the mainline, the demand function for the ramp (equation $H^5_{k,i}$), then the total flow through the junction (equation $H^6_{k,i}$).

$$H^3_{k,i}: \quad \delta_i(k) = \min\left(F_i, v_i \rho_i(k)\right) \quad \forall i \in \{0, \dots, N\}, k \in \{0, \dots, T-1\} \tag{H3}$$

$$H^4_{k,i}: \quad \sigma_i(k) = \min\left(F_i, w_i\left(\rho^{\text{jam}}_i - \rho_i(k)\right)\right) \quad \forall i \in \{1, \dots, N\}, k \in \{0, \dots, T-1\} \tag{H4}$$

$$H^5_{k,i}: \quad d_i(k) = \min\left(l_i(k), u_i(k)\right) \quad \forall i \in \{1, \dots, N-1\}, k \in \{0, \dots, T-1\} \tag{H5}$$

$$H^6_{k,i}: \quad f^{\text{in}}_i(k) = \min\left(\delta_{i-1}(k)\left(1 - \beta_{i-1}(k)\right) + d_{i-1}(k), \sigma_i(k)\right) \quad \forall i \in \{2, \dots, N\}, k \in \{0, \dots, T-1\} \tag{H6a}$$

$$H^6_{k,1}: \quad f^{\text{in}}_1(k) = \min\left(\delta_0(k), \sigma_1(k)\right) \quad \forall k \in \{0, \dots, T-1\} \tag{H6b}$$

When there is an actual offramp at the junction (i.e. $\beta_i(k) > 0$), the flow is uniquely determined by flow maximization across the junction (see fig. 2b). When the split ratio $\beta_i(k) = 0$ (equivalently, when there is no off-ramp), the solution of the junction problem may not be unique. In order to guarantee uniqueness of the solution, we use a fixed[1] priority vector [?] given by P_i for junction i. The unique

[1]comment about fixed priority and proportional priority

solution is given for all cases (see Fig. 6)

$$H^7_{k,i} : f^{\text{out}}_i(k) = \begin{cases} f^{\text{in}}_{i+1}(k) / (1-\beta_i(k)) & \text{if } (R1_{k,i}) : \beta_i(k) > 0 \text{ and } f^{\text{in}}_{i+1}(k) < (1-\beta_i(k))\,\delta_i(k) \\ \delta_i(k) & \text{if } (R2_{k,i}) : \beta_i(k) > 0 \text{ and } f^{\text{in}}_{i+1}(k) \geq (1-\beta_i(k))\,\delta_i(k) \\ \delta_i(k) & \text{if } (R3_{k,i}) : \beta_i(k) = 0 \text{ and } \frac{P_i}{1-P_i} > \frac{\delta_i(k)}{f^{\text{in}}_{i+1}(k)-\delta_i(k)} \\ f^{\text{in}}_{i+1}(k) - d_i(k) & \text{if } (R4_{k,i}) : \beta_i(k) = 0 \text{ and } \frac{P_i}{1-P_i} < \frac{f^{\text{in}}_{i+1}(k)-d_i(k)}{d_i(k)} \\ P_i f^{\text{in}}_{i+1}(k) & \text{otherwise } (R5_{k,i}) \end{cases}$$

$$\forall i \in \{1,\ldots,N-1\}, k \in \{0,\ldots,T-1\} \quad \text{(H7a)}$$

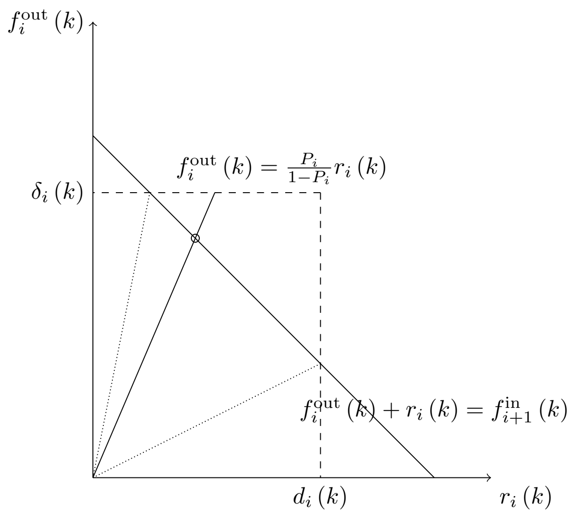
$$H^7_{k,0} : \quad f^{\text{out}}_0(k) = f^{\text{in}}_1(k) \quad \forall k \in \{0,\ldots,T-1\} \quad \text{(H7b)}$$

where Equation (H7b) is a special case for the flow out at the source dummy cell. Here we use $R1_{k,i},\ldots,R5_{k,i}$ to denote the sets of state vectors that satisfy the corresponding condition. They form a partition of X. This will be a useful notation in the adjoint system expression, where we will use the indicator function $1_{R1_{k,i}}$, with implicit argument x, to denote the function

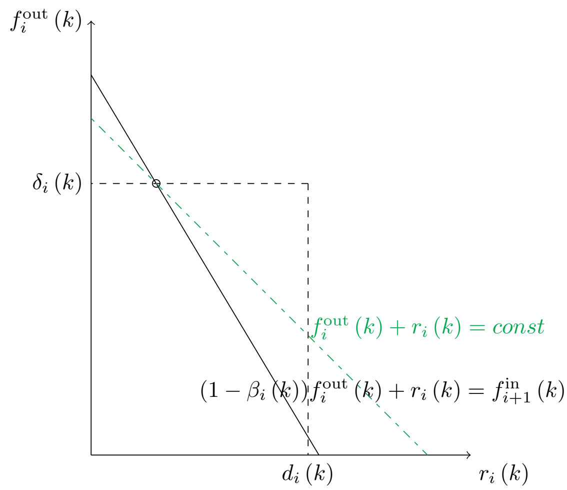
$$\begin{aligned} 1_{R1_{k,i}} &: X \to \{0,1\} \\ x &\mapsto \begin{cases} 1 & \text{if } x \in R1_{k,i} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Finally, the ramp flow is simply given by the conservation of flows:

$$H^8_{k,i} : \quad r_i(k) = f^{\text{in}}_{i+1}(k) - f^{\text{out}}_i(k)(1-\beta_i(k)) \quad \forall i \in \{1,\ldots,N-1\}, k \in \{0,\ldots,T-1\} \quad \text{(H8)}$$



(a) $\beta_i(k) = 0$



(b) $\beta_i(k) \neq 0$

Figure 2: Junction flows

4.4 Ordering of state vector and constraints

In order create a lower triangular $\frac{\partial H}{\partial x}$ matrix, we must have the proper ordering of the variables in the $\mathbf{x}$ vector. This process is similar to understanding the nature of the dependencies in the forward simulation. Given the boundary and initial conditions, the discretized system can be exactly solved by solving the constraints in a particular order. From this consideration, we create a "topological ordering" of the variables, or the order in which the variables must be solved from the particular constraints.

$$
\begin{aligned}
&\rho_i(0), l_j(0) && i \in \{0, \ldots, N\}, j \in \{1, \ldots, N-1\} \\
&\delta_i(0), \sigma_j(0), d_k(0) && i \in \{0, \ldots, N\}, j \in \{1, \ldots, N\}, k \in \{1, \ldots, N-1\} \\
&f_i^{\text{in}}(0) && i \in \{1, \ldots, N\} \\
&f_i^{\text{out}}(0) && i \in \{0, \ldots, N\} \\
&r_i(0) && i \in \{1, \ldots, N-1\} \\
&\rho_i(1), l_j(1) && i \in \{0, \ldots, N\}, j \in \{1, \ldots, N-1\} \\
&\ldots && \\
&r_i(k) && i \in \{1, \ldots, N-1\} \\
&\rho_i(k+1), l_j(k+1) && i \in \{0, \ldots, N\}, j \in \{1, \ldots, N-1\} \\
&\ldots && \\
&r_i(T) && i \in \{1, \ldots, N-1\}
\end{aligned} \tag{6}
$$

The ordering has three nested loops: first over time index, then over "variable type", then over cell index. Each row in the set of equations only depends upon variables in above rows, or more precisely, they will only depend on variables in rows above *and* on the same time step k or $k-1$.

The constraints in the system equations section were presented in a particular way, in order to have every variable in the system appear alone on the LHS of exactly one equation, and have the RHS of each equation only be a function of variables with a "lower" topological order. Therefore, it is clear that if we order the $\mathbf{x}$ variables according to Equation 6, and order the constraints according to when the LHS variable of the constraint appears in $\mathbf{x}$, then the $\frac{\partial H}{\partial x}$ matrix will have a lower-triangular structure. Additionally, there will be all 1's on the diagonal.

