Gramian-Based Optimization for the Analysis and Control of Traffic Networks

Gianluca Bianchin, *Student Member, IEEE*, and Fabio Pasqualetti, *Member, IEEE*

***Abstract*— This paper proposes a simplified version of classical models for urban traffic networks and studies the problem of optimizing the network overall efficiency by controlling the signalized intersections. Differently from traditional approaches to control traffic signaling, the tractability of our framework allows us to effectively model large-scale interconnections and enables the design of critical parameters while considering network-wide measures of efficiency. Motivated by the increasing availability of real-time high-resolution traffic data, we cast an optimization problem that formalizes the goal of optimizing vehicle evacuation by controlling the durations of green lights at the intersections under the current congestion conditions. Our framework allows us to relate efficiency objectives with the optimization of a metric of controllability of the associated dynamical network. We then provide a technique to efficiently solve the optimization by parallelizing the computation among a group of distributed agents. Last, we assess the benefits of the proposed modeling and optimization framework through macroscopic and microscopic simulations on daily commute scenarios for the urban interconnection of Manhattan, NY, USA.**

***Index Terms*— Traffic control, signalized intersections, traffic networks, distributed optimization, controllability.**

I. Introduction

EFFECTIVE control of transportation systems is at the core of the smart city paradigm, and has the potential for improving efficiency and reliability of urban mobility. Modern urban transportation architectures comprise two fundamental components: traffic intersections and interconnecting roads. Intersections connect and regulate conflicting traffic flows among adjacent roads, and their effective control can sensibly improve travel time and prevent congestion. Congestion is the result of networks operating close to their capacity, and often leads to degraded throughput and increased travel time.

The increasing availability of sensors for vehicle detection and flow estimation, combined with modern communication capabilities (e.g. vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication), have inspired the development of infrastructure control algorithms that are adaptive [1], that is, policies that adjust the operation of the system based on the current traffic conditions. Nevertheless, the remarkable complexity of modern urban transportation infrastructures has recently promoted the diffusion of control policies that are distributed, that is, algorithms that adapt the operation of individual network components based on the partial knowledge of the current network state and dynamics. The lack of a global network model, capable of capturing the interactions between spatially-distributed components and capable of modeling all the relevant network dynamics, often results in suboptimal performance [2]. In this paper, we propose a simplified model to capture the time and spatial relationships between traffic flows in urban traffic networks in certain regimes. The model represents a tradeoff between accuracy and complexity, and sets out as a tractable framework to study efficiency and reliability of this class of dynamical systems. The proposed dynamical framework is employed in this work for the control of signalized traffic intersections.

Related Work: The design of feedback policies for the control of traffic infrastructures is an intensively-studied topic, and the available techniques can mainly be divided into three categories: routing policies, flow control, and intersections control. Routing policies rely on game-theoretic models to capture the behavior of the drivers and to influence their turning preferences in order to optimize congestion objectives, and have been studied both in a centralized [3] and distributed [4] framework. Flow control uses a combination of speed limits and gating techniques to regulate the road flows and network inflows, respectively [5], [6]. Conversely, intersection control refers to the design of the scheduling of the (automated) intersections so that the flow through intersections is maximized, and can be achieved: (i) by controlling the signaling sequence and offset, and/or (ii) by designing the durations of the signaling phases. The control of signaling offsets typically aims at tuning the synchronization of green lights between adjacent intersections in order to produce green-wave effects [7], and consists of solving a group of optimization problems that take into account certain subparts of the infrastructure, while minimizing metrics such as the number of stops experienced by the vehicles. In contrast, the durations of green times at intersections affects the average behavior of the traffic flows in the network, and plays a significant role in the efficiency of large-scale networks [2].

Widely-used distributed signaling control programs include SCOOT [8], RHODES [9], OPAC [10], SCATS [11], and emerge as the most common techniques currently employed in major cities. The sub-optimal performance of these methods has recently motivated the development of Max-Pressure techniques [1]. The Max-Pressure controller is based on a store-and-forward model, where queues at intersections

Manuscript received October 24, 2018; revised March 26, 2019; accepted May 31, 2019. This work was supported in part by the Army Research Office (ARO) under Grant 71603NSYIP and in part by the NSF under Grant CNS1646641. The Associate Editor for this paper was H. Jula. *(Corresponding author: Gianluca Bianchin.)*

The authors are with the Department of Mechanical Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: gianluca@engr.ucr.edul; fabiopas@engr.ucr.edu).

Digital Object Identifier 10.1109/TITS.2019.2922900

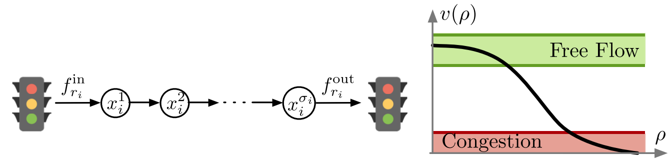
have unlimited capacity and, under this assumption, Max-Pressure is guaranteed to maximize the network throughput by stabilizing the network. Centralized policies require higher modeling efforts but, in general, have better performance guarantees [12]. Among the centralized policies, the Traffic-Responsive Urban Control [2] has received considerable interest for its simplicity and good performance. Based on a store-and-forward modeling paradigm, the method consists in minimizing the network queue lengths through a linear-quadratic regulator that uses a relaxation of the physical constraints to abide with the high complexity. Variations of these techniques to incorporate physical constraints have been studied in [13], [14].

The tremendous complexity of urban traffic networks has recently motivated the adoption of model-free control methods that rely only on historical data [15] and, concurrently, the development of simplified models to deal with the switching nature of the traffic signals [16]. However, the highly-nonlinear behavior of these dynamical systems still limits our capability to consider adequate optimization and prediction horizons, and the development of tractable models capable of capturing all the relevant network dynamics is still an open problem.

Contribution: Motivated by the considerable complexity of urban transportation systems, in this paper we propose a simplified framework to capture the behavior of traffic networks operating in free-flow regimes with arbitrary travel speeds. In this model, each road is modeled through multiple state variables, representing the spatial evolution of traffic densities within the road. This assumption allows us to capture the non-uniform spatial displacement of traffic within each road, and to construct a simplified network model that results in a more- tractable framework for optimization.

We employ the proposed model to design the durations of the green times at the intersections, and we relate congestion objectives with the optimization of a metric of controllability of the dynamical system associated with the traffic network. To the best of the authors' knowledge, this work represents a novel, computationally-tractable, method to perform network-wide optimization of the green-splits durations at intersections. We provide conditions that guarantee stability of the system, and we characterize the performance of the control policy in relation to the network congestion. We use the concept of smoothed spectral abscissa [17] to solve the optimization, and we demonstrate the benefits of our methods through a microscopic simulation on the urban interconnection of Manhattan, NY. We characterize the complexity of our algorithms, and propose a method to parallelize the computation so that it can be solved efficiently by a group of cooperating distributed agents. Our results and simulations suggest that the increased system performance obtained by our control method justifies the increment in complexity deriving from the adoption of a global system description.

Organization: The rest of this paper is organized as follows. Section II illustrates our model of traffic network, and formalizes the problem of designing the durations of the green times at the intersections with the goal of optimizing



(a) (b)

Fig. 1. (a) The portion of road comprised between two signalized intersections is modeled with a set of σ_i variables. (b) The almost-flat behavior in regimes of free-flow or congestion motivates our approximation $\gamma(\rho_i) \approx \gamma_i$.

vehicle evacuation. The section includes a discussion of the benefits in adopting a simplified model, and presents a comparison with more-established macroscopic models. Section III illustrates the proposed centralized approach to numerically solve the optimization problem, while Section IV presents a technique to parallelize the computation among a set of distributed agents for more efficient computation. Section V is devoted to macroscopic and microscopic simulations to validate our modeling assumptions and optimization techniques. Finally, Section VI concludes the paper.

