2015 American Control Conference
Palmer House Hilton
July 1-3, 2015. Chicago, IL, USA

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

S. Samaranayake, J. Reilly, W. Krichene, J.B. Lespiau, M.L. Delle Monache, P. Goatin and A. Bayen

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

I. Introduction

Dynamic traffic assignment (DTA) is the process of allocating time-varying *origin-destination* (OD) based traffic demand across a road network [1], [2]. There are two types of traffic assignment: the *user equilibrium* or Wardrop equilibrium (UE-DTA), where users minimize individual travel-time in a selfish manner, and the *system optimal* allocation (SO-DTA) where a central authority picks the route for each user to minimize the aggregate total travel-time over all users [3]. *User equilibrium* (UE) traffic assignment can lead to inefficient network utilization, highlighted by Braess' Paradox [4], where adding capacity to the network can actually result in longer travel times for all users. This inefficiency can occur in real road networks [5]. SO traffic assignment on the other hand leads to optimal utilization of the network resources, but is hard to achieve in practice since the overriding objective for individual vehicles is to minimize their own travel-time. Setting a toll on each road segment equal to the marginal delay of the demand results in an SO allocation, even with selfish behavior [6]. However, imposing time-varying tolls on each road segment is impractical and difficult to implement in many settings due to both infrastructure and political considerations.

An alternative approach is to attempt to control a fraction of the vehicles[1] by assigning routes via a central authority (e.g. smart phone application) that tries to minimize system wide total travel-time. This strategy has been attempted in communication networks with non-decreasing latency functions and vertical queues. However, these assumptions are generally not satisfied in road traffic networks, with horizontal queues due to congestion propagation and more complex latency functions due to the physics of flows and driver behavior [7]. The literature on partial control in traffic assignment is sparse and the scope of this work has been limited. For example, Aswani et al. [8] use vertical queues and non-decreasing latency functions, while Krichene et al. [9] consider simple parallel networks.

Ziliaskopoulos [10] formulated the single destination SO-DTA problem (with full control) as a *Linear Program* (LP) under a LP relaxation of the non-linear system dynamics. However, the SO-DTA problem with partial control can not be formulated as a convex problem, even in the case of a single destination, without violating the *first-in-first-out* (FIFO) condition [11], due to the multiple commodities (selfish and cooperative agents) in this problem. Furthermore, solving the SO-DTA problem with an LP relaxation of the dynamics can lead to the holding of vehicles on links when the model allows for a larger flow. This is however not a feasible solution for actual roadways. Thus, there is a need for a more general solution that does not impose holding.

We formulate the *system optimal dynamic traffic assignment problem with partial control* (SO-DTA-PC), using a traffic dynamics model based on a Godunov discretization of the *Lighthill-Williams-Richards* (LWR) *partial differential equation* (PDE) [12], [13] with a triangular fundamental diagram[2]. This gives a horizontal queuing model with a latency function that that is well accepted in the transportation community as a good first order approximation of road traffic dynamics.

S. Samaranayake, J. Reilly, W.Krichene and A. Bayen are with the University of California Berkeley e-mail: {samitha, jackdreilly, walid, bayen}@berkeley.edu

J.B. Lespiau is with Ecloe Polytechnique e-mail: jean-baptiste.lespiau@polytechnique.edu@path.berkeley.edu

M.L. Delle Monache and P. Goatin are with Inria Sophia Antipolis, e-mail: maria-laura.delle_monache, paola.goatin@inria.fr

[1]Controlling all the vehicles would be ideal, but not always possible.
[2]The flux as a function of the density takes a triangular shape.

One major difficulty of DTA in practical settings is the unavailability of *origin-destination* (OD) data for the entire demand. Therefore, we formulate the partial control problem in a manner that only requires full OD information for the agents that can be controlled by the central authority and only requires junction split ratios (which are much easier to obtain via for example inductive loop detectors) for the rest of the demand.