When the adjoint system is considered, the transpose of the $\frac{\partial H}{\partial x}$ matrix is taken, thus creating an upper-triangular structure. Such a system can be solved backwards in time, as opposed to the original system being solved forward in time from the initial conditions.

Note that some of the constraints contain special cases for initial conditions and boundary conditions, but it should be clear to make the 1-to-1 mapping with the variables based on the time and space indices being considered.

4.5 Forward simulation

In the forward simulation step, we are interested in finding the state at time $k+1$ given the state at time k and the boundary conditions at time $k+1$. This can be done trivially by iteratively solving the equations H1 through H10, since the order of the equations corresponds to the ordering of the dependencies within the equations. We start the the initial conditions at time T=0 and solve the system of equations one time step at a time for all cells.

4.6 Objective function

TODO: generalize this to include total travel distance.
TODO: cars have to be flushed out of the network at the end of the T time steps. Discuss how this is done.

The objective function is the total travel time, given by

$$
J(x, u) = \sum_{k=0}^{T} \left(\sum_{i=0}^{N} \rho_i(k) + \sum_{i=1}^{N-1} l_i(k) \right)
$$

4.7 Adjoint system

The adjoint system is given by $\frac{\partial J}{\partial x}^T + \frac{\partial H}{\partial x}^T \lambda = 0$. We observe that in our case, the forward system is affine in x around any given point $(x^{(0)}, u^{(0)})$, and can be written in the form

$$
H(x, u^{(0)}) = A(x^{(0)}, u^{(0)})x + b(x^{(0)}, u^{(0)}) \tag{7}
$$

where $A(x^{(0)},u^{(0)})$ is a matrix and $b(x^{(0)},u^{(0)})$ is a vector that are entirely determined by the state $(x^{(0)},u^{(0)})$, and can be computed during the forward simulation step. The adjoint system is then given simply by $A(x^{(0)},u^{(0)})^T\lambda+\frac{\partial J}{\partial x}=0$. Since $A(x^{(0)},u^{(0)})$ is lower-triangular by construction (see Section 4.4), the adjoint system is upper triangular and can be solved backwards in time. Next, we explicitly give the adjoint system in equation form (not in matrix form due to the large size of the matrix)

4.7.1 Adjoint equations

The adjoint equations are given by $\frac{\partial J}{\partial x}+\lambda^T\frac{\partial H}{\partial x}=0$. This can be rewritten formally as

$$\frac{\partial J}{\partial x}+\sum_{k,h,i}\lambda_{k,i}^h\frac{\partial H_{k,i}^h}{\partial x}=0$$

where $h\in\{1,\dots,8\}$ indexes the type of constraint, i and k are in the appropriate ranges, i.e. wherever $H_{k,i}^h$ is defined. Here we defined one adjoint variable $\lambda_{k,i}^h$ for each constraint $H_{k,i}^h$.

We give the equation corresponding to each variable x. In the formulation below, we distinguish final conditions that correspond to the final time step T, namely equations (F), from the rest of the equations, denoted by (G).

Taking the partial derivative with respect to the density variable $\rho_i(k)$, we obtain the following equations

$$G_{T,i}^1:\quad 0=1-\lambda_{T,i}^1 \qquad \forall i\in\{0,\dots,N\},k=T \tag{F1}$$

$$G_{k,0}^1:\quad 0=1-\lambda_{k,0}^1+\lambda_{k+1,0}^1+\lambda_{k,0}^3v_0 1_{\{v_0\rho_0(k)<F_0\}} \qquad i=0,\forall k\in\{0,T-1\} \tag{G1a}$$

$$G_{k,i}^1:0=1-\lambda_{k,i}^1+\lambda_{k+1,i}^1+\lambda_{k,i}^3v_i1_{\{v_i\rho_i(k)<F_i\}}-\lambda_{k,i}^4w_i1_{\left\{w_i(\rho_i^{\text{jam}}-\rho_i(k))<F_i\right\}} \qquad \forall i\in\{1,\dots,N\},\forall k\in\{0,T-1\} \tag{G1b}$$

The equations corresponding to the partial derivative with respect to the ramp queue $l_i(k)$ are

$$G_{T,i}^2:\quad 0=1-\lambda_{T,i}^2 \qquad \forall i\in\{1,\dots,N-1\},k=T \tag{F2}$$

$$G_{k,i}^2:\quad 0=1-\lambda_{k,i}^2+\lambda_{k+1,i}^2+\lambda_{k,i}^5 1_{\{l_i(k)<u_i(k)\}} \qquad \forall i\in\{1,\dots,N-1\},\forall k\in\{0,T-1\} \tag{G2}$$

The equations corresponding to the partial derivative with respect to the demand $\delta_i(k)$ are

$$G_{k,i}^3:0=-\lambda_{k,i}^3+(1-\beta_i(k))\lambda_{k,i+1}^6 1_{\{(1-\beta_i(k))\delta_i(k)+d_i(k)<\sigma_{i+1}(k)\}}+\lambda_{k,i}^7 1_{R2_{k,i}\cup R3_{k,i}} \qquad i\in\{1,\dots,N-1\},\forall k\in\{0,T-1\} \tag{G3a}$$

$$G_{k,0}^3:\quad 0=-\lambda_{k,i}^3+\lambda_{k,1}^6 \qquad i=0,\forall k\in\{0,T-1\} \tag{G3b}$$

$$G_{k,N}^3:\quad 0=-\lambda_{k+1,N}^1-\lambda_{k,N}^3 \qquad i=N,\forall k\in\{0,T-1\} \tag{G3c}$$

The equations corresponding to the partial derivative with respect to the supply $\sigma_i(k)$ are

$$G_{k,i}^4:\quad 0=-\lambda_{k,i}^4+\lambda_{k,i}^6 1_{\{\sigma_i(k)<(1-\beta_{i-1}(k))\delta_{i-1}(k)+d_{i-1}(k)\}} \qquad i\in\{2,\dots,N\},\forall k\in\{0,T-1\} \tag{G4a}$$

$$G_{k,1}^4:\quad 0=-\lambda_{k,1}^4+\lambda_{k,1}^6 1_{\{\sigma_1(k)<\delta_0(k)\}} \qquad i=1,\forall k\in\{0,T-1\} \tag{G4b}$$

The equations corresponding to the partial derivative with respect to the ramp demand $d_i(k)$ are

$$G_{k,i}^5:0=-\lambda_{k,N}^5+\lambda_{k,i+1}^6 1_{\{(1-\beta_i(k))\delta_i(k)+d_i(k)<\sigma_{i+1}(k)\}}-\lambda_{k,i}^7 1_{R4_{k,i}} \qquad i\in\{1,\dots,N-1\},\forall k\in\{0,T-1\} \tag{G5}$$

The equations corresponding to the partial derivative with respect to the in-flux $f_i^{\text{in}}(k)$ are

$$G_{k,i}^6:0=\frac{\triangle t}{\triangle x}\lambda_{k+1,N}^1-\lambda_{k,i}^6+\lambda_{k,i-1}^7\left[\frac{1}{1-\beta_{i-1}(k)}1_{R1_{k,i-1}}+1_{R4_{k,i-1}}+P_i1_{R5_{k,i-1}}\right]+\lambda_{k,i-1}^8 \qquad i\in\{2,\dots,N\},\forall k\in\{0,T-1\} \tag{G6a}$$

$$G^6_{k,1}: \qquad 0 = \frac{\triangle t}{\triangle x}\lambda^1_{k+1,1} - \lambda^6_{k,1} + \lambda^7_{k,0} \qquad i = 1, \forall k \in \{0, T-1\} \tag{G6b}$$

The equations corresponding to the partial derivative with respect to the out-flux $f_i^{\text{out}}(k)$ are

$$G^7_{k,i}: \qquad 0 = -\frac{\triangle t}{\triangle x}\lambda^1_{k+1,i} - \lambda^7_{k,i} - \lambda^8_{k,i}(1-\beta_i(k)) \qquad i \in \{1,\ldots,N-1\}, \forall k \in \{0, T-1\} \tag{G7a}$$

$$G^7_{k,0}: \qquad 0 = -\frac{\triangle t}{\triangle x}\lambda^1_{k+1,0} - \lambda^7_{k,0} \qquad i = 0, \forall k \in \{0, T-1\} \tag{G7b}$$

The equations corresponding to the partial derivative with respect to the on-ramp flux $r_i(k)$ are

$$G^8_{k,i}: \qquad 0 = -\triangle t \lambda^2_{k+1,i} + \lambda^8_{k,i} \qquad i \in \{1,\ldots,N-1\}, \forall k \in \{0, T-1\} \tag{G8}$$

As mentioned above, the adjoint linear system is upper triangular, and can be solved backwards in time, starting from the last time step T (i.e. solve equations $G^1_{T,i}$ and $G^2_{T,i}$). At each time step, the system is solved for decreasing h (starting from $h = 8$) then for decreasing i.