II. Dynamical Model of Traffic Networks and Problem Formulation

We model urban traffic networks as a group of one-way roads interconnected through signalized intersections. Within each road, vehicles move at uniform velocity, while traffic flows are exchanged between adjacent roads by means of the signalized intersection connecting them. In this section, we discuss a concise dynamical model for traffic networks in certain regimes, that will be employed for the analysis.

A. Model of Road and Traffic Flow

Let $\mathcal{N} = (\mathcal{R}, \mathcal{I})$ denote a traffic network with roads $\mathcal{R} = \{r_1, \ldots, r_{n_r}\}$ and intersections $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_{n_\mathcal{I}}\}$. Each element in the set $\mathcal{R}$ models a one way link interconnecting two signalized intersections, whereas intersections regulate conflicting flows of traffic among adjacent roads (see Section II-B). We assume that exogenous inflows enter the network at (source) roads $\mathcal{S} \subseteq \mathcal{R}$ and, similarly, vehicles exit the network at (destination) roads $\mathcal{D} \subseteq \mathcal{R}$, with $\mathcal{S} \cap \mathcal{D} = \varnothing$. The following standard connectivity assumption ensures that vehicles are allowed to leave the network.

Assumption 1: For every road $r_i \in \mathcal{R}$ there exists at least one path in $\mathcal{N}$ from r_i to a road $r_j \in \mathcal{D}$.

We denote by $\ell_i \in \mathbb{R}_{>0}$ the length of road r_i, and we model each road r_i by discretizing it into $\sigma_i = \lceil \ell_i / h \rceil$ segments of uniform length $h \in \mathbb{R}_{\geq 0}$ (see Fig. 1). We denote by $x_i^k \in \mathbb{R}$ the traffic density associated with the k-th segment of road r_i, $k \in \{1, \ldots, \sigma_i\}$. We assume that inflows of vehicles $f_{r_i}^{\text{in}}$ enter the road in correspondence of its upstream segment (i.e. $k = 1$); accordingly, outflows $f_{r_i}^{\text{out}}$ leave the road in correspondence of its downstream segment (i.e. $k = \sigma_i$). We approximate the relationship between traffic flows and densities by assuming that vehicles move from upstream to downstream with uniform velocity γ_i. Then, the dynamics of

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

the road state $x_i = [x_i^1 \cdots x_i^{\sigma_i}]^\mathsf{T}$ are described by:

$$\begin{bmatrix} \dot{x}_i^1 \\ \dot{x}_i^2 \\ \vdots \\ \dot{x}_i^{\sigma_i} \end{bmatrix} = \frac{\gamma_i}{h} \underbrace{\begin{bmatrix} -1 & & & \\ 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}}_{D_i} \begin{bmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^{\sigma_i} \end{bmatrix} + \begin{bmatrix} f_{r_i}^{\text{in}} \\ 0 \\ \vdots \\ -f_{r_i}^{\text{out}} \end{bmatrix}. \quad (1)$$

Differently from more-established network models where a single state variable is associated to a uniform road segment (e.g., [1], [18]), the choice of a constant space-discretization step allows us to capture the fact that the density of vehicles may not be uniform along the road.

Remark 1 (Equivalence With Hydrodynamic Models): The dynamical model (1) *derives from the mass-conservation continuity equation [19] in certain traffic regimes, as we explain next. Let $\rho_i = \rho_i(s,t) \geq 0$ denote the (continuous) density of vehicles within road r_i at the spatial coordinate $s \in [0, \ell_i]$ and time $t \in \mathbb{R}_{\geq 0}$. Let $f_i = f_i(s,t) \geq 0$ denote the (continuous) flow of vehicles along the road, and let traffic densities and traffic flows follow the hydrodynamic relation*

$$\frac{\partial \rho_i}{\partial t} + \frac{\partial f_i}{\partial s} = 0.$$

We complement the above equation with the Lighthill-Whitham-Richards relation $f_i = f_i(\rho_i)$, where flows instantaneously change with the density. Then, we include the speed-density fundamental relationship $f_i = \rho_i \upsilon(\rho_i)$, where $\upsilon : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ represents the speed of flow, to obtain

$$\frac{\partial \rho_i}{\partial t} + \left(\upsilon(\rho_i) + \rho_i \frac{d\ \upsilon(\rho_i)}{d\rho_i} \right) \frac{\partial \rho_i}{\partial s} = 0.$$

Solutions to the above relation are kinematic waves [20], moving at speed $\gamma(\rho_i) = \upsilon(\rho_i) + \rho_i \frac{d\upsilon(\rho_i)}{d\rho_i}$. We consider regimes where the speed of the kinematic wave can be approximated as $\gamma(\rho_i) \approx \gamma_i$. As illustrated in Fig. 1(b), this approximation is accurate in regimes of free flow or congestion, characterized by $\frac{d\upsilon(\rho_i)}{d\rho_i} \approx 0$. By letting γ_i denote the average speed of flow, the approximated continuity equation reads

$$\frac{\partial \rho_i}{\partial t} + \gamma_i \frac{\partial \rho_i}{\partial s} = 0.$$

We then discretize in space the above relation by defining the discrete spatial coordinate

$$s_k = kh, \quad k \in \{0, \ldots, \sigma_i\},$$

and by replacing the partial derivative with respect to s with the difference quotient

$$\frac{\partial \rho_i(s_k,t)}{\partial t} = -\gamma_i \frac{\rho_i(s_k,t) - \rho_i(s_{k-1},t)}{h}.$$

This discretization leads to the dynamical model (1)*, by introducing the boundaries inflows $f_{r_i}^{\text{in}}$ and outflows $f_{r_i}^{\text{out}}$, and by replacing $\rho_i(s_k,t)$ with the compact notation x_i^k.* □

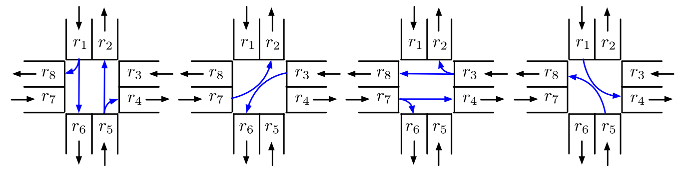


Fig. 2. Typical set of phases at a four-ways intersection.

B. Model of Intersection and Interconnection Flow

Signalized intersections alternate the right-of-way of vehicles to coordinate and secure conflicting flows between adjacent roads. Every signalized intersection $\mathcal{I}_j \in \mathcal{I}$, $j \in \{1, \ldots, n_\mathcal{I}\}$, is modeled as a set $\mathcal{I}_j \subseteq \mathcal{R} \times \mathcal{R}$, consisting of all allowed movements between the intersecting roads. For road $r_i \in \mathcal{R}$, let $\mathcal{I}_{\text{in}}^{r_i}$ denote the (unique) intersection at road upstream; similarly, let $\mathcal{I}_{\text{out}}^{r_i}$ denote the (unique) intersection at road downstream. We model the switching behavior of a signalized intersection through the green split function $s : \mathcal{R} \times \mathcal{R} \times \mathbb{R}_{\geq 0} \to \{0, 1\}$ that assumes Boolean values 1 (green phase) or 0 (red phase), and let the interconnection flows be

$$\begin{aligned} f_{r_i}^{\text{in}} &= \sum_{(r_i,r_k) \in \mathcal{I}_{\text{in}}^{r_i}} s(r_i, r_k, t)\, f(r_i, r_k) + u_{r_i}, \\ f_{r_i}^{\text{out}} &= \sum_{(r_k,r_i) \in \mathcal{I}_{\text{out}}^{r_i}} s(r_k, r_i, t)\, f(r_k, r_i) + w_{r_i}, \end{aligned} \quad (2)$$

where $f : \mathcal{R} \times \mathcal{R} \to \mathbb{R}_{\geq 0}$ denotes the intersection transmission rate. We remark that the notation $f(r_i, r_k)$ represents the transmission rate from road r_k to r_i and, similarly, $s(r_i, r_k, t)$ denotes the green split function that controls traffic flows from r_k and directed to r_i. We note that equation (2) incorporates the exogenous inflows and outflows to each road (flows of traffic that are not originated or merge to modeled intersections or roads) through the terms $u_{r_i} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and $w_{r_i} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, respectively. We note that $u_{r_i} \neq 0$ if and only if r_i is a source road (that is, $r_i \in \mathcal{S}$), and $w_{r_i} \neq 0$ if and only if r_i is a destination road (that is, $r_i \in \mathcal{D}$).