Solving the SO-DTA-PC subject to this model requires solving a non-linear optimization problem. While gradient based methods do not provide any guarantees of converging to the optimal solution in non-linear optimization problems, they can still be used to find local minima and it is a common approach to use gradient descent methods with multiple start points. One of the main computational challenges in this approach is the efficient computation of the gradient, since this computation must be repeated a large number of times. We show how the structure of our dynamical system allows for very efficient computation of the gradient via the discrete adjoint method [14]. If the state vector is n dimensional and the control vector is m dimensional, direct computation of the gradient takes $O(n^2m)$ time. The adjoint methods reduces the complexity to $O(n^2 + nm)$, but the structure of our system allows for further reduction of the complexity to $O(n+m)$ by avoiding a matrix inversion and solving the system via backwards substitution.

The contributions of this article are as follows: 1) formulation of the SO-DTA-PC problem as a multi-commodity finite horizon optimal control problem, 2) defining the appropriate multi-commodity junction model for the network dynamics, 3) solving the gradient of the system with $O(n+m)$ time complexity for a n dimensional state space and m dimensional control vector using the discrete adjoint method, 4) Experimental results for showcasing the benefits and applications of the technique[3].

II. Traffic model

The aggregate traffic dynamics are modeled using a macroscopic traffic flow model [12], [13]. We use a multicommodity variant of earlier PDE model developed in [15]. This model imposes strong boundary conditions at the entrances to the network, so that no vehicles are dropped due to congestion propagating outside the bounds of the network, an important consideration in the optimal control setting. We then use a Godunov discretization [16] of the network PDE model as explained in [17] to obtain an equivalent discrete model.

A. Network model

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

The supply of a cell i at time step k, denoted $\sigma_i(k)$, is the flow it can accept from its predecessor cell, while the demand $\delta_i(k)$ is the flow that is trying to leave the cell. Note that buffers have no supply and the sinks have no demand.

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

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

The inflow (resp. outflow) from a cell i at time step k, denoted $f_i^{\text{in}}(k)$ (resp. $f_i^{\text{out}}(k)$), is the total flow leaving (resp. entering at) the cell at time step k. Note that buffers have no inflow and sinks have no outflow. The inflow (resp. outflow) from a cell i at time step k corresponding to commodity c, denoted $f_{i,c}^{\text{in}}(k)$ (resp. $f_{i,c}^{\text{out}}(k)$), is the total flow of commodity c leaving (resp. arriving at) the cell at time step k.

The state of the network at time step k is given by the density $\rho_{i,c}(k)$ of each commodity c at each cell i. The density evolution is governed by the following dynamics under the time discretization of Δt.[4]

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

$$\rho_{i,c}(k) = \rho_{i,c}(k-1) + \frac{\Delta t}{L_i} \cdot f_{i,c}^{\text{in}}(k-1) \quad \forall i \in \mathcal{S},\ \forall k \in [\![1, T_f]\!],\ \forall c \in \mathcal{C} \tag{2}$$

[3]It should be noted that this framework does not consider the demand response of the selfish agents in response to the control, which will impact the network conditions in the setting of a repeated game.

[4]The discretization of the system must satisfy the *Courant-Friedrichs-Lewy* (CFL) for numerical stability. See section 2.1 in [18] for more details.

with initial condition

$$\rho_{i,c}(0) = \rho^0_{i,c} \qquad \forall i \in \mathcal{A} \setminus \mathcal{S},\ \forall c \in \mathcal{C} \tag{3}$$

$$\rho_{i,c}(0) = 0 \qquad \forall i \in \mathcal{S},\ \forall c \in \mathcal{C} \tag{4}$$

Assumption 1. The flux function defining the relationship between density and flow is obtained using given a first order approximation of the empirical relationship between flow and density [19], which results in the standard triangular fundamental diagram.

Assumption 2 (First-in first-out (FIFO) property). We assume that no vehicles leaving the origin at a time step $t > t'$ will overtake the agents that have left the origin at time step t'.

B. Controllable and non-controllable flow

There are two types of flows that are transported in the network. Controllable flows that have origin destination requirements, but can be routed along any path in the network, and non-controllable flows that have fixed paths. These flows are modeled by distributing the total flow of the network into multiple commodities.

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

There is a single non-controllable commodity c_n that represents all non-controllable flow in the network. The paths of the flow corresponding to the non-controllable commodity are defined via the junction split ratios. The split ratio of a commodity c at cell i and time step k among the outgoing cells $j \in \Gamma(i)$, denoted $\beta_{ij,c}(k)$, is the fraction of the commodity c flow out of cell i that is entering cell j at time step k.

$$\sum_{j \in \Gamma(i)} \beta_{ij,c}(k) = 1 \tag{5}$$

The controllable commodities $c_c \in \mathcal{CC}$ correspond to the controllable flow. There is a unique controllable commodity that corresponds to each path that the controllable flow can be routed along in the network. A controllable commodity is then equivalent to a tuple $(origin, destination, path)$. The path of a controllable commodity is defined via the binary junction split ratio for each commodity.