4.8 Computing the gradient

After solving for λ (adjoint system), we can compute the gradient, given formally by

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

Since the objective function J does not depend on the control parameter u, we have

$$\frac{\partial J}{\partial u} = 0$$

The system constraints H have only one equation H5 that depends on the control u. Therefore, the derivative of all other constraints with respect to the control u is zero.

$$\frac{\partial H^5_{k,i}}{\partial u_i(k)} = -1_{\{l_i(k) > u_i(k)\}} \tag{8}$$

5 Triangular system formulation for forward and backward problem

In this section, we detail the triangular nature of our forward-simulation system by showing a dependency diagram of the variables with respect to time, variable type, and cell type. The dependency chain is given in Figure 3. Initial densities and ramp queues have no dependencies, since they are input into the system. From these values, all variables related to demand (δ, σ, d) for time step 0 can be determined with no dependency on the cell type. All variables for time step 0 can be determined by following the dependency chart. Once time step 0 is known, then all of time step 1 can be determined. This process can be repeated for all time steps, at which point all variables are known.

Since the system is piecewise-affine (see [?]), the next timestep x_k can be calculated by $A_m x_{k-1} + b_m$, where m is the "mode" of the system at time step $k-1$. The mode, and in turn the A, b matrices, is only determined after a "forward simulation" step is done on the previous time step. This process can be repeated for every time step, and we thus know the Jacobian $\frac{\partial H}{\partial x}$ matrix to be constant, if the entire state of the system is known. Furthermore, since all variables can be determined from previously solved variables (Figure 3), we know the structure of this matrix to be lower-triangular if the state vector is topologically ordered.

While this may appear to be circuitous (to use knowledge of the entire state of the system to determine how the system will evolve), the adjoint system assumes complete knowledge of the state. Therefore, without any forward-simulation step, the entire $\frac{\partial H}{\partial x}$ matrix can be solved (and is constant). The adjoint variables are determined as the solution of Equation 4, which is a system of linear equation for our case. Furthermore, the transpose of $\frac{\partial H}{\partial x}$ is an upper triangular matrix, and can therefore be efficiently solved using back-substitution. The sparsity of the $\frac{\partial H}{\partial x}$ matrix allows us to use even more efficient solution methods, which explicitly use the dependency diagram in Figure 3.

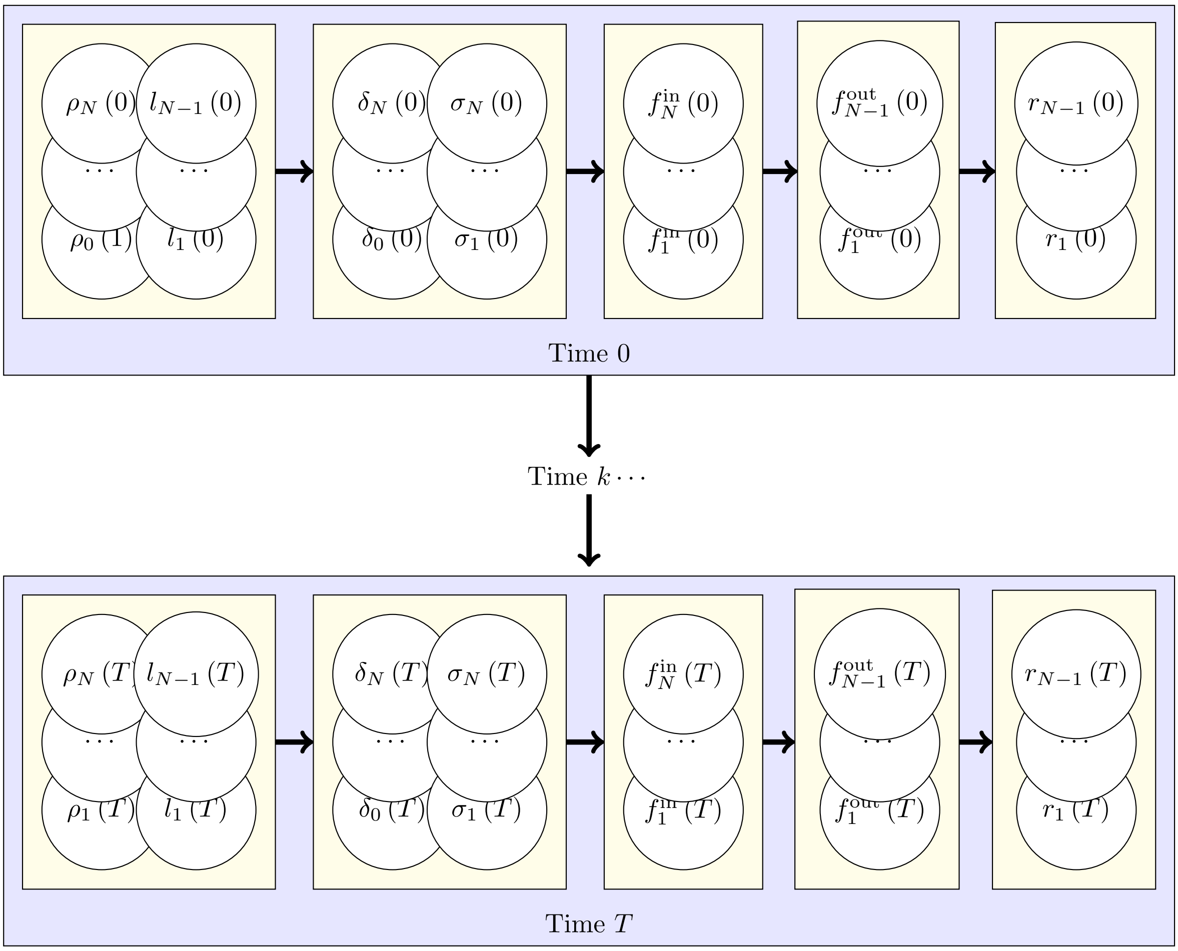


Figure 3: Dependency diagram of the variables in the system.

6 Gradient descent methods for the ramp metering problem

We use gradient descent algorithm (to be determined). We are given an initial density, ρ_i^0 and initial queue lengths l_i^0.

- start from an initial feasible control $u^{(0)} \in \mathcal{U}$ (e.g. given by setting the ramp control at the maximal rate $u_i(k) = u_i^{\max}$ $\forall i \in \{1, \dots, N-1\}$ and $\forall k \in \{0, \dots, T\}$).
- Until convergence: we use superscript (p) for variables at step p
  - forward simulate to compute the corresponding state $x^{(p)}$ induced by control $u^{(p)}$, i.e. solve $H(x^{(p)}, u^{(p)}) = 0$
  - compute the adjoint $\lambda^{(p)}(k), k \in \{0, \dots, T\}$ by solving the adjoint equations
  - use the adjoint to compute the gradient $\nabla_u J(x^{(p)}, u^{(p)})$
  - update the control $u^{(p+1)} = u^{(p)} - t^{(p)} \nabla_u J(x^{(p)}, u^{(p)})$. Here $t^{(p)}$ is the step size and can be determined using line search.

6.1 Avoiding local minima when the control parameter $u_i(k) > l_i(k)$

From equation(5) we have,

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

In section 4.8 we showed that,

$$\frac{\partial J}{\partial u} = 0$$

and $\frac{\partial H}{\partial u}$ only having non-zero entries for constraint H^5

$$\frac{\partial H^5_{k,i}}{\partial u_i(k)} = 1_{\{u_i(k)<l_i(k)\}}$$

Thus, the system can get stuck at a local minimum when $u_i(k) > l_i(k)$. To avoid this problem we add a penalty term to the objective function that forces the solution to a point where $u_i(k) \leq l_i(k)$. The modified objective function is as follows.

$$J(x,u) = \sum_{k=0}^{T} \left(\sum_{i=0}^{N} \rho_i\left(k\right) + \sum_{i=1}^{N-1} l_i\left(k\right) \right) + R \sum_{k=0}^{T-1} \sum_{i=1}^{N-1} \max\left(u_i(k) - l_i(k), 0\right)$$

where R is a constant that regulates the magnitude of the penalty. Now,

$$\frac{\partial J}{\partial u_i(k)} = R\left(u_i(k) - l_i(k)\right) \cdot 1_{\{u_i(k)\geq l_i(k)\}}$$

7 Implementation details

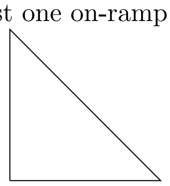
7.1 Variable dimensions

The following table describes the variables used in the implementation and their dimensions.