Example 1 (Intersections and Scheduling Functions): Consider the four-ways intersection $\mathcal{I}_1$ illustrated in Fig. 2. The intersection is modeled through the set of allowed movements

$$\begin{aligned} \mathcal{I}_1 = \{&(r_1, r_6), (r_1, r_8), (r_5, r_2), (r_5, r_4), (r_7, r_2), (r_3, r_6), \\ &(r_3, r_8), (r_3, r_2), (r_7, r_4), (r_7, r_6), (r_5, r_8), (r_1, r_4)\}. \end{aligned}$$

Allowed movements at a certain intersections are typically grouped into sets of phases, where each phase represents a set of movements that can occur simultaneously. For $\mathcal{I}_1$, a typical set of phases is $\{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4\}$, where

$$\begin{aligned} \mathcal{P}_1 &= \{(r_1, r_6), (r_1, r_8), (r_5, r_2), (r_5, r_4)\}, \\ \mathcal{P}_2 &= \{(r_7, r_2), (r_3, r_6)\}, \\ \mathcal{P}_3 &= \{(r_3, r_8), (r_3, r_2), (r_7, r_4), (r_7, r_6)\}, \\ \mathcal{P}_4 &= \{(r_5, r_8), (r_1, r_4)\}. \end{aligned}$$

The green split function alternates the set of available phases within the cycle time $T \in \mathbb{R}_{>0}$, that is, for given scalars t_0, t_1, t_2, t_3, t_4, with $0 = t_0 \leq t_1 \leq t_2 \leq t_3 \leq t_4 = T$, denoting *the switching instants, the green split function is*

$$s(r_i, r_k, t) = \begin{cases} 1 & \text{if } (r_i, r_k) \in \mathcal{P}_j \text{ and } t \in [t_{j-1}, t_j), \\ 0 & \text{otherwise}, \end{cases}$$

where $j \in \{1, \ldots, 4\}$. □

We model the transmission rate during a green-light phase as a function proportional to the density of vehicles "waiting" in the downstream section of the road, that is,

$$f(r_i, r_k) = c(r_i, r_k) x_k^{\sigma_k}, \tag{3}$$

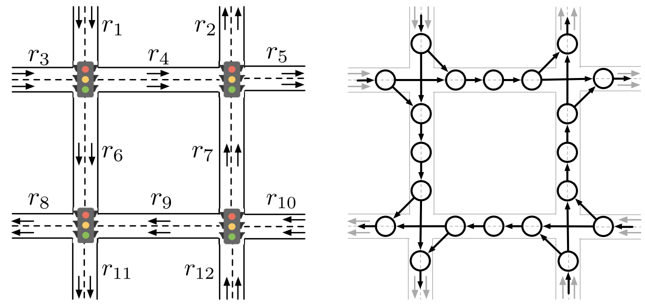
where $c : \mathcal{R} \times \mathcal{R} \to \mathbb{R}_{\geq 0}$. In particular, $c(r_i, r_k)$ incorporates the turning preferences of the drivers when decomposing $c(r_i, r_k) = \varphi(r_i, r_k)\phi(r_i, r_k)$, where $\varphi : \mathcal{R} \times \mathcal{R} \to [0, 1]$ represents the average routing ratio of vehicles entering road r_i after exiting r_k, $\sum_i \varphi(r_i, r_k) = 1$, and $\phi : \mathcal{R} \times \mathcal{R} \to \mathbb{R}_{\geq 0}$ captures the speed of the outflow from the dedicated turn lane.

Differently from traditional traffic network models where a single state variable is typically used to model a uniform road segment [1], [18], our model associates multiple state variables to each road segment interconnecting two signalized intersections. This approach allows us to capture the fact that the density of vehicles may not be uniform along each link, and to model the outflows during a green-light phase as functions that depend only on the state of the section of road that is located in the proximity of the intersection. The precision of the illustrated model is demonstrated through a set of microscopic simulations in [21] for a small scale network.

Remark 2 (Model Validity and Limitations): Two main limitations can be identified in the simplified modeling settings considered in this work with respect to more comprehensive models, such as [18]. First, our model assumes a constant speed of flow along each road segment connecting two signalized intersections. Second, the linear approximation does not allow to limit the inflow to a certain road when that road is congested, which corresponds to the assumption that roads have infinite capacity. We remark, however, that these two phenomena can be captured by appropriately tuning the parameters γ_i *and* $\phi(r_i, r_k)$*, respectively, when these situations occur. Thus, if the network conditions do not change significantly fast with respect to the network dynamics, one can tune the parameters* γ_i *and* $\phi(r_i, r_k)$ *and occasionally re-update the model to capture the current network conditions. We anticipate that, although this approach implies that the model is accurate only in the current network regime, the approach is well-suited for the receding horizon technique that will be later adopted in this paper (see Section II-D).* □

C. Switching and Time-Invariant Traffic Network Dynamics

Individual road dynamics can be combined into a joint network model that captures the interactions among all modeled routes and intersections. To this aim, we adopt an approach similar to [14], and assume that exogenous outflows are proportional to the number of vehicles in the road, that is, $w_{r_i} = \bar{w}_{r_i} x_i^{\sigma_i}$, $\bar{w}_{r_i} \in [0, 1]$. By combining Equations (1), (2) and (3), we obtain



(a) (b)

Fig. 3. Network model associated with a traffic network composed of $n_{\mathcal{I}} = 4$ intersections and $n_{\rm r} = 12$ roads. Each road is associated with a set of states that represent the density of the cells within the roads.

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n_{\rm r}} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n_{\rm r}} \\ A_{21} & A_{22} & \ddots & A_{2n_{\rm r}} \\ \vdots & \ddots & \ddots & \vdots \\ A_{n_{\rm r}1} & A_{n_{\rm r}2} & \cdots & A_{n_{\rm r}n_{\rm r}} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_{\rm r}} \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} I_{n_1} & 0 & \cdots & 0 \\ 0 & I_{n_2} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & I_{n_{\rm r}} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n_{\rm r}} \end{bmatrix}}_{u}, \tag{4}$$

where $A \in \mathbb{R}^{n \times n}$, $n = \sum_{i=1}^{n_{\rm r}} \sigma_i$ is the overall number of states, u derives from (2), and

$$A_{ik} = \begin{cases} s(r_i, r_k, t) c(r_i, r_k) e_1 e_{\sigma_k}^{\mathsf{T}}, & \text{if } i \neq k, \\ D_i - \left(\sum_\ell s(r_\ell, r_i, t) c(r_\ell, r_i) + \bar{w}_{r_i}\right) e_{\sigma_i} e_{\sigma_i}^{\mathsf{T}}, & \text{if } i = k, \end{cases}$$

where $e_i = [0 \ldots 1 \ldots 0]^{\mathsf{T}}$ is a vector with a single nonzero entry with value 1 in position i and of appropriate dimension.

We note that the matrix A in (4) is typically sparse because not all roads are adjacent in the interconnection, and its sparsity pattern varies over time as determined by the splits $s(r_i, r_k, t)$. Thus, the network model (4) is a linear switching system, where the switching signals are the split functions.