The number of controllable vehicles wanting to travel from origin $o \in \mathcal{B}$ to destination $s \in \mathcal{S}$ at time step k, denoted $D_{(o,s)}(k)\,\Delta t$ is an exogenous input.

Assumption 4 (Data requirements). We assume that the origin destination requirements of all the controllable flows and the aggregate path information for all the non-controllable flows are known.

While at first glance this might seem like a lot of information to gather, it is in fact reasonable to assume in road traffic networks. We assume that the controllable flows are vehicles that are cooperating with the traffic coordination system that is trying to route vehicles efficiently and therefore will share their origin destination requirements. The aggregate paths of the non-controllable flows can be obtained by looking at historical traffic patterns and the empirical aggregate split ratios seen in this data. The caveat is that split ratios also include the contribution of the controllable flows and therefore must be pre-processed to remove this contribution. The junction split ratios for the non-controllable commodities are time-dependent while the junction split ratios for the controllable commodities are not.

Definition 1 (Equivalence between compliant commodity and path). Any compliant commodity $c \in \mathcal{CC}$ is a tuple (o, s, p) where $o \in \mathcal{B}$, $s \in \mathcal{S}$ and p describes a path (i.e. a sequence of cells). We define the function Ω as follows:

$$\begin{aligned} \Omega\colon\ \mathcal{CC} &\to \mathcal{S} \times \mathcal{B} \\ c &\mapsto (o, s). \end{aligned} \tag{6}$$

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

Definition 2 (Compliant flow control). A control u is an allocation of the compliant agents over the set $\Omega^{-1}(o, s)$ for each time step. Formally u is defined as:

$$\begin{aligned} u\colon\ \mathcal{CC} \times [\![0, T-1]\!] &\to [0, 1] \\ (c, k) &\mapsto \gamma_c(k) \end{aligned} \tag{7}$$

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

Definition 3 (Physically feasible control). A physically feasible control u verifies the mass conservation of the compliant agents:

$$\sum_{c \in \Omega^{-1}(o,s)} \gamma_c(k) = 1 \quad \forall k \in [\![0, T]\!], (o, s) \in \mathcal{B} \times \mathcal{S} \tag{8}$$

Let $\mathcal{U}$ be the set of all physically feasible controls.

III. Forward system

A. Junction model

The junction model defines the dynamics of the flow between neighboring cells. It is required to satisfy the following properties.

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

$$f_{i,c}^{\text{out}}(k) = f_i^{\text{out}}(k) \frac{\rho_{i,c}(k)}{\rho_i(k)} \tag{9}$$

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

$$f_{ij,c}(k) = f_{i,c}^{\text{out}}(k) \cdot \beta_{ij,c}(k) \tag{10}$$

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

$$0 \leq f_i^{\text{in}}(k) \leq \sigma_i(k) \quad \forall k \in [\![0, T]\!] \tag{11}$$

$$0 \leq f_i^{\text{out}}(k) \leq \delta_i(k) \quad \forall k \in [\![0, T]\!] \tag{12}$$

We wish to define a multi-commodity junction flow solver that assigns flows across the network in a manner that is consistent with the above requirements.

Definition 4 (Priority vector). In the case of a junction where there is more than one incoming cell and the aggregate demand of these cells is greater than the aggregate supply of the outgoing cells, the available supply needs to be distributed among the competing demand according to some allocation vector. The priority vector P_j for cell j definies the allocation of its supply over the incoming cells $i \in \Gamma^{-1}(j)$. The priority for a given incoming cell i is given by P_{ij}.

$$\sum_{i \in \Gamma^{-1}(j)} P_{ij} = 1 \tag{13}$$

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

$$\begin{aligned} \beta_{ij}(k) &= \sum_{c \in \mathcal{C}} \frac{\rho_{i,c}(k)}{\rho_i(k)} \beta_{ij,c}(k) \\ &= \frac{1}{\rho_i(k)} \sum_{c \in \mathcal{C}} \rho_{i,c}(k) \beta_{ij,c}(k) \end{aligned} \tag{14}$$

Remark 1. The aggregate split ratio is only defined for positive aggregate densities, i.e. $\rho_i(k) > 0$

Now we consider the solutions to the flows across each junction in the model. We limit the discussion in this article to the junction solutions and refer to reader to section 3.1 of [18] for a detailed explanation of the junction model and proof of uniqueness of the solutions.