Variable	Description	Dimension
T	total time steps	1
N	total cells	1
V	total number of variables	1
C	total number of constraints	1
J	objective function	1
H	system of constraints	$1 \times (T \cdot C \cdot N)$
λ	adjoint variables	$1 \times (T \cdot C \cdot N)$
u	control variables	$1 \times (T \cdot N - 1)$

7.2 Example 1: one on-ramp and no off-ramps

This section shows an example of how ramp metering can help in a very simple highway segment with just one on-ramp and no off ramps.



8 Adjoint framework for general junction problem

We wish to study a number of junction problems with the adjoint framework to comparison purposes. Therefore, we wish to extract the shared aspects of the adjoint approach to ramp metering. The initial conditions, boundary conditions, mass balance equations across junctions and ramps, and control parameter are all the same. What differs is the solution of the flow that is sent from the one cell and one ramp to the next cell. Therefore, we use a general description of the flow sent and received from a cell or ramp based on data from the previous time step.

The density introduced into cell i at time k is:

$$\rho^{\text{in}}_{i,k}\left(\rho_i\left(k-1\right), \rho_{i-1}\left(k-1\right), l_{i-1}\left(k-1\right), u_{i-1}\left(k-1\right)\right)$$

The density sent out of the cell is:

$$\rho^{\text{out}}_{i,k}\left(\rho_i\left(k-1\right),\rho_{i+1}\left(k-1\right),l_i\left(k-1\right),u_i\left(k-1\right)\right)$$

Similarly, the flow leaving the ramp i is:

$$l^{\text{out}}_{i,k}\left(\rho_i\left(k-1\right),\rho_{i+1}\left(k-1\right),l_i\left(k-1\right),u_i\left(k-1\right)\right)$$

For the Garavello, Piccoli model [?], the above functions are fully detailed in the present report. It is not efficient to think of these functions as independent modules, as they share many intermediate values to determine their values. The formulation is useful from a conceptual perspective, as it allows us to explicitly isolate the core equations, initial conditions, and the boundary conditions for both the forward and adjoint systems for any junction model.

From an implementation standpoint, this method does include intermediate variables explicitly in the state vector, which reduces the size of our linear system that must be solved for the adjoint variables. The trade-off is that the construction of the $\frac{\partial H}{\partial x}$ matrix will take longer. If both systems are implemented as sparse matrices with a triangular form, and the matrix solver is perfectly efficient, then the two approaches should be practically identical from a computational stand-point.

8.1 Forward system

System for density variables:

$$\begin{aligned}H_{k,i,\rho} =& \rho_i\left(k\right)-\rho_i\left(k-1\right)\\&+\rho^{\text{in}}_{i,k}\left(\rho_i\left(k-1\right),\rho_{i-1}\left(k-1\right),l_{i-1}\left(k-1\right),u_{i-1}\left(k-1\right)\right)\\&-\rho^{\text{out}}_{i,k}\left(\rho_i\left(k-1\right),\rho_{i+1}\left(k-1\right),l_i\left(k-1\right),u_i\left(k-1\right)\right)\qquad \forall i\in\{0,\ldots,N\},\forall k\in\{1,\ldots,T\}\end{aligned}$$

System for ramp queue variables:

$$\begin{aligned}H_{k,i,l} =& l_i\left(k\right)-l_i\left(k-1\right)+D_i\left(k-1\right)\\&-l^{\text{out}}_{i,k}\left(\rho_i\left(k-1\right),\rho_{i+1}\left(k-1\right),l_i\left(k-1\right),u_i\left(k-1\right)\right)\qquad \forall i\in\{0,\ldots,N\},\forall k\in\{1,\ldots,T\}\end{aligned}$$

Initial conditions:

$$\begin{aligned}H_{0,i,\rho} =& \rho_i\left(0\right)-\rho^0_i && \forall i\in\{0,\ldots,N\}\\H_{0,i,l} =& l_i\left(0\right)-l^0_i && \forall i\in\{1,\ldots,N-1\}\end{aligned}$$

Boundary conditions for densities:

$$H_{k,i,\rho} = \rho_i\left(k\right) \qquad \forall i\in\{-1,N+1\},\forall k\in\{0,\ldots,T\}$$

Boundary conditions for ramp queues:

$$H_{k,-1,l} = l_{-1}\left(k\right) \qquad \forall k\in\{0,\ldots,T\}$$

8.2 Objective

$$\frac{\partial J}{\partial x_{k,i,k}} = 1\forall i\in\{-1,\ldots,N+1\},\forall k\in\{0,\ldots,T\},\forall p\in\{\rho,l\}$$

8.3 Adjoint system

The adjoint system for density constraints:

$$\begin{aligned}
\lambda_{k,i,\rho} + \left(-1 - \frac{\partial \rho^{\text{out}}_{i,k+1}}{\partial \rho_{i,k}} - \frac{\partial \rho^{\text{in}}_{i,k+1}}{\partial \rho_{i,k}}\right)\lambda_{k+1,i,\rho} \\
+ \left(\frac{\partial \rho^{\text{out}}_{i-1,k+1}}{\partial \rho_{i,k}}\right)\lambda_{k+1,i-1,\rho} + \left(-\frac{\partial \rho^{\text{in}}_{i+1,k+1}}{\partial \rho_{i,k}}\right)\lambda_{k+1,i+1,\rho} \\
+ \left(\frac{\partial l^{\text{out}}_{i,k+1}}{\partial \rho_{i,k}}\right)\lambda_{k+1,i,l} + \left(\frac{\partial l^{\text{out}}_{i-1,k+1}}{\partial \rho_{i,k}}\right)\lambda_{k+1,i-1,l} = 1 \\
\forall i \in \{0,\ldots,N\}, \forall k \in \{0,\ldots,T-1\}
\end{aligned}$$

The adjoint system for ramp queue constraints:

$$\begin{aligned}
\lambda_{k,i,l} + \left(-1 - \frac{\partial l^{\text{out}}_{i,k+1}}{\partial l_{i,k}}\right)\lambda_{k+1,i,l} + \left(\frac{\partial \rho^{\text{out}}_{i,k+1}}{\partial l_{i,k}}\right)\lambda_{k+1,i,\rho} + \left(-\frac{\partial \rho^{in}_{i+1,k+1}}{\partial l_{i,k}}\right)\lambda_{k+1,i+1,\rho} = 1 \\
\forall i \in \{0,\ldots,N\}, \forall k \in \{0,\ldots,T-1\}
\end{aligned}$$

Initial conditions:

$$\begin{aligned}
\lambda_{T,i,\rho} &= 1 \forall i \in \{-1,\ldots,N+1\} \\
\lambda_{T,i,l} &= 1 \forall i \in \{-1,\ldots,N\}
\end{aligned}$$

Boundary conditions for densities:

$$\begin{aligned}
\lambda_{k,-1,\rho} + \left(-\frac{\partial \rho^{\text{in}}_{0,k+1}}{\partial \rho_{-1,k}}\right)\lambda_{k+1,0,\rho} = 1 \forall k \in \{0,\ldots,T-1\} \\
\lambda_{k,N+1,\rho} + \left(\frac{\partial \rho^{\text{out}}_{N,k+1}}{\partial \rho_{N+1,k}}\right)\lambda_{k+1,N,\rho} + \left(\frac{\partial l^{\text{out}}_{N,k+1}}{\partial \rho_{N+1,k}}\right)\lambda_{k+1,N,l} = 1 \\
\forall k \in \{0,\ldots,T-1\}
\end{aligned}$$

Boundary conditions for ramp queues:

$$\begin{aligned}
\lambda_{k,-1,l} + \left(-\frac{\partial \rho^{\text{in}}_{0,k+1}}{\partial l_{-1,k}}\right)\lambda_{k+1,0,\rho} = 1 \\
\forall k \in \{0,\ldots,T-1\}
\end{aligned}$$