Example 2 (Traffic Network Interconnection): Consider the network illustrated in Fig. 3, with $\mathcal{R} = \{r_1, \ldots, r_{12}\}$ *and* $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_4\}$*. The network comprises four destination roads* $\mathcal{D} = \{r_2, r_5, r_8, r_{11}\}$ *(*$\bar{w}_{r_i} = 1$ *for all* $r_i \in \mathcal{D}$*, and* $\bar{w}_{r_i} = 0$ *otherwise), and four source roads (*$\mathcal{S} = \{r_1, r_3, r_{10}, r_{12}\}$*, with* $u_{r_i} \neq 0$ *only if* $r_i \in \mathcal{S}$*). Let* $\ell_i / h = 3$ *and* $\gamma_i / h = 3$ *for all* $i \in \{1, \ldots, n_r\}$*. Then, the matrices in* (4) *read as*

$$A_{ii} = \begin{bmatrix} -1 & & \\ 1 & -1 & \\ & 1 & -\left(\sum_\ell s(r_\ell, r_i, t) c(r_\ell, r_i) + \bar{w}_{r_i}\right) \end{bmatrix},$$

$$A_{ij} = \begin{bmatrix} 0 & 0 & s(r_i, r_j, t) c(r_i, r_j) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

for all $i, j \in \{1, \ldots, n_r\}$*. Notice that* $s(r_i, r_j, t) = 0$ *for all times if* $(r_j, r_k) \notin \mathcal{I}_k$ *for all* $k \in \{1, \ldots, n_{\mathcal{I}}\}$*.* □

Next, we make the classical assumption that scheduling functions are periodic, with period $T \in \mathbb{R}_{>0}$. That is, for all $(r_i, r_k) \in \mathcal{I}_j$, $j \in \{1, \ldots, n_{\mathcal{I}}\}$, and for all times t:

$$s(r_i, r_k, t) = s(r_i, r_k, t + T).$$

Let $\mathcal{T} = \{\tau_1, \ldots, \tau_m\}$ denote the set of time instants when a scheduling function changes its value, that is,

$$\mathcal{T} = \{\tau \in [0, T] \; : \; \exists (r_i, r_k) \in \mathcal{I}, \\ \lim_{t \to \tau^-} s(r_i, r_k, t) \neq \lim_{t \to \tau^+} s(r_i, r_k, t)\}.$$

Notice that the matrix A in (4) remains constant between consecutive time instants τ_{i-1} and τ_i. We denote each constant matrix by A_i, and refer to it as to the i-th *network mode*. Further, let $d_i = \tau_i - \tau_{i-1}$, with $i \in \{1, \ldots, m\}$ and $\tau_0 = 0$, denote the duration of the i-th network mode. We employ a state-space averaging technique [22] and define a linear, time-invariant, approximation of the switching network model (4):

$$\dot{x}_{\text{av}} = A_{\text{av}} x_{\text{av}} + B u_{\text{av}}, \tag{5}$$

where $A_{\text{av}} = \frac{1}{T} \sum_{i=1}^{m} d_i A_i$, and $u_{\text{av}} = [u_{\text{av},1} \ldots u_{\text{av},n_r}]$, $u_{\text{av},i} = (1/T) \int_0^T u_i(\tau) \, d\tau$. We note that the averaging technique preserves the sparsity pattern of the network, that is, $A_{\text{av}}(i, j) \neq 0$ if and only if $A_k(i, j) \neq 0$ for some k.

In general, the approximation of the behavior of the switching system (4) with the average dynamics (5) is accurate if the operating period T is short in comparison to the underlying system dynamics. Under suitable technical assumptions, a bound on the deviation of average models with respect to the network instantaneous state has been characterized in [22]. In particular, the bound becomes tighter for decreasing values of T and increasing values of road lengths. A numerical validation of the averaging technique and its validity in relation to T is discussed in Section V (see Fig. 5).

D. Problem Formulation

In this paper, we consider the average model (5) and focus on the problem of designing the durations of the green split functions so that a measure of network efficiency is optimized. Motivated by the relationship $A_{\text{av}} = \frac{1}{T} \sum_{i=1}^{m} d_i A_i$ (see (5)), the average model allows us to design the durations of the network modes, rather than their exact sequence. This approach motivates the adoption of a two-stage optimization process. First, the durations of the modes is optimized by considering a global model that captures the dynamics of the entire interconnection. Second, offset optimization techniques (e.g. [7]) can be employed to decide the specific sequence of phases given the durations of the splits, and by considering reduced or local interconnection models. This paper is devoted to the former. To formalize our optimization problem, we denote by y_{av} the vector of the queue lengths originated by the signalized intersections, and model y_{av} as the density of vehicles "waiting" at the downstream section of each road:

$$y_{\text{av}} = C_{\text{av}} x_{\text{av}}, \quad C_{\text{av}} = \begin{bmatrix} e_{\sigma_1}^{\mathsf{T}} & \ldots & 0 \\ \vdots & \ddots & \\ 0 & \ldots & e_{\sigma_{n_r}}^{\mathsf{T}} \end{bmatrix}. \tag{6}$$

We assume the network is initially at a certain initial state x_0, and focus on the problem of optimally designing the mode durations $\{d_1, \ldots d_m\}$ that minimize the $\mathcal{H}_2$-norm of the vector of queue lengths y_{av}, formalized as follows

$$\begin{aligned} \min_{d_1, \ldots, d_m} \quad & \int_0^\infty \|y_{\text{av}}\|_2^2 \; dt, \\ \text{subject to} \quad & \dot{x}_{\text{av}} = A_{\text{av}} x_{\text{av}}, && \text{(7a)} \\ & y_{\text{av}} = C_{\text{av}} x_{\text{av}}, && \text{(7b)} \\ & x_{\text{av}}(0) = x_0, && \text{(7c)} \\ & A_{\text{av}} = \frac{1}{T} \left(d_1 A_1 + \cdots + d_m A_m \right), && \text{(7d)} \\ & T = d_1 + \cdots + d_m, && \text{(7e)} \\ & d_i \geq 0 \quad i \in \{1, \ldots, m\}. && \text{(7f)} \end{aligned}$$

Loosely speaking, the optimization problem (7) seeks for an optimal set of split durations that minimize the $\mathcal{L}_2$-norm of the impulse-response of the system to the initial conditions. Thus, similarly to [6], our framework considers the "cool down" period, where exogenous inflows and outflows are not known a priori, and the goal is to evacuate the network as fast as possible where the final condition is an empty system. In order to take into account for the model inaccuracies due to linearization and time-averaging, the matrix A_{av} and the initial state x_0 shall be updated when the network conditions have significantly changed, and the solution to (7) shall be recomputed with the updated parameters. In particular, we denote by T_{update} the time interval between two updates, and note that T_{update} is a fundamental design parameter that should be accurately chosen. Finally, we note that constraint (7e) implies that for any solution to (7) there exists a set of split with the selected green durations, and thus ensures feasibility of the solutions.

III. Design of Optimal Network Mode Durations

In this section we propose a method to determine solutions to the optimization problem (7). The approach we discuss is centralized, namely, it requires full knowledge of the network dynamics and initial state. An extension of the framework to fit a distributed implementations is proposed in Section IV. Our approach consists in rewriting the cost function in (7) in terms of the controllability Gramian of the associated dynamical system, and is formalized next.