1) Diverge solver ($1 \times m$): We consider a diverging junction z with one incoming link i and m outgoing links. There are $|\mathcal{C}|$ commodities that flow through the network each with their own time-varying split ratio $\beta_{ij,c}(k)$.

Remark 2. If $\rho_i(k) = 0$, $\delta_i(k) = 0$ and $f_i^{\text{out}}(k)$ is zero. We consider the case of $\rho_i(k) \neq 0$.

The solution to the junction problem is given by the following equation.

$$f_i^{\text{out}}(k) = \min\left(\left\{ \frac{\sigma_j(k)}{\beta_{ij}(k)},\ \forall j \in \mathcal{J}_z^{\text{out}} \mid \beta_{ij}(k) > 0 \right\}, \delta_i(k) \right) \tag{15}$$

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

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

$$f_i^{\text{out}}(k) = \delta_i(k) \quad \forall i \in \mathcal{J}_z^{\text{in}} \tag{16}$$

Otherwise the problem is supply constrained and the solution to the junction problem is given by solving the following quadratic optimization problem that finds the flow maximizing solution with the smallest violation of the priority vector, where the violation is measured using the L_2 distance:

$$\min_{t, \left\{ f_i^{\text{out}}(k),\ \forall i \in \mathcal{J}_z^{\text{in}} \right\}} \sum_{i \in \mathcal{J}_z^{\text{in}}} \left(f_i^{\text{out}}(k) - t \cdot p_{ij} \right)^2 \tag{17}$$

subject to

$$\sum_{i \in \mathcal{J}_z^{\text{in}}} f_i^{\text{out}}(k) = \sigma_j(k)$$

$$0 \leq f_i^{\text{out}}(k) \leq \delta_i(k) \qquad \forall i \in \mathcal{J}_z^{\text{in}}$$

3) Merge-diverge solver ($2 \times m$): We consider a junction with 2 incoming links and m outgoing links[5].

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

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

allocate their available supply to the 2 incoming links. It satisfies $P_i = P_{i1} = P_{i2}$ and $\sum_{i \in \mathcal{J}_z^{\text{in}}} P_i = 1$.

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

$$f_i^{\text{out}}(k) = \delta_i(k) \quad \forall i \in \mathcal{J}_z^{\text{in}} \tag{18}$$

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

$$\min_{t, f_i^{\text{out}}(k)\ \forall i \in \mathcal{J}_z^{\text{in}}} \sum_{i \in \mathcal{J}_z^{\text{in}}} \left(f_i^{\text{out}}(k) - t \cdot P_i \right)^2 \tag{19}$$

subject to

$$\sum_{i \in \mathcal{J}_z^{\text{in}}} \beta_{ij}(k)\, f_i^{\text{out}}(k) \leq \sigma_j(k) \qquad \forall j \in \mathcal{J}_z^{\text{out}}$$

$$\max_j \left(\sum_{i \in \mathcal{J}_z^{\text{in}}} \beta_{ij}(k)\, f_i^{\text{out}}(k) - \sigma_j(k) \right) = 0 \quad \forall j \in \mathcal{J}_z^{\text{out}}$$

$$f_i^{\text{out}}(k) \leq \delta_i(k) \qquad \forall i \in \mathcal{J}_z^{\text{in}}$$

In all three cases, the total outflow $f_i^{\text{out}}(k)$ for each incoming link i is then divided among the commodities according to the FIFO law:

$$f_{i,c}^{\text{out}}(k) = \frac{\rho_{i,c}(k)}{\rho_i(k)} f_i^{\text{out}}(k) \tag{20}$$

The commodity flows are split among the outgoing links according to the split ratios constraints:

$$f_{j,c}^{\text{in}}(k) = \sum_{i:(i,j) \in A} \beta_{ij,c}(k)\, f_{i,c}^{\text{out}}(k) \tag{21}$$