9 Continuous model with onramp buffer

9.1 Weak boundary conditions and ramp flux demands

One condition we wish our continuous model to possess is mass-conservation on the ramp demands. In other words, for a ramp at cell i, we want the demand at the ramp to apply to the system in a *strong* sense. For horizontal queues with density having a spacial and temporal dependency, backwards-moving shocks could cause a density boundary condition to not apply, thus *weak* boundary conditions are considered for these models. Then, the flux at such a boundary condition would be determined from the solution of a junction problem at the weak boundary condition. Let $\bar{D}_i(t)$ be the boundary flow specification, and let $D_i(t)$ be the actual boundary flux. Then our condition amounts to:

$$\int_{t=0}^{T} \bar{D}_i(t)\, dt = \int_{t=0}^{T} D_i(t)\, dt \tag{9}$$

For the horizontal queueing model, we have:

$$D_i(t) = f_i(\hat{\rho}\left(f_i^{-1,\mathrm{ff}}\left(\bar{D}_i(t)\right), \rho_i(a_i, t)\right) \leq \bar{D}_i(t)$$

where $f(\cdot)$ is the flux function mapping density to flux, $f_i^{-1,\mathrm{ff}}\left(\bar{D}_i(t)\right)$ is the inverse mapping of flux to the corresponding free-flow density, $\rho_i(a_i, t)$ is the density at the beginning of cell i at time t, and $\hat{\rho}$ is the solution of the Riemann problem at the boundary. The solution of $\hat{\rho}$ admits fluxes that can be strictly less than $\bar{D}_i(t)$, but never greater than $\bar{D}_i(t)$. If this limiting flux condition occurs over a time period $[t_1, t_2]$, then Equation (9) becomes:

$$\begin{aligned}
\int_{t=\bar{t_1}}^{\bar{t_2}} D_i(t)\, dt &\leq \textstyle\int_{t=\bar{t_1}}^{\bar{t_2}} \bar{D}_i(t) \quad \forall \bar{t_1} \leq \bar{t_2}, && (10)\\
\int_{t=t_1}^{t_2} D_i(t)\, dt &< \textstyle\int_{t=t_1}^{t_2} \bar{D}_i(t)\, dt \quad \Longrightarrow && (11)\\
\int_{t=0}^{T} D_i(t)\, dt &< \textstyle\int_{t=0}^{T} \bar{D}_i(t)\, dt && (12)
\end{aligned}$$

This shows that having a horizontal queue model for the onramp will not guarantee a strong application of flux boundary conditions.

9.2 Buffer model for ramps

To overcome this shortcoming on ramps, we propose to use a *buffer* ODE for the ramp model, and have the ODE interact with the larger PDE system via the junction model. For a ramp at cell i, let the ODE of queue length $l_i(t)$ be:

$$\frac{dl_i(t)}{dt} = \bar{D}_i(t) - \gamma_i^{\mathrm{r}}(t) \tag{13}$$

where γ denotes the outgoing flux from the ramp. It is obvious from this model that the inflow into the ramp will be equal to the condition specified in Equation (9).

For the junction model, we solve for the fluxes across junctions by maximizing the flow across a junction given the demands of the incoming links and the supplies of the outgoing links. The mainline links determine their demands and supplies as done in [?]. The onramp demand is given by the following equation:

$$d_i(t) = \begin{cases} \gamma_i^{\mathrm{r,max}} & l_i(t) > 0 \\ \bar{D}_i(t) & \text{otherwise} \end{cases}$$

which permits the physically allowable flux out of the ramp when there is a queue, and the current input flux of the buffer otherwise. Finally, the offramp should be considered to have infinite supply.

The linear program will solve for three relevant fluxes: $\gamma_i(t), \gamma_i^{\mathrm{r}}(t)$, and $\gamma_{i+1}(t)$ which represent the flux from the current cell into the junction, the flux out of the ramp, and the flux out of the junction into the downstream cell respectively. The boundary densities, $\hat{\rho}$, for the mainlines would be determined as in [?]. The following system would constitute a full solution of the junction problem, and would also determine the ramp flux $\gamma_i^{\mathrm{r}}(t)$ necessary to virtualSelfthe ramp virtualSelfSimilarE in Equation (13).

9.3 Unique and self-similar solutions for the continuous model

9.3.1 Virtual Junction Problem Approach

Reference Figure 4 for the notation used in the current junction problem. Let γ, ρ be the vector notation for fluxes and densities on the links respectively. Let the $\hat{()}$ notation denote variables pertaining to a

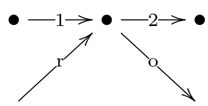


Figure 4: Illustration of junction under consideration

Riemann solution, and let the ($'$) notation denote variables pertaining to the *virtual* junction problem. Let $\bar{\gamma}_i$ denote the maximum flux allowed from or into a link i, depending on the context. For instance, $\hat{\bar{\gamma}}_2'$ would be the maximum flux allowed into link 2 for the virtual junction problem, considering the state given from the Riemann solution of the initial data ρ.

Define the mapping from densities and ramp queues to maximum flux as $\Omega\left(\rho, l_{\mathrm{r}}, u_{\mathrm{r}}\right) = \bar{\gamma}$, where the maximum fluxes are determined as in Section 4.

Define the mapping from our maximum fluxes $\bar{\gamma}$ to a "virtual" flux $\bar{\gamma}'$:

$$\bar{\gamma}_i' = V\left(\bar{\gamma}_i, \beta\right) = \begin{cases} \left(1-\beta\right)\bar{\gamma}_i & i = 1 \\ \bar{\gamma}_i & \text{o.w.} \end{cases}$$

Define the virtual junction flux solver as $JS'\left(\bar{\gamma}', P\right) = \gamma'$, which takes the virtual demands from the incoming mainline and ramp and the supply of the outgoing mainline, as well as a priority factor P, and produces resultant fluxes γ' by applying the flow maximization criteria across a 2-to-1 merge model of [**?**].

Lemma 1. $\sum JS'\left(\bar{\gamma}', P\right) = 2\left(\min\left(\bar{\gamma}_1\left(1-\beta\right) + \bar{\gamma}_r, \bar{\gamma}_2\right)\right)$

Proof. Result of flow maximization criteria and the virtual mapping $V\left(\right)$. ☐

Let the limiting side, LS ($\bar{LS}$ being the complement), being either *demand-limited* (DL) or *supply-limited* (SL), be defined by:

$$LS\left(\bar{\gamma}\right) = \begin{cases} DL & \bar{\gamma}_2 < \bar{\gamma}_1\left(1-\beta\right) + \bar{\gamma}_{\mathrm{r}} \\ SL & \text{otherwise} \end{cases}$$

The definition is modified for the virtual problem to be:

$$LS'\left(\bar{\gamma}'\right) = \begin{cases} DL & \bar{\gamma}_2' < \bar{\gamma}_1' + \bar{\gamma}_{\mathrm{r}}' \\ SL & \text{otherwise} \end{cases}$$

Lemma 2. $LS\left(\bar{\gamma}\right) = LS'\left(V\left(\bar{\gamma}\right)\right)$

Proof. From the LS, LS' definitions and the definition of the virtual map $V\left(\right)$. ☐

Define the mapping from initial density ρ and resultant flux γ to link-boundary density $\hat{\rho}$ as $\hat{\rho}_{\mathrm{in}}\left(\gamma, \rho\right)$ or $\hat{\rho}_{\mathrm{out}}\left(\gamma, \rho\right)$, and let these definitions match [**?**].

We can now define or Riemann solver $RS\left(\rho_1, \rho_2, l_{\mathrm{r}}, u, \beta, P, \bar{D}\right) = \left(\hat{\rho}_1, \hat{\rho}_2, \gamma_{\mathrm{r}}\right)$. Determined by the following process:

1. Determine demands and supplies $\bar{\gamma} = \Omega\left(\rho, l, u\right)$ for the mainlines and ramp.

2. Map to virtual fluxes $\bar{\gamma}' = V\left(\bar{\gamma}\right)$

3. Apply the virtual junction solver $\gamma' = JS'\left(\bar{\gamma}'\right)$

4. Map back to standard fluxes $\gamma = V^{-1}\left(\gamma'\right)$

5. Solve for boundary densities $\hat{\rho}_{\mathrm{in}}\left(\gamma_1, \rho_1\right)$, $\hat{\rho}_{\mathrm{out}}\left(\gamma_2, \rho_2\right)$, ramp flow γ_{r} and offramp flow $\beta\gamma_1$

Existence and uniqueness of the outputs of $RS(\cdot)$ are given by the existence and uniqueness of our virtual mapping functions $V(\cdot), V^{-1}(\cdot)$ and the results of [?]. The density profile solution is also admissible in the sense of [?], as the split ratio matrix is satisfied, there is flow conservation across the junction (Rankine-Huginiot condition at junctions), and the density profiles within cells are consistent with the weak solution of the LWR equation. What we have left to show is that our Riemann solver is self-similar, or more specifically:

$$RS\left(\hat{\rho}_1, \hat{\rho}_2, l_{\mathrm{r}}, u, \beta, P, \bar{D}\right) = RS\left(\rho_1, \rho_2, l_{\mathrm{r}}, u, \beta, P, \bar{D}\right) = \left(\hat{\rho}_1, \hat{\rho}_2, \gamma_{\mathrm{r}}\right)$$

We note that the onramp's demand is the same for both expressions above and it uses the γ_{r} value to updates its queue length l:

$$\dot{l} = \bar{D} - \gamma_{\mathrm{r}}$$

To show the self-similar property, we show that we can equivalently demonstrate that the virtual flux out of link 1 is self-similar in the virtual junction solver $JS'(\cdot)$.