Lemma 1 (Controllability Gramian Cost Function): Let

$$\mathcal{W}(A_{\text{av}}, x_0) = \int_0^\infty e^{A_{\text{av}} t} x_0 x_0^{\mathsf{T}} e^{A_{\text{av}}^{\mathsf{T}} t} \;\; dt.$$

The following minimization problem is equivalent to (7)*:*

$$\begin{aligned} \min_{d_1, \ldots, d_m} \quad & \text{Trace} \left(C_{\text{av}} \; \mathcal{W}(A_{\text{av}}, x_0) C_{\text{av}}^{\mathsf{T}} \right), \\ \textit{subject to} \;\; & A_{\text{av}} = \frac{1}{T} \left(d_1 A_1 + \cdots + d_m A_m \right), \\ & T = d_1 + \cdots + d_m, \\ & d_i \geq 0, \quad i \in \{1, \ldots, m\}. \end{aligned} \tag{8}$$

Proof: By incorporating (7a), (7b), and (7c) into the cost function of optimization problem (7), we can rewrite:

$$\begin{aligned}\int_0^\infty \|y_{\text{av}}\|_2^2\, dt &= \int_0^\infty x_0^\top e^{A_{\text{av}}^\top t} C_{\text{av}}^\top C_{\text{av}} e^{A_{\text{av}} t} x_0 dt \\ &= \text{Trace}\left(\int_0^\infty x_0^\top e^{A_{\text{av}}^\top t} C_{\text{av}}^\top C_{\text{av}} e^{A_{\text{av}} t} x_0\, dt\right) \\ &= \int_0^\infty \text{Trace}\left(C_{\text{av}} e^{A_{\text{av}} t} x_0 x_0^\top e^{A_{\text{av}}^\top t} C_{\text{av}}^\top\right) dt \\ &= \text{Trace}\left(C_{\text{av}} \int_0^\infty e^{A_{\text{av}} t} x_0 x_0^\top e^{A_{\text{av}}^\top t} dt C_{\text{av}}^\top\right),\end{aligned}$$

from which the claimed statement follows. □

We now use the above result to characterize the stability of the proposed control policy. To this aim, we note that the cost function in (7) is finite only if the choice of parameters $\{d_1, \ldots, d_m\}$ leads to a matrix A_{av} that is Hurwitz-stable. Requiring Hurwitz stability of A_{av} corresponds to imposing that all real parts of the eigenvalues of A_{av} are strictly negative. Formally, we require $\alpha(A_{\text{av}}) < 0$, where $\alpha(A_{\text{av}}) := \sup\{\Re(s) : s \in \mathbb{C}, \det(sI - A_{\text{av}}) = 0\}$ denotes the spectral abscissa of A_{av}. The following result proves stability of the system under optimal green time durations.

Theorem 1 (Stability of Optimal Solutions): Let Assumption 1 be satisfied and let $s(r_i, r_k, \bar{t}) \neq 0$ for all $(r_i, r_k) \in \mathcal{I}$, and for some $\bar{t} \in [0, T]$. Then,

$$\alpha(A_{\text{av}}) < 0.$$

Proof: From the structure of (4) and from the assumption $s(r_i, r_k, \bar{t}) \neq 0$ follows that $A_{\text{av}}(i, i) < 0$ for all $i \in \{1, \ldots, n\}$, while $A_{\text{av}}(i, j) \geq 0$ for all $j \in \{1, \ldots, n\}$, $j \neq i$. Moreover, all columns of A_{av} have nonpositive sum. In particular, the columns corresponding to destination cells have strictly negative sum, that is, $\sum_{i=1}^n A_{\text{av}}(i, j) \leq 0$ for all $j \in \{1, \ldots, n\}$, and $\sum_{i=1}^n A_{\text{av}}(i, j) < 0$ for all j such that $r_j \in \mathcal{D}$. To show $\alpha(A_{\text{av}}) < 0$, we use the fact that destination cells in $\mathcal{D}$ have no departing edges, and re-order the states so that

$$A_{\text{av}} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix},$$

where $A_{22} \in \mathbb{R}^{n_d \times n_d}$, $n_d = |\mathcal{D}|$, is the submatrix that describes the dynamics of the destination cells, $A_{11} \in \mathbb{R}^{(n-n_d)\times(n-nd)}$, and $A_{21} \in \mathbb{R}^{n_d\times(n-nd)}$. The fact $\alpha(A_{22}) < 0$ immediately follows from (2). The stability of A_{11} follows from the connectivity assumption in the original network, and from the analysis of grounded Laplacian matrices (see e.g. [23, Theorem 1]). □

Next, we discuss a method to determine solutions to the optimization problem (8). Our technique relies on constructing a new optimization problem that constitutes a relaxation of (8), and that builds upon the concept of smoothed spectral abscissa. The smoothed spectral abscissa is a generalization of the spectral abscissa [17] and, for a dynamical system of the form (5)-(6), it is defined as the root $\tilde{\alpha} \in \mathbb{R}$ of the implicit equation

$$\text{Trace}\left(C_{\text{av}}\ \mathcal{W}(A_{\text{av}} - \tilde{\alpha} I, B)\ C_{\text{av}}^\top\right) = \epsilon^{-1}, \tag{9}$$

where $\epsilon \in \mathbb{R}_{\geq 0}$. It is worth noting that the root $\tilde{\alpha}$ is unique [17], and for fixed B and C_{av} it is a function of both ϵ and A_{av}. Formally, we shall denote $\tilde{\alpha} = \tilde{\alpha}(\epsilon, A_{\text{av}})$.

Remark 3 (Properties of the Smoothed Spectral Abscissa): For any $\epsilon > 0$, the smoothed spectral abscissa is an upper bound to $\alpha(A)$, and this bound becomes exact as $\epsilon \to 0$. To see this, we first observe that the integral $\int_0^\infty e^{(A_{\text{av}}-\tilde{\alpha} I)t} B B^\top e^{(A_{\text{av}}-\tilde{\alpha} I)^\top t} dt$ exists and is finite for any $\tilde{\alpha} > \alpha(A_{\text{av}})$, as the function $e^{(A_{\text{av}}-\tilde{\alpha} I)t}$ is bounded and convergent as $t \to +\infty$. On the other hand, for any $\tilde{\alpha} < \alpha(A_{\text{av}})$ the function $e^{(A_{\text{av}}-\tilde{\alpha} I)t}$ becomes unbounded for $t \to +\infty$ and the above integral is infinite. It follows that, the left-hand side of (9) is finite only if $\tilde{\alpha} > \alpha(A)$ or, in other words, for any finite ϵ, $\tilde{\alpha}$ satisfies $\tilde{\alpha} > \alpha(A)$. □

By letting $\tilde{\alpha} = 0$ in (9), we recast the optimization problem (8) in terms of the smoothed spectral abscissa as follows:

$$\begin{aligned} \min_{d_1,\ldots,d_m,\epsilon} \quad & \epsilon^{-1}, \\ \text{subject to} \quad & A_{\text{av}} = \frac{1}{T}(d_1 A_1 + \cdots + d_m A_m), \\ & T = d_1 + \cdots + d_m, \\ & d_i \geq 0, \quad i \in \{1, \ldots, m\}, \\ & \tilde{\alpha}(\epsilon, A_{\text{av}}) = 0, \end{aligned} \tag{10}$$

where the parameter ϵ is now an optimization variable. In what follows, we denote by $\{d_1^*, \ldots, d_m^*, \epsilon^*\}$ the value of the optimization parameters at optimality of (10). Problem (10) is a nonlinear optimization problem [17], because the optimization variables $\{d_1, \ldots, d_m\}$ and ϵ are related by means of the nonlinear equation (9).

For the solution of (10), we propose an iterative two-stages numerical optimization process. In the first stage, we fix the value of ϵ and seek for a choice of $\{d_1, \ldots d_m\}$ that leads to a smoothed spectral abscissa that is identically zero. In other words, we let $\epsilon = \bar{\epsilon}$ (fixed), and solve the following minimization problem:

$$\begin{aligned} \min_{d_1,\ldots,d_m} \quad & |\tilde{\alpha}(\bar{\epsilon}, A_{\text{av}})| \\ \text{subject to} \quad & A_{\text{av}} = \frac{1}{T}(d_1 A_1 + \cdots + d_m A_m), \\ & T = d_1 + \cdots + d_m, \\ & d_i \geq 0, \quad i \in \{1, \ldots, m\}. \end{aligned} \tag{11}$$

We note that every a solution to (11), namely $\bar{A}_{\text{av}}$, which satisfies $\tilde{\alpha}(\bar{\epsilon}, \bar{A}_{\text{av}}) = 0$, is a point in the feasible set of (10) that corresponds to a cost of congestion $\int_0^\infty \|y_{\text{av}}\|_2^2\, dt = 1/\bar{\epsilon}$.