B. Boundary conditions

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

Definition 5 (Boundary demand). The number of vehicles of commodity c leaving from cell $i \in \mathcal{B}$ at time step k is the boundary demand of commodity $d_{i,c}(k)$. Let c_{n} be the commodity corresponding to non-controllable flow. The non-zero terms are defined as:

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

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

Since the inflow to the network is limited by the max flow capacity and density of the immediate downstream link, all of the demand at a given time step might not make it into the network. A source buffer is used to accumulate the flow that can not enter the network to guarantee the conservation of boundary flows. Using a single source buffer could however violate the FIFO condition at the source. See section 3.2 of [18] for a discussion on preserving the FIFO condition at the source.

C. System dynamics

For a given control u, we can determine the evolution of the network using the system dynamics. Let $x(u)$ give the state of the network under these dynamics subject to the control u. The system of equations governing the evolution of the network (implicit definition of x) are written formally in the form $H(x, u) = 0$. The explicit formulation is given in section 3.3 of [18]. The dynamics equations have a topological ordering that allows for an efficient forward simulation algorithm, where much of the computation can be done in parallel.

IV. Adjoint based optimization

A. Problem formulation

The system optimal dynamic traffic assignment with partial compliance (SO-DTA-PC) is a physically acceptable (see Definition 2) division of the compliant agents among the different commodities that minimizes the total travel-time (including the travel-time of the non-compliant commodities). The total travel-time $J(x(u))$ is defined as:

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

The solution is obtained by minimizing the cost function $J(x(u))$ subject to the system dynamics given in section III-C and the following control constraints.

$$\gamma_c(k) \geq 0 \qquad \forall c \in \mathcal{CC}, k \in [\![0, T_f]\!]$$

$$\sum_{c \in \Omega^{-1}\{(o,s)\}} \gamma_c(k) = 1 \qquad \forall k \in [\![0, T_f]\!]$$

Note that this is a non-convex optimization problem that might contain multiple global minima. Therefore, gradient methods will not guarantee global optimality. However, descent algorithms can still be used to obtain locally optimal solutions and non-convex optimization techniques such as subgradient and interior point methods [20] can be aided by having the gradient of the system. We use the discrete adjoint method, which will be explained in the next section, to efficiently solve for the gradient of the system.

Obtaining a physically feasible control requires satisfying the two control constraints. However, the adjoint method that we will use to efficiently compute the gradient of the system does not allow for inequality constraints. Therefore, the $\gamma_c(k) \geq 0$ inequality constraint is satisfied by projecting any control values of the solution given by the gradient decent to the boundary of the feasible set of $\gamma_c(k) = 0$. This can be handles either via a barrier function or by augmenting the state space. See section 4.1 of [18] for an explanation.

B. Overview of the adjoint method

We consider the following general optimization problem:

$$\begin{aligned} \min_{u} \quad & J(x,u) \\ \text{subject to} \quad & H(x,u) = 0 \end{aligned} \tag{22}$$

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

The adjoint method [21] 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 a gradient descent based optimization. We suppose that for any control u, $\frac{\partial H}{\partial x}(x,u)$ is not singular. Under equality constraints $H(x,u) = 0$, the Lagrangian

$$L(x,u,\lambda) = J(x,u) + \lambda^T H(x,u) \tag{23}$$

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

$$\begin{aligned} \nabla_u L(x,u,\lambda) &= \frac{\partial J}{\partial u} + \frac{\partial J}{\partial x}\frac{d\mathcal{X}}{du} + \lambda^T\left(\frac{\partial H}{\partial u} + \frac{\partial H}{\partial x}\frac{dx}{du}\right) \\ &= \frac{\partial J}{\partial u} + \lambda^T\frac{\partial H}{\partial u} + \left(\frac{\partial J}{\partial x} + \lambda^T\frac{\partial H}{\partial x}\right)\frac{d\mathcal{X}}{du} \end{aligned} \tag{24}$$

In particular, if λ satisfies the adjoint equation:

$$\frac{\partial J}{\partial x} + \lambda^T\frac{\partial H}{\partial x} = 0 \tag{25}$$

then the gradient is,

$$\nabla_u L(x,u) = \frac{\partial J}{\partial u} + \lambda^T\frac{\partial H}{\partial u} \tag{26}$$

Remark 3. The solution for λ exists and is unique if $\frac{\partial H}{\partial x}$ is not singular, which is the case in our forward system, as explained in the following section.