Lemma 3. $\gamma = \bar{\gamma} \implies \hat{\Omega} = \Omega,$

Lemma 4. $\hat{\Omega}_{\bar{LS}} \supseteq \Omega_{\bar{LS}}$

Lemma 5. $\hat{\Omega}_{LS} = \Omega_{LS}$

Proof. From definition of $\hat{\rho}_{\mathrm{in}}(\gamma, \rho)$ and $\hat{\rho}_{\mathrm{out}}(\gamma, \rho)$. □

Lemma 6. $\hat{LS'} = LS'$

Proof. From Lemmas 4, 5 □

Lemma 7. $\hat{\gamma'}_2 = \gamma'_2$

Proof. From Lemma 5, 6 and flow maximization in Lemma 1, we have that the total flux across the junction for the Riemann solution is identical to the initial problem. □

Lemma 8. *If* $\hat{\gamma'}_1 = \gamma'_1$, then $RS(\cdot)$ is self-similar.

Proof. From Lemma 7, we have $\hat{\gamma'}_1 = \gamma'_1 \implies \hat{\gamma'} = \gamma'$. Clearly, $\hat{\hat{\gamma}}_{\mathrm{r}} = \hat{\gamma}_{\mathrm{r}}$. Now we show show $\hat{\gamma'} = \gamma'$ implies the Riemann solver is self-similar. The $\hat{\rho}$ solutions have the property that $\hat{\rho}(f(\hat{\rho}), \hat{\rho}) = \hat{\rho}$, and $f(\hat{\rho}) = \gamma$. From the uniqueness of the inverse virtual mapping $V^{-1}(\cdot)$, $\hat{\gamma'} = \gamma' \implies \hat{\gamma} = \gamma$. Therefore,

$$\hat{\gamma'} = \gamma' \implies \tag{14}$$

$$\hat{\gamma} = \gamma \implies \tag{15}$$

$$\hat{\rho}(\hat{\gamma}, \hat{\rho}) = \hat{\rho}(\gamma, \hat{\rho}) = \hat{\rho}(f(\hat{\rho}), \hat{\rho}) = \hat{\rho} \tag{16}$$

□

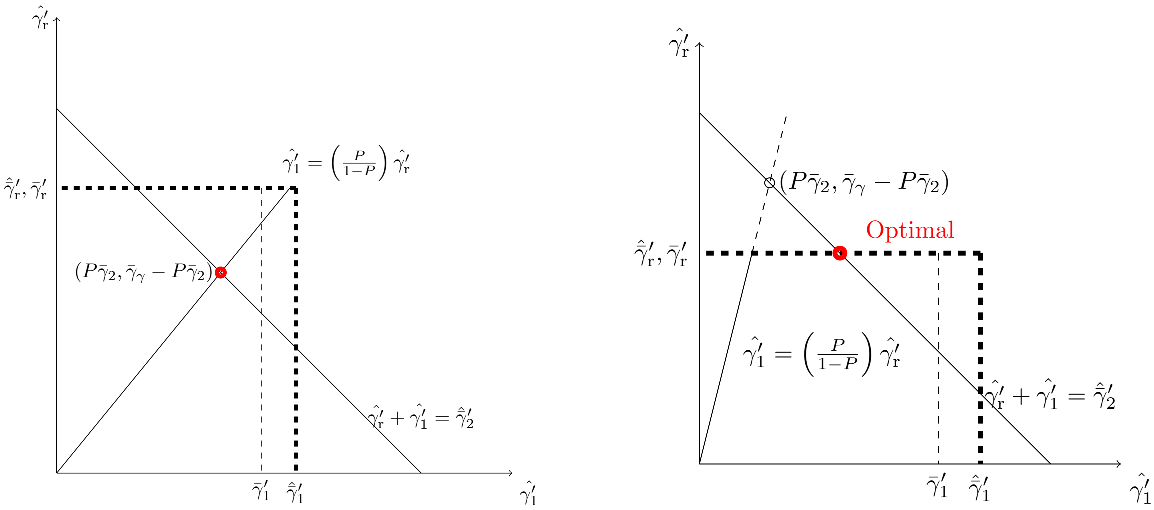
Theorem 9. $RS(\cdot)$ is self-similar.

We need to show $\hat{\gamma'}_1 = \gamma'_1$. When the problem is demand-limited, it is obvious using Lemmas 5 and 6. When the problem is supply-limited, we give some properties. The virtual incoming fluxes are determined by $JS'(\cdot)$ by minimizing the Euclidean distance of $(\gamma_1, \gamma_{\mathrm{r},})$ from $(P\bar{\gamma}_2, \bar{\gamma}_\gamma - P\bar{\gamma}_2)$ such that $(\gamma_1, \gamma_{\mathrm{r},})$ is feasible. Note that the reference point $(P\bar{\gamma}_2, \bar{\gamma}_\gamma - P\bar{\gamma}_2)$ will not change with assumption of being supply-limited. Additionally, Lemma 4 tells us the feasible set of $(\gamma_1, \gamma_{\mathrm{r},})$ only increases.

We consider the case when $\gamma_1 < \bar{\gamma}_1, \gamma_{\mathrm{r}} < \bar{\gamma}_{\mathrm{r}}$, and otherwise. For the first case, the reference point is feasible, and thus reference point will be optimal for both the original and final problem.

Otherwise, one of the incoming links i is at $\bar{\gamma}_i$, and the optimal point of the initial problem resides on the boundary of the feasible region. Lemma 3 tells us the maxed link's demand will not change. Combined with the fact that the feasible region of the demands for the virtual junction problem in the Riemann solution $(\bar{\gamma}'_1, \bar{\gamma}'_{\mathrm{r}})$ only increases (from Lemma 4), the optimal point of the junction problem will remain at the same boundary point.

For all cases, we have demonstrated $\gamma'_1 = \gamma_1$, thus completing the proof. See Figure 5 for an illustration of this proof.



(a) Interior Example

(b) Boundary Example

Figure 5: Illustration of Theorem 9

9.4 Junction model

We consider a junction i with one incoming mainline i, modeled by the real interval $(-\infty, 0]$, with outgoing flux $f_i^{\text{out}}(t)$, one outgoing mainline $i+1$, modeled by the real interval $[0, +\infty)$, with incoming flux $f_{i+1}^{\text{in}}(t)$, one onramp, and one offramp. The density on the mainline satisfies the PDE

$$\partial_t \rho_i(x,t) + \partial_x f(\rho_i(x,t)) = 0$$

The onramp is modeled by a buffer, with size $l_i(t)$. The dynamics of the ramp buffer are simply given by the ODE

$$\frac{d}{dt} l(t) = \bar{D}_i(t) - r_i(t) \tag{17}$$

$$l_i(0) = l_i^0 \tag{18}$$

where l_i^0 is a given initial size for the buffer, and $r_i(t)$ is the outgoing flux from the ramp, given by solving the junction problem.

$$d_i(t) = \begin{cases} \min(u_i(t), r_i^{\max}) & \text{if } l_i(t) > 0 \\ \min(u_i(t), r_i^{\max}, \bar{D}_i(t)) & \text{if } l_i(t) = 0 \end{cases}$$

$$\delta_i(t) = \min\left(F_i, v_i \rho_i(t)\right)$$

$$\sigma_{i+1}(t) = \min\left(F_{i+1}, w_{i+1}\left(\rho_{i+1}^{\text{jam}} - \rho_{i+1}(t)\right)\right)$$

$$f_{i+1}^{\text{in}}(t) = \min\left((1-\beta_i)\delta_i(t) + d_i(t), \sigma_{i+1}(t)\right)$$