In the second stage of the optimization, we perform a line-search over the parameter ϵ. In particular, the value of ϵ is iteratively increased until the minimizer ϵ^* is achieved. This approach is motivated by the fact that the optimizer of (11) with ϵ set to $\epsilon = \epsilon^*$ is $\{d_1^*, \ldots, d_m^*\}$, that is, the optimal solution to (10). Finally, the iterative process is concluded when $\tilde{\alpha}(\bar{\epsilon}, A_{\text{av}}) = 0$ is no longer achievable in (11).

The benefit of considering a two-stage optimization process and of solving (11) as opposed to (10) is that we can derive an expression for the gradient of the cost function $\tilde{\alpha}(\bar{\epsilon}, A_{\text{av}})$ with respect to the parameters $\{d_1, \ldots, d_m\}$. The derivation of the descent direction is the focus of the remaining part of

Algorithm 1: Centralized Solution to (7).

Input: Matrix C_{av}, vector x_0, scalars ξ, μ
Output: $\{d_1^*, \dots d_m^*, \epsilon^*\}$ solution to (7)
1 Initialize: $d^{(0)}$, $\bar{\epsilon} = 0$, $k = 1$
while $\tilde{\alpha}_{\bar{\epsilon}}^{(k)} = 0$ **do**
2   **repeat**
3   Compute $\tilde{\alpha}_{\bar{\epsilon}}^{(k)}$ by solving (9);
4   Solve for P and Q: $(A_{\text{av}}^{(k)} - \alpha_{\bar{\epsilon}}^{(k)} I)P + P(A_{\text{av}}^{(k)} - \alpha_{\bar{\epsilon}}^{(k)} I)^{\mathsf{T}} + x_0 x_0^{\mathsf{T}} = 0$; $(A_{\text{av}}^{(k)} - \alpha_{\bar{\epsilon}}^{(k)} I)^{\mathsf{T}} Q + Q(A_{\text{av}}^{(k)} - \alpha_{\bar{\epsilon}}^{(k)} I) + C_{\text{av}} C_{\text{av}}^{\mathsf{T}} = 0$;
5     $\nabla \leftarrow K^{\mathsf{T}} \left(\frac{QP}{\text{Trace}(QP)} \right)^{\text{vec}}$;
6   Compute projection matrix $\mathcal{P}^{(k)}$;
7     $d^{(k)} \leftarrow d^{(k)} - \mu \ \mathcal{P}^{(k)} \nabla$;
8     $A_{\text{av}}^{(k)} \leftarrow \frac{1}{T}(d_1 A_1 + \dots + d_m A_m)$;
9     $k \leftarrow k + 1$;
10   **until** $\mathcal{P}^{(k)} \nabla = 0$;
11   $\bar{\epsilon} \leftarrow \bar{\epsilon} + \xi$;
12 **end**
13 **return** d;

this section. In the remainder, with a slight abuse of notation, we use the compact form $\tilde{\alpha}(\bar{\epsilon}, A_{\text{av}}) = \tilde{\alpha}_{\bar{\epsilon}}$ and,for a matrix $M = [m_{ij}] \in \mathbb{R}^{m \times n}$, we denote its vectorization by $M^{\text{vec}} = [m_{11} \dots m_{m1}, m_{12} \dots m_{mn}]^{\mathsf{T}}$.

Lemma 2 (Descent Direction): Let $\tilde{\alpha}_{\bar{\epsilon}}$ denote the unique root of (9) *with $\bar{\epsilon} \in \mathbb{R}_{>0}$. Let $d = [d_1, \dots, d_m]^{\mathsf{T}}$, and let $K = \frac{1}{T}[A_1^{\text{vec}} \ A_2^{\text{vec}} \ \dots \ A_m^{\text{vec}}]$. Then,*

$$\frac{\partial \tilde{\alpha}_{\bar{\epsilon}}}{\partial d} = K^{\mathsf{T}} \left(\frac{QP}{\text{Trace}\,(QP)} \right)^{\text{vec}}$$

where $P \in \mathbb{R}^{n \times n}$ and $Q \in \mathbb{R}^{n \times n}$ are the unique solution to the two Lyapunov equations

$$\begin{aligned} (A_{\text{av}} - \tilde{\alpha}_{\bar{\epsilon}} I) \ P + P(A_{\text{av}} - \tilde{\alpha}_{\bar{\epsilon}} I)^{\mathsf{T}} + x_0 x_0^{\mathsf{T}} &= 0, \\ (A_{\text{av}} - \tilde{\alpha}_{\bar{\epsilon}} I)^{\mathsf{T}} Q + Q(A_{\text{av}} - \tilde{\alpha}_{\bar{\epsilon}} I) + C_{\text{av}} C_{\text{av}}^{\mathsf{T}} &= 0, \end{aligned} \quad (12)$$

and $I \in \mathbb{R}^{n \times n}$ denotes the identity matrix.

Proof: The expression for the partial derivative of the smoothed spectral abscissa with respect to d can be obtained from the composite function

$$\frac{\partial \tilde{\alpha}_{\bar{\epsilon}}}{\partial d} = \frac{\partial A_{\text{av}}}{\partial d} \frac{\partial \tilde{\alpha}_{\bar{\epsilon}}}{\partial A_{\text{av}}},$$

where $\frac{\partial A_{\text{av}}}{\partial d}$ follows immediately from (7d), and the expression for the derivative of $\tilde{\alpha}_{\bar{\epsilon}}$ with respect to A_{av} is given in [17, Theorem 3.2]. □

We remark that the equations (12) always admit a unique solution. To see this, we use the fact that $\tilde{\alpha}$ is an upper bound to $\alpha(A_{\text{av}})$, and observe that $(A_{\text{av}} - \tilde{\alpha}_{\bar{\epsilon}} I)$ is Hurwitz-stable for every A_{av}. A gradient descent method based on Lemma 2 is illustrated in Algorithm 1. Each iteration of the algorithm comprises the following steps. First, (lines $4-6$) a (possibly non feasible) descent direction ∇ is derived as illustrated in Lemma 2. Second, (line $7-8$) a gradient-projection technique [24] is used to enforce constraints (7d)-(7f). The update-step follows (line 9). Algorithm 1 employs a fixed stepsize $\mu \in (0, 1)$, and a terminating criterion (line 11) based on the Karush-Kuhn-Tucker conditions for projection methods [24]. The ϵ-update step, which constitutes the outer while-loop (line $2-13$), is then performed at each iteration of the gradient descent phase, and the line-search is terminated when $|\tilde{\alpha}_{\epsilon}| = 0$ cannot be achieved. To prevent the algorithm from stopping at local minima, the gradient descent algorithm can be repeated over multiple feasible initial conditions $d^{(0)}$. Finally, we illustrate in Table I typical execution times of Algorithm 1 on a commercial (laptop) processor.