C. Applying the adjoint method

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

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

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

$$\mathcal{X} = \begin{pmatrix} \left(\left(\rho_{i,c}(k)\right)_{c\in\mathcal{C}}\right)_{i\in\mathcal{A}} \\ \left(\left(f^{\text{out}}_{i,c}(k)\right)_{c\in\mathcal{C}}\right)_{i\in\mathcal{A}} \\ \left(\left(f^{\text{in}}_{i,c}(k)\right)_{c\in\mathcal{C}}\right)_{i\in\mathcal{A}} \end{pmatrix} \quad \mathcal{H} = \begin{pmatrix} \left(\left(H^1_{k,i,c}\right)_{c\in\mathcal{C}}\right)_{i\in\mathcal{A}} \\ \left(\left(H^5_{k,i,c}\right)_{c\in\mathcal{C}}\right)_{i\in\mathcal{A}} \\ \left(\left(H^6_{k,i,c}\right)_{c\in\mathcal{C}}\right)_{i\in\mathcal{A}} \end{pmatrix}$$

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

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

$$\nabla_u J(x,u) = \frac{\partial J}{\partial x} \cdot \frac{d\mathcal{X}}{du} + \frac{\partial J}{\partial u} \tag{27}$$

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

The discrete adjoint methods reduces this complexity to $O(n^2 + nm)$ by first computing the adjoint system. Computing the adjoint variables $\lambda^T \in R^n$ using equation (25) only takes $O(n^2)$ because it only requires

solving one $n \times n$ linear system. Multiplying $\lambda^T \frac{\partial H}{\partial u}$ and adding $\frac{\partial J}{\partial u}$ to complete the computation in equation (26) takes $O(nm)$ time, so the total computation time is $O(n^2 + nm)$.

The structure of our system allows for further reduction of the complexity to $O(n + m|\mathcal{C}|)$. As shown in section IV-C, $\frac{\partial H}{\partial x}$ is a lower triangular matrix and therefore we can compute the solution to equation (25) using backwards substitution. We will exploit the fact that the matrix $\frac{\partial H}{\partial x}$ is extremely sparse. The maximum row cardinality is four because the forward system does not contain any constraints with more than four variables. Therefore, equation (25) can be solved in $O(n)$ time. If the maximum in degree of the network is d_{in}, the maximum column cardinality is $2+|\mathcal{C}|(1+d_{\text{in}})$, as explained in section 4.4 of [18]. Assuming that d_{in} is a small constant, the multiplication step in equation (26) takes $O(m|\mathcal{C}|)$ time. This leads to a total computation time of $O(n + m|\mathcal{C}|)$.

D. Adjoint equations

The adjoint equations are given by the system:

$$\frac{\partial J}{\partial x} + \lambda^T \frac{\partial H}{\partial x} = 0 \qquad \forall x \in \mathcal{X} \qquad (28)$$

$$\Rightarrow \frac{\partial J}{\partial x} + \sum_{x' \in \mathcal{X}} \lambda'_x \frac{\partial H_{x'}}{\partial x} = 0 \qquad (29)$$

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

Solving the adjoint system requires the following partial derivatives.

1) $\frac{\partial J}{\partial \rho_{i,c}(k)} = \begin{cases} L_i & \forall c \in \mathcal{C},\ \forall i \in \mathcal{A} \setminus \mathcal{S},\ \forall k \in [\![0, T]\!] \\ 0 & \text{otherwise} \end{cases}$

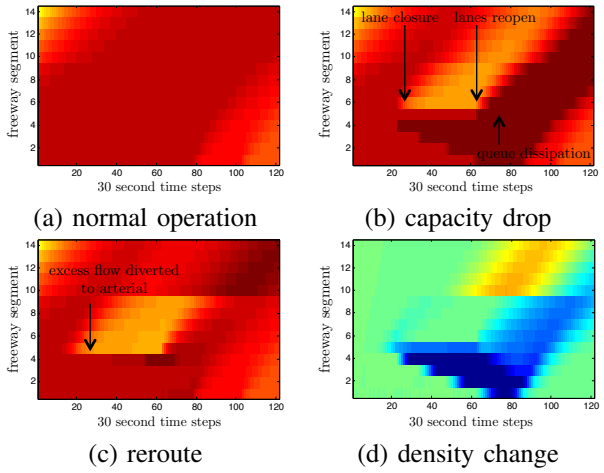
and 2) $\lambda^T \frac{\partial H}{\partial x}$, which is non-trivial to compute.