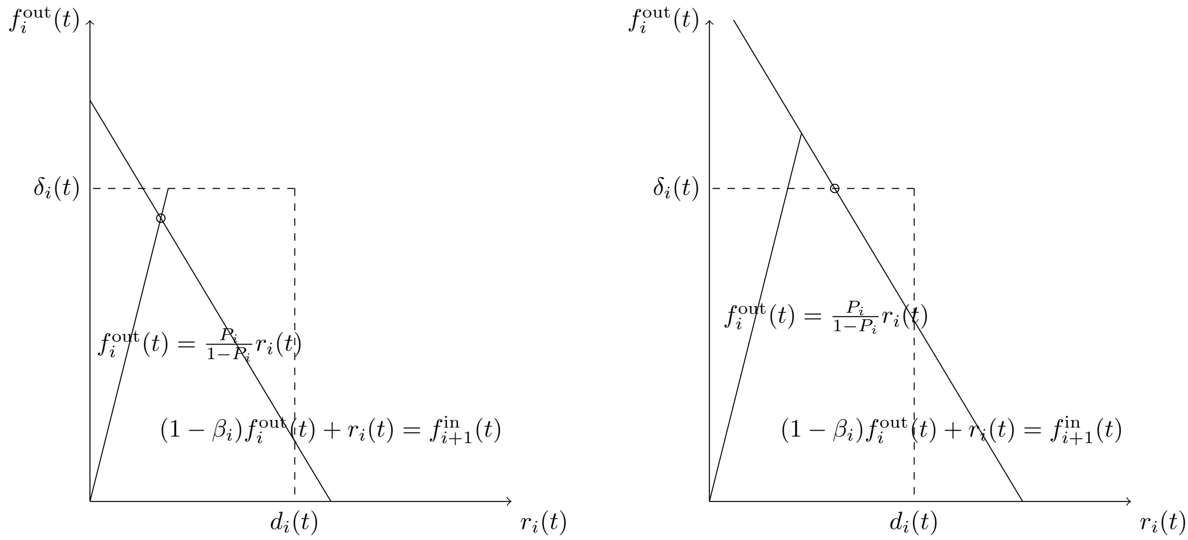
finally, the outgoing flux from the mainline $f_i(t)$ and the outgoing flux from the ramp $r_i(t)$ are given by the solution to the following problem

$$\begin{aligned} \text{minimize} \quad & \left\| \begin{pmatrix} r_i(t) \\ f_i^{\text{out}}(t) \end{pmatrix} - \begin{pmatrix} r_i(t) \\ f_i^{\text{out}}(t) \end{pmatrix} \cdot \alpha^{P_i} \alpha^{P_i} \right\|_2^2 \\ \text{subject to} \quad & f_{i+1}^{\text{in}}(t) = (1-\beta_i) f_i^{\text{out}}(t) + r_i(t) \\ & r_i(t) \le d_i(t) \\ & f_i^{out}(t) \le \delta_i(t) \end{aligned} \tag{19}$$

where α^{P_i} is the normalized vector

$$\alpha^{P_i} = \frac{1}{\sqrt{P_i^2 + (1-P_i)^2}} \begin{pmatrix} P_i \\ 1 - P_i \end{pmatrix}$$

This junction model ensures that the flux into the mainline is maximized, and the flux solution of the junction solver is unique (the objective function of the optimization problem is a non-degenerate quadratic, thus strictly convex). The optimization problem guarantees a unique solution, by minimizing the distance to the line $\Delta_i^P : f_i^{\text{out}}(t) = \frac{P_i}{1-P_i} r_i(t)$.



(a) Intersection inside the feasible set

(b) Intersection outside the feasible set

Figure 6: Junction flows given by the junction solver

9.5 Riemann problem

We consider a Riemann problem at junction i and at time t_0, where the inputs are constant in time, and we look for a flux solution that is also constant in time (until the next shock), thus we will drop the dependency on time in the input flux and the junction fluxes. Since the fluxes are constant, we have $\frac{d}{dt} l_i(t) = \bar{D}_i - r_i$, therefore the buffer size grows linearly in time.

TODO: define a solution to the Riemann problem

9.6 Self-similar solution

At t_0^+, starting from the solution $(\bar{\rho}_i(t_0), \bar{\rho}_{i+1}(t_0))$ of the Riemann solver, we show that the flux solution to the junction solver are invariant,

$$\mathcal{JS}_{l_i(t_0^+)}(\bar{\rho}_i(t_0), \bar{\rho}_{i+1}(t_0)) = \mathcal{JS} l_i(t_0)(\rho_i(t_0), \rho_{i+1}(t_0))$$

and as a consequence, the solution of the Riemann solver is invariant

$$\mathcal{RS}_{l_i(t_0^+)}(\bar{\rho}_i(t_0), \bar{\rho}_{i+1}(t_0)) = (\bar{\rho}_i(t_0), \bar{\rho}_{i+1}(t_0))$$

In other words, we need to show that $r_i(t_0^+) = r_i(t_0)$, and $f_i^{\text{out}}(t_0^+) = f_i^{\text{out}}(t_0)$.

9.6.1 Initially empty buffer

We first consider the case where the buffer is initially empty, $l_i(t_0) = 0$, and the input flux is greater than the output flux, i.e. $\bar{D}_i > r_i$, and the buffer is growing linearly. Thus $l_i(t_0^+) > 0$. From these assumptions, the onramp demand is given by

$$d_i(t_0) = \min(u_i, r_i^{\max}, \bar{D}_i)$$
$$d_i(t_0^+) = \min(u_i, r_i^{\max})$$

Remark We observe that the ramp demand at time t_0^+ can only increase, i.e.

$$d_i(t_0^+) \geq d_i(t_0) \tag{20}$$

since

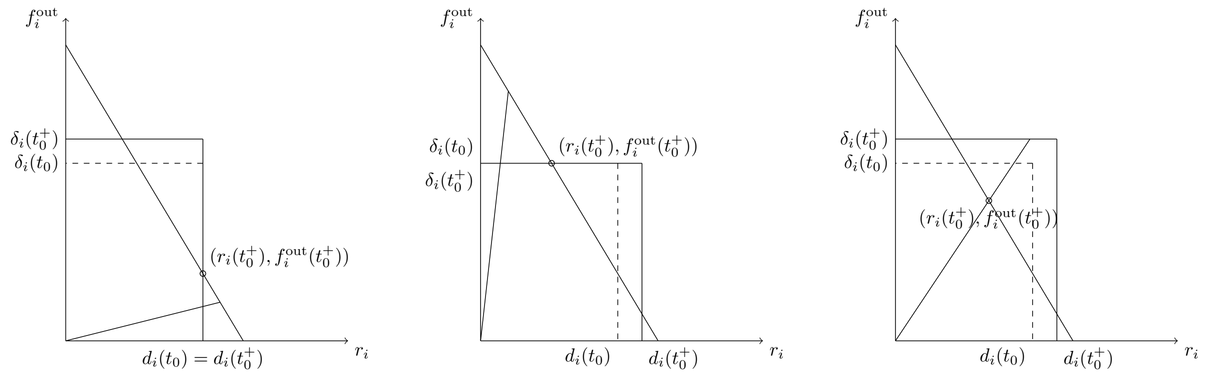
$$\begin{aligned} d_i(t_0^+) &= \min(u_i, r_i^{\max}) \\ &\geq \min(u_i, r_i^{\max}, \bar{D}_i) \\ &= d_i(t_0) \end{aligned}$$

Moreover, if $r_i(t_0) = d_i(t_0)$, then we have equality $d_i(t_0^+) = d_i(t_0)$. Proof: since $r_i(t_0) = d_i(t_0) = \min(u_i, r_i^{\max}, \bar{D}_i)$ and $r_i(t_0) < \bar{D}_i$ (the buffer is growing), we necessarily have $\min(u_i, r_i^{\max}) < \bar{D}_i$, thus $\min(u_i, r_i^{\max}) = \min(u_i, r_i^{\max}, \bar{D}_i)$.

Supply-constrained case First, we assume that the junction is supply-constrained at time t_0, i.e. $(1-\beta_i)\delta_i(t_0) + d_i(t_0) > \sigma_{i+1}(t_0)$. Therefore the mainline supply at time t_0^+ is

$$\sigma_{i+1}(t_0^+) = \sigma_{i+1}(t_0) \tag{21}$$

We consider three cases, depending on the flux solution of the junction solver.



(a) Solution on the boundary $r_i = d_i$ (b) Solution on the boundary $f_i^{\text{out}} = \delta_i$ (c) Solution in the interior of the feasible set

Figure 7: Self-similar solution in the case of a supply-constrained junction problem at time t_0

(a) The intersection is on the boundary of the feasible set

$$r_i(t_0) = d_i(t_0)$$

By the previous remark, we have

$$d_i(t_0) = d_i(t_0^+)$$

The mainline demand can only increase $\delta_i(t_0^+) \geq \delta_i(t_0)$ (in fact $\delta_i(t_0^+) = f_i^{\max}$), and by (21), $\sigma_{i+1}(t_0^+) = \sigma_{i+1}(t_0)$. Therefore we have

$$\begin{aligned} r_i(t_0) &= d_i(t_0^+) \\ f_i^{\text{out}}(t_0) &\leq \delta_i(t_0^+) \\ (1-\beta_1)f_i^{\text{out}}(t_0) + r_i(t_0) &< \sigma_{i+1}(t_0^+) \end{aligned}$$

Therefore $(f_i^{\text{out}}(t_0), r_i(t_0))$ is a feasible point for the junction problem at time t_0^+, and is thus the unique solution (see Figure 7a)