Sections 4.4 and 4.5 in [18] show how to compute all the individual partial derivatives that are required. Once they are computed, we can simply solve the system via backwards substitution since $\frac{\partial H}{\partial x}$ is lower triangular.

V. Numerical Results

A. Interstate 210 network

The experimental analysis was conducted on a 8 mile corridor of Interstate 210 in Arcadia, California with a parallel arterial route. The network has 24 cells corresponding to a discretization time step of 30 seconds. We consider a prototypical one hour time horizon during the morning commute[6]. The density profile of the freeway under the calibrated parameters and estimated boundary flows is shown in figure 1(a). We use the Rprop [22] algorithm as our gradient descent technique. All the experiments were run on a 1.8 GHz Intel Core i5 dual-core processor with 8GB of RAM.



(a) normal operation (b) capacity drop

(c) reroute (d) density change

Fig. 1: The density evolution along the 14 freeway road links with; (a) no incident, (b) a two lane capacity drop from minutes 10-30 at link 5, (c) flow rerouted to the arterial, and (d) the density difference between the incident profiles with and without rerouting.

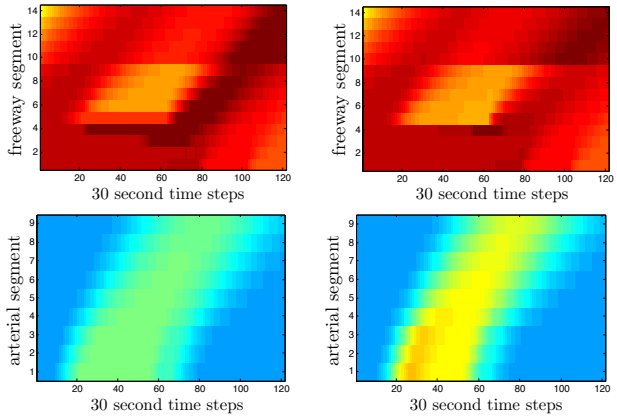


Fig. 2: A comparison of the density evolution with; (a) only 40% of the arterial capacity utilized for rerouting vehicles and (b) the entire arterial capacity utilized for rerouting vehicles.

We analyze the behavior of the freeway corridor in the event of a capacity drop caused by some incident. We

[6]The demand data and model parameters were obtained from the Connected Corridors project at UC Berkeley.

assume that the capacity drop occurs at the fifth freeway road segment 10 minutes into the simulation and that it lasts for 20 minutes, as illustrated in figure 1 (b). The freeway capacity at segment five will be assumed to be reduced by half during this period, corresponding to a closure of two lanes (out of four) at the location of the incident. Figure 1 shows the density profile corresponding to; (a) normal operation with capacity drop, (b) a capacity drop due to a two lane closure during the incident with no traffic diversion, (c) the same capacity drop with traffic being diverted to the parallel arterial, and (d) the change in the density profile due to the traffic diversion. As the figure shows, rerouting the excess flow to the parallel arterial eliminates the bottleneck during the incident and improves the throughput of the freeway corridor. In this example, the parallel arterial is assumed to prioritize vehicles being routed from the freeway and the full arterial capacity is used for this purpose. However, in certain situations, municipalities may want to allocate some capacity of the parallel arterial for local traffic. In this case, the optimizer can be limited to only use a certain fraction of the capacity of the parallel arterial. Figure 2 shows the density evolution when the arterial capacity for rerouting traffic is limited to 40% in comparison to full arterial utilization. See section 5 in [18] for more detailed numerical results.

VI. Conclusion

This article presents a model and optimization framework for solving the *System Optimal Dynamic Traffic Assignment problem with Partial Control* (SO-DTA-PC) for general networks with horizontal queuing dynamics. The model only requires full origin-destination (OD) information for the fraction of the agents that are controllable, with aggregate split ratios being sufficient for the non-controllable (selfish) agents. We show that the sparsity pattern of the forward system allows us to compute the gradient of the system with linear computational complexity and memory with respect to the state space, using the discrete adjoint method. Finally, we apply this framework to find the optimal vehicles rerouting strategy in response to a capacity loss in the network, and show the congestions reductions that can be achieved.

A longer technical report with additional details available online for the reviewer's convenience at: `http://dx.doi.org/10.7922/G23X84KV`.

References

[1] D. K. Merchant and G. L. Nemhauser, "A Model and an Algorithm for the Dynamic Traffic Assignment Problems," *Transportation science*, vol. 12, no. 3, pp. 183–199, 1978.

[2] D. K. Merchant and G. L. Nemhauser, "Optimality conditions for a dynamic traffic assignment model," *Transportation Science*, vol. 12, no. 3, pp. 183–199, 1978.

[3] J. G. Wardrop, "Some theoretical aspects of road traffic research," *Proceedings of the Institution of Civil Engineers*, vol. 1, pp. 325–378, 1952.

[4] D. Braess, "Über ein Paradoxon aus der Verkehrsplanung," *Mathematical Methods of Operations Research*, vol. 12, no. 1, pp. 258–268, 1968.

[5] F. P. Kelly, "Network routing," *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, vol. 337, no. 1647, pp. 343–367, 1991.

[6] T. Roughgarden, "The Price of Anarchy is Independent of the Network," *Computer*, no. May, pp. 1–24, 2002.

[7] C. Daganzo, "The cell transmission model, part II: Network traffic," *Transportation Research Part B*, vol. 29, no. 2, pp. 79–93, 1995.

[8] A. Aswani and C. Tomlin, "Game-theoretic routing of GPS-assisted vehicles for energy efficiency," in *American Control Conference (ACC), 2011*, pp. 3375–3380, IEEE, 2011.

[9] W. Krichene, J. Reilly, S. Amin, and A. M. Bayen, "Stackelberg Routing on Parallel Networks with Horizontal Queues," *IEEE Transactions on Automatic Control (in review)*, 2013.

[10] A. K. Ziliaskopoulos, "A linear programming model for the single destination system optimum dynamic traffic assignment problem," *Transportation science*, vol. 34, no. 1, pp. 37–49, 2000.

[11] M. Carey, "Nonconvexity of the dynamic traffic assignment problem," *Transportation Research Part B: Methodological*, vol. 26, no. 2, pp. 127–133, 1992.

[12] M. J. Lighthill and G. B. Whitham, "On kinematic waves. II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.

[13] P. I. Richards, "Shock waves on the highway," *Operations research*, vol. 4, no. 1, pp. 42–51, 1956.

[14] M. B. Giles and N. A. Pierce, "An introduction to the adjoint approach to design," *Flow, Turbulence and Combustion*, vol. 65, no. 3, pp. 393–415, 2000.

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

[16] S. K. Godunov, "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics," *Matematicheskii Sbornik*, vol. 89, no. 3, pp. 271–306, 1959.

[17] J. Reilly, M. L. D. Monache, S. Samaranayake, W. Krichene, P. Gaotin, and A. Bayen, "An efficient method for coordinated ramp metering using the discrete adjoint method," *Journal of Optimization Theory and Applications, in review*, 2013.

[18] S. Samaranayake, W. Krichene, J. Reilly, M. L. Delle Monache, P. Gaotin, and A. Bayen, "System Optimal Dynamic Traffic Assignment with Partial Compliance (SO-DTA-PC)," *Technical report*, 2014. `http://dx.doi.org/10.7922/G23X84KV`.

[19] G. Dervisoglu, G. Gomes, J. Kwon, R. Horowitz, and P. Varaiya, "Automatic calibration of the fundamental diagram and empirical observations on capacity," in *Transportation Research Board 88th Annual Meeting*, no. 09-3159, 2009.

[20] A. Wachter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*. 2005.

[21] A. C. Duffy, "An Introduction to Gradient Computation by the Discrete Adjoint Method," tech. rep., Florida State University, 2009.

[22] M. Riedmiller and H. Braun, "Rprop-a fast adaptive learning algorithm," in *Proceedings of the International Symposium on Computer and Information Science VII), Universitat*, Citeseer, 1992.