(b) The intersection is on the boundary of the feasible set

$$f_i^{\text{out}}(t_0) = \delta_i(t_0)$$

In this case the mainline demand is $\delta_i(t_0^+) = f_i^{\text{out}}(t_0)$, and by (20) and (21)

$$\begin{aligned} r_i(t_0) &\le d_i(t_0^+) \\ f_i^{\text{out}}(t_0) &= \delta_i(t_0^+) \\ (1-\beta_1)f_i^{\text{out}}(t_0) + r_i(t_0) &< \sigma_{i+1}(t_0^+) \end{aligned}$$

Therefore $(f_i^{\text{out}}(t_0), r_i(t_0))$ is a feasible point for the junction problem at time t_0^+, and is thus the unique solution (see Figure 7b)

(c) The intersection is strictly inside the feasible set

$$\begin{aligned} r_i(t_0) &< d_i(t_0) \\ f_i^{\text{out}}(t_0) &< \delta_i(t_0) \end{aligned}$$

The mainline demand is $\delta_i(t_0^+) = f_i^{\max}$. The ramp demand is $d_i(t_0^+) = \min(u_i, r_i^{\max}) \ge \min(u_i, r_i^{\max}, \bar{D}_i) = d_i(t_0)$. Therefore we have

$$\begin{aligned} r_i(t_0) &< d_i(t_0^+) \\ f_i^{\text{out}}(t_0) &< \delta_i(t_0^+) \\ (1-\beta_1)f_i^{\text{out}}(t_0) + r_i(t_0) &< \sigma_{i+1}(t_0^+) \end{aligned}$$

Therefore $(f_i^{\text{out}}(t_0), r_i(t_0))$ is a feasible point for the junction problem at time t_0^+, and is thus the unique solution (see Figure 7c)

Demand-constrained case Now assume the junction is demand-constrained, i.e. $(1-\beta_i)\delta_i(t_0) + d_i(t_0) < \sigma_{i+1}(t_0)$. Then the feasible set contains a single point, and we have

$$\begin{aligned} f_i^{\text{out}}(t_0) &= \delta_i(t_0) \\ r_i(t_0) &= d_i(t_0) \end{aligned}$$

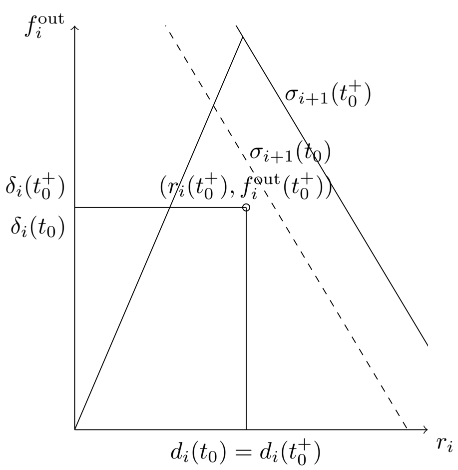


Figure 8: Self-similar solution in the case of a demand-constrained junction problem at time t_0

At time t_0^+, the mainline demand is $\delta_i(t_0^+) = f_i^{\text{out}}(t_0)$, the ramp demand is $d_i(t_0^+) = d_i(t_0)$ (by the previous remark), and the supply can only increase, $\sigma_{i+1}(t_0^+) \geq \sigma_{i+1}(t_0)$. Therefore

$$\begin{aligned} r_i(t_0) &= d_i(t_0^+) \\ f_i^{\text{out}}(t_0) &= \delta_i(t_0^+) \\ (1-\beta_1) f_i^{\text{out}}(t_0) + r_i(t_0) &< \sigma_{i+1}(t_0^+) \end{aligned}$$

i.e. $(f_i^{\text{out}}(t_0), r_i(t_0))$ is a feasible point for the junction problem at time t_0^+, and is thus the unique solution (see Figure 8)

9.6.2 Initially non-empty buffer

A Partial derivatives with respect to x

The state variables x contains two variables ρ and l that the objective function J depends on. Therefore, $\frac{\partial J}{\partial x}$ is a row vector of length $|x|$ with zero's in all the partial derivative with respect to all other variables of the state vector.

$$\frac{\partial J}{\partial x_0(k)} = (1, 0, 0)$$
$$\frac{\partial J}{\partial x_i(k)} = (1, 0, 0, 0, 0, 0, 0, 1)$$
$$\frac{\partial J}{\partial x_N(k)} = (1, 0, 0)$$

The matrix $\frac{\partial H}{\partial x}$ is given in sparse, tabular format in Table 4.

Constraint	Variable	Partial Derivative
$H^1_{k,i}$	$\rho_i(k)$	1
	$\rho_i(k-1)$	-1
	$f_i^{\text{in}}(k-1)$	$-\frac{\triangle t}{\triangle x}$
	$f_i^{\text{out}}(k-1)$	$\frac{\triangle t}{\triangle x}$
$H^1_{k,0}$	$\rho_0(k)$	1
	$\rho_0(k-1)$	-1
	$f_0^{\text{out}}(k-1)$	$\frac{\triangle t}{\triangle x}$
$H^1_{k,N}$	$\rho_N(k)$	1
	$\rho_N(k-1)$	-1
	$f_N^{\text{out}}(k-1)$	$-\frac{\triangle t}{\triangle x}$
	$\delta_N(k)$	$\frac{\triangle t}{\triangle x}$
$H^1_{0,i}$	$\rho_i(0)$	1
$H^2_{k,i}$	$l_i(k)$	1
	$l_i(k-1)$	-1
	$r_i(k)$	$\triangle t$
$H^2_{0,i}$	$l_i(0)$	1
$H^3_{k,i}$	$\delta_i(k)$	1
	$\rho_i(k)$	$\begin{cases} -v_i & v_i\rho_i(k) < F_i \\ 0 & \text{otherwise} \end{cases}$
$H^4_{k,i}$	$\sigma_i(k)$	1
	$\rho_i(k)$	$\begin{cases} w_i & w_i\left(\rho_i^{\text{jam}} - \rho_i(k)\right) < F_i \\ 0 & \text{otherwise} \end{cases}$
$H^5_{k,i}$	$d_i(k)$	1
	$l_i(k)$	$\begin{cases} -1 & l_i(k) < u_i(k) \\ 0 & \text{otherwise} \end{cases}$
$H^6_{k,i}$	$f_i^{\text{in}}(k)$	1
	$d_{i-1}(k)$	$\begin{cases} -1 & \delta_{i-1}(k)(1-\beta_{i-1}(k)) + d_{i-1}(k) < \sigma_i(k) \\ 0 & \text{otherwise} \end{cases}$
	$\delta_{i-1}(k)$	$\begin{cases} -(1-\beta_{i-1}(k)) & \delta_{i-1}(k)(1-\beta_{i-1}(k)) + d_{i-1}(k) < \sigma_i(k) \\ 0 & \text{otherwise} \end{cases}$
	$\sigma_i(k)$	$\begin{cases} -1 & \sigma_i(k) \le \delta_{i-1}(k)(1-\beta_{i-1}(k)) + d_{i-1}(k) \\ 0 & \text{otherwise} \end{cases}$

$H_{k,1}^{6}$	$f_1^{\text{in}}(k)$	1
	$\delta_0(k)$	$\begin{cases} -1 & \delta_0(k) < \sigma_i(k) \\ 0 & \text{otherwise} \end{cases}$
	$\sigma_0(k)$	$\begin{cases} -1 & \sigma_i(k) \leq \delta_0(k) \\ 0 & \text{otherwise} \end{cases}$
$H_{k,i}^{7}$	$f_i^{\text{out}}(k)$	1
	$f_{i+1}^{\text{in}}(k)$	$\begin{cases} \frac{-1}{(1-\beta_i(k))} & \text{Case 1, 4} \\ -\frac{P_i}{(1-\beta_i(k))} & \text{Case 5} \\ 0 & \text{otherwise} \end{cases}$
	$\delta_i(k)$	$\begin{cases} -\frac{1}{(1-\beta_i(k))} & \text{Case 2,3} \\ 0 & \text{otherwise} \end{cases}$
	$d_i(k)$	$\begin{cases} 1 & \text{Case 4} \\ 0 & \text{otherwise} \end{cases}$
$H_{k,0}^{7}$	$f_0^{\text{out}}(k)$	1
	$f_1^{\text{in}}(k)$	-1
$H_{k,i}^{8}$	$r_i(k)$	1
	$f_{i+1}^{\text{in}}(k)$	-1
	$f_i^{\text{out}}(k)$	$(1-\beta_i(k))$

Table 4: Sparse format of $\frac{\partial H(i,k)}{\partial x}$

References

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

[2] M. Garavello and B. Piccoli. *Traffic flow on networks*, volume 1. American institute of mathematical sciences Springfield, MA, USA, 2006.

[3] Jerome Thai. Polyhedral hybrid systems and applications to the Cell Transmission Model.