TABLE I
EXECUTION TIME OF ALGORITHM 1 ON A 2.7 GHz INTEL CORE I5

n	23	113	265	309	431	665	852
Exec. Time [sec]	0.5	4.5	33.4	59.3	142.9	409	893

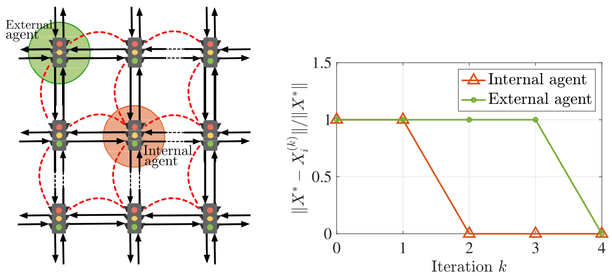
Remark 4 (Complexity of Algorithm 1): The computational complexity of Algorithm 1 can be derived as follows. First, solving (9) *to determine the value of the smoothed spectral abscissa can be performed via a root-finding algorithm (such as the bisection algorithm), whose complexity is a logarithmic function of the desired accuracy. Since computing the trace of a matrix has linear complexity in the matrix size, for given accuracy the complexity of this operation is $\mathcal{O}(n)$. Second, modern methods to solve Lyapunov equations (i.e.,* (12)*) rely on the Schur decomposition of the matrix A_{av} [25], whose complexity is $\mathcal{O}(n^3)$. It is worth noting that, given the Schur decomposition $A_{\text{av}} = UTU^{\mathsf{T}}$, where T is upper triangular and U is unitary, a decomposition for $(A_{\text{av}} - \tilde{\alpha} I)$ follows immediately by shifting T to $(T - \tilde{\alpha} I)$. Therefore, a single decomposition is required at each iteration of the gradient descent and the complexity of Algorithm 1 is therefore $\mathcal{O}(n^3)$.*

The space complexity of the algorithm can be derived as follows. Storing each matrix $A_{\text{av}}^{(k)}, Q, P$ requires n^2 units of space, while each vector Δ and d require m units of space. Thus, the space complexity of Algorithm 1 is $\mathcal{O}(n^2 + m)$.

Finally, we note that a constant-step discretization technique (1) *implies that the system size n scales linearly with $1/h$.* □

IV. DISTRIBUTED GRADIENT DESCENT

The centralized computation of $\{d_1^*, \dots, d_m^*, \epsilon^*\}$ assumes the complete knowledge of matrices $A_1, \dots A_m$, and requires to numerically solve the Lyapunov equations (12). For large-scale traffic networks, such computation imposes a limitation in the dimension of the matrix A_{av} and, consequently, on the number of signalized intersections that can be optimized simultaneously. Since the performance of the proposed optimization technique depends upon the possibility of modeling and optimizing large network interconnections, a limitation on the number of modeled roads and intersections constitutes a bottleneck toward the development of more efficient infrastructures. A possible solution to address this issue is to distribute the computation of the descent direction in Algorithm 1 among a group of agents, in a way that each agent is responsible for a subpart of the computation (e.g. see Fig. 4). In addition, certain model parameters describing the instantaneous state of network components (e.g. the current speed of flow in a certain road or the instantaneous value of



(a) (b)

Fig. 4. (a) Manhattan-like traffic interconnection. In this example, agents are signalized intersections that have local knowledge of the road interconnection (attached black arrows) and can communicate with neighbors (dashed red lines). Colored circles illustrate the information available to each agent. (b) Error between distributed and centralized solution vs iterations. Internal agents experience faster convergence due to shorter longest paths in the graph.

the transmission rate at a certain intersection), may be readily measurable by an agent that is located in the proximity of that component, while they may be unknown to other agents that are remotely located in the network. In this case, the benefit of a distributed implementation is that it allows local agents to directly include these model parameters into the optimal solution, thus avoiding unnecessary overheads due to transmission. In particular, agents may represent geographically-distributed control centers or clusters in parallel computing, each responsible for the control of a subset of the network.

In order to distribute the computation of solutions to (12), we focus on distributively solving equations of the form

$$\Lambda X + X\Lambda^{\mathsf{T}} + D = 0, \tag{13}$$

where $X = X^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ is unknown, $D = D^{\mathsf{T}} \in \mathbb{R}^{n\times n}$ is a given matrix, and $\Lambda \in \mathbb{R}^{n\times n}$. Let Λ be partitioned as

$$\Lambda = \Lambda_1 + \cdots + \Lambda_\nu, \tag{14}$$

where $\Lambda_i \in \mathbb{R}^{n\times n}$, $i \in \{1, \ldots, \nu\}$. We assume that each agent i knows Λ_i only. Note that Λ_i are sparse matrices, and their sparsity pattern depends upon the subpart of infrastructure associated with that agent. In addition, we assume that neighboring agents are allowed to exchange information by means of a communication interconnection. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the communication graph, where each vertex $i \in \{1, \ldots, \nu\}$ represents one agent, and $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ represents the communication lines. The method we propose to distributively compute X relies on an equivalent decomposition of equation (13) as a set of ν independent linear equations, as discussed next.

Lemma 3 (Distributed Solutions to (13)*): Let Λ be Hurwitz-stable. The following statements are equivalent*:

(i) X^* solves (13)*;*

(ii) For all $i \in \{1, \ldots, \nu\}$, there exists $D_i \in \mathbb{R}^{n\times n}$ s.t.

$$\Lambda_i X^* + X^*\Lambda_i^{\mathsf{T}} + D_i = 0, \text{ and } \sum_{i=1}^{\nu} D_i = D.$$

Proof: In order to prove the claim, we first observe that under the assumption of Hurwitz-stability for Λ, the solution X^* to (13) is unique.

$(i) \Rightarrow (ii)$. Let X^* denote the unique solution to (13). By expanding $\Lambda = \Lambda_1 + \cdots + \Lambda_\nu$, we obtain

$$\sum_{i=1}^{\nu}(\Lambda_i X^* + X^* A_i^{\mathsf{T}}) + D = 0.$$

Thus, by letting $D_i = -(\Lambda_i X^* + X^* A_i^{\mathsf{T}})$, (ii) immediately follows.

$(ii) \Rightarrow (i)$. Let $(\tilde{X}, \tilde{D}_1, \ldots, \tilde{D}_\nu)$ satisfy (ii), that is, for all $i \in \{1, \ldots, \nu\}$

$$\Lambda_i \tilde{X} + \tilde{X}\Lambda_i^{\mathsf{T}} + \tilde{D}_i = 0, \quad \sum_{i=1}^{\nu} \tilde{D}_i = \tilde{D}.$$

Notice that the existence of the solution to (13) guarantees the existence of $(\tilde{X}, \tilde{D}_1, \ldots, \tilde{D}_\nu)$. By substitution, we obtain $-\sum_{i+1}^{\nu}(\Lambda_i \tilde{X} + \tilde{X}\Lambda_i^{\mathsf{T}}) = D$, or in other words, $\tilde{X}$ satisfies $\Lambda\tilde{X} + \tilde{X}\Lambda^{\mathsf{T}} + D = 0$. The uniqueness of the solution to (13) implies $\tilde{X} = X^*$ and concludes the proof. □

Next, we show that the unknown matrices $X^*, D_1, \ldots, D_\nu$ can be reconstructed by the set of agents by cooperatively exchanging information. To this aim, for all $i \in \{1, \ldots, \nu\}$, we vectorize the set of Lyapunov equations in Lemma 3, and let $\bar{\Lambda}_i = \Lambda_i \otimes I + I \otimes \Lambda_i$. Then, from Lemma 3, we can restate (13) as a system of linear equations of the form

$$\underbrace{\begin{bmatrix} \bar{\Lambda}_1 & I & 0 & \cdots & 0 \\ \bar{\Lambda}_2 & 0 & I & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \\ 0 & I & \cdots & I & I \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} X^{\text{vec}} \\ D_1^{\text{vec}} \\ \vdots \\ D_\nu^{\text{vec}} \end{bmatrix}}_{w} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ D^{\text{vec}} \end{bmatrix}}_{z}, \tag{15}$$

where H is a given (known) matrix and w is an unknown parameter. In order to distribute the computation of vector w (and thus of X^*) among the ν distributed agents, we let

$$H_i = \begin{bmatrix} \bar{\Lambda}_i & 0 & \cdots & I & \cdots & 0 \\ 0 & I & \cdots & & \cdots & I \end{bmatrix}, \qquad z_i = \begin{bmatrix} 0 \\ D^{\text{vec}} \end{bmatrix},$$

for all $i \in \{1, \ldots, \nu\}$. At every iteration k, each agent i constructs a local estimate $\hat{w}_i^{(k+1)}$ by performing the following operations in order for all its neighbors:

(i) Receive $\hat{w}_j^{(k)}$ and $K_j^{(k)}$ from neighbor j;

(ii) $\hat{w}_i^{(k+1)} = \hat{w}_i^{(k)} + [K_i^{(k)} \ 0][K_i^{(k)} \ K_j^{(k)}]^{\dagger}(\hat{w}_i^{(k)} - \hat{w}_j^{(k)})$;

(iii) $K_i^{(k+1)} = \text{Basis}(\text{Im}(K_i^{(k)}) \cap \text{Im}(K_i^{(k)}))$;

(iv) Transmit $\hat{w}_i^{(k+1)}$ and $K_i^{(k+1)}$ to neighbor j;

where,

$$\hat{w}_i^{(0)} = H_i^{\dagger} z_i, \qquad K_i^{(0)} = \text{Basis}(\text{Ker}(H_i)).$$

The convergence of the procedure (i)-(iv) can be ensured by adopting an approach similar to the one discussed in [26].

Remark 5 (Communication Complexity): To characterize the communication complexity of the distributed algorithm, we observe that at every iteration each agent is required to transmit a set of packets describing the vector $\hat{w}_i^{(k+1)}$ and the subspace $K_i^{(k+1)}$. From (15)*, we note that each vector $\hat{w}_i^{(k+1)}$ has length νn^2, while the dimension of the subspace $K_i^{(k+1)}$ is variable at each iteration. In particular, the size of* *the subspace decreases at each iteration when the index i increases, with* $\dim(K_i^{(k+1)}) = 2\ n^2\nu$ *when* $i = 0$*, and* $\dim(K_i^{(k+1)}) = 0$ *at the final iteration. Thus, at each step (iv) of the algorithm a set of packets describing (at most)* $n^2\nu$ *variables is transmitted. Finally, we note that the number of iterations that each agent is required to perform depends on the cardinality of its neighbors and on the diameter of the network. We also note that, in order to reduce the communication burden of the algorithm, each agent can first perform operations (i)-(iii) sequentially for all its neighbors, and then re-transmit (i.e., perform step (iv)).* □

Next, we numerically validate the algorithm for a test-case traffic interconnection. To this aim, we consider the Manhattan-like network interconnection depicted in Fig. 4 [27], and assume that each signalized intersection is equipped with a computational unit that is responsible for a subpart of the computation of (12), and is allowed to exchange information with the neighboring intersections by means of a set of communication channels (dashed-red lines in Fig. 4). To decompose the system as in (14), we assume that each agent has the sole knowledge of: (i) the local structure of the traffic interconnection, that is, the layout of interconnection between roads that are adjacent to that intersection (colored areas in Fig. 4(a)), and (ii) the current values of the intersection outflow parameters $c(r_i, r_k)$, and of the speed of the flow γ_i in the adjacent roads. We illustrate in Fig.4(b) the convergence of the distributed procedure (i)-(iv), by comparing the accuracy of the local estimate $\hat{X}_i^{(k)}$ with respect to the centralized solution X^* as a function of the iteration step k. As discussed in [26], this class of procedures will compute $\hat{X}_i^{(k)} = X^*$ in at most $\text{diam}(\mathcal{G})$ steps, where $\text{diam}(\mathcal{G})$ denotes the diameter of $\mathcal{G}$.

V. Simulations

This section provides numerical simulations in support to the methods presented in this paper. We generate test cases using real-world traffic networks from the OpenStreet Map database and validate the techniques on a microscopic simulator based on *Sumo* [28]. A demo of the experimental setup adopted in this section is available online [29].

A. Averaging Technique

In order to validate our averaging technique, we first focus on a single road connected at downstream to a signalized intersection. To illustrate the discharging pattern emerging from the switching behavior of the signalized intersection, we assume the road has initially $x^\sigma(0) = 65$ vehicles in its downstream section, and zero inflows at all times. In all our simulations, we assume that each green phase is followed by a yellow phase, and we incorporate the durations of each yellow phase (clearance time) into the green times. We illustrate in Fig. 5(a) a comparison between the road discharging patterns in the microscopic simulation and in the average model (5) for different choices of the cycle time T. The precision of the model is quantified in Fig. 5(b), where we illustrate the approximation error for different T, where

$$\text{Error}_\% = \frac{1}{H}\int_0^H \frac{\|x - x_{\text{av}}\|}{\|x_{\text{av}}\|}\, dt,$$

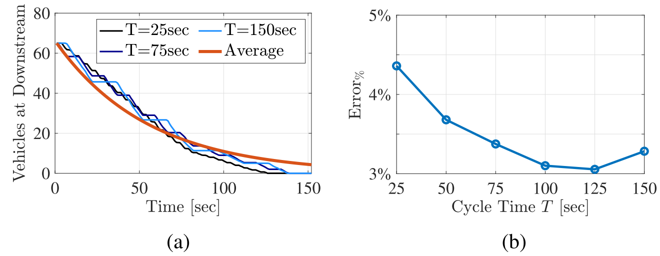


Fig. 5. Accuracy of average dynamical model (5) with respect to microscopic simulations for a single signalized road. (a) Time evolution of the density at downstream for different intersection cycle time T. (b) Accuracy of the average dynamics in relation to the intersection cycle time T.

captures the deviation between the microscopic simulation and the average model, normalized over the time horizon $[0, H]$. As illustrated in the figure, inaccuracies due to linearization and time-averaging are lower than 5% for common cycle times.

B. Macroscopic Simulations

To validate our modeling assumptions and optimization techniques, we initially perform a set of macroscopic simulations based on the well-established Cell Transmission Model (CTM) [18]. We consider its averaged version discussed in [16], [22], with piecewise affine demand and supply functions, and adopt a proportional allocation rule to model congestion at the intersections [3]. The averaged Cell Transmission Model builds upon the traditional (non-averaged) version of the model by replacing the switching behavior of the signalized intersections with the average flow through the junction (see [16]). We stress that the Cell Transmission Model is adopted here to simulate the actual dynamics of the network, while model (5) is the system description used in the optimization.

We consider the Manhattan-like traffic network interconnection sketched in Fig. 4(a), with $n_\text{r} = 24$ roads of length $\ell = 0.1$mi. We construct the CTM by associating a state (cell) to each section of road interconnecting two signalized intersections. For all cells, we let the free-flow speed be $v_\text{ff} = 30$mi/h, the speed of backward propagation be $v_\text{bp} = -30$mi/h, the jam density be $x^{\max} = 20$veh, and use maximum flows ${f_s}^{\max} = {f_d}^{\max} = 30$veh/min. Turning ratios at each intersections are chosen so that vehicles are split equally among all outgoing links, and the cycle time is $T = 100$sec.

In order to generate comparable results between the CTM and our model, we construct (4) by letting $\sigma_i = \ell_i$ for all $r_i \in \mathcal{R}$ (i.e. we model each road by means of a single state variable). In all simulations, source roads $\mathcal{S}$ and destination roads $\mathcal{D}$ are the roads at the boundaries of the network, network inflows are identical $u_{r_i} = u_r$ for all $r_i \in \mathcal{S}$, and $x_0 = 10$veh in all roads.

We evaluate the benefits of our intersection-control method by comparing its performance with: (i) a fixed-time control policy, and (ii) the control technique proposed in [16], that we briefly illustrate in the following. The fixed-time control policy consists in assigning constant split times at all the intersections, where green times are divided uniformly among all links connected at downstream to that intersection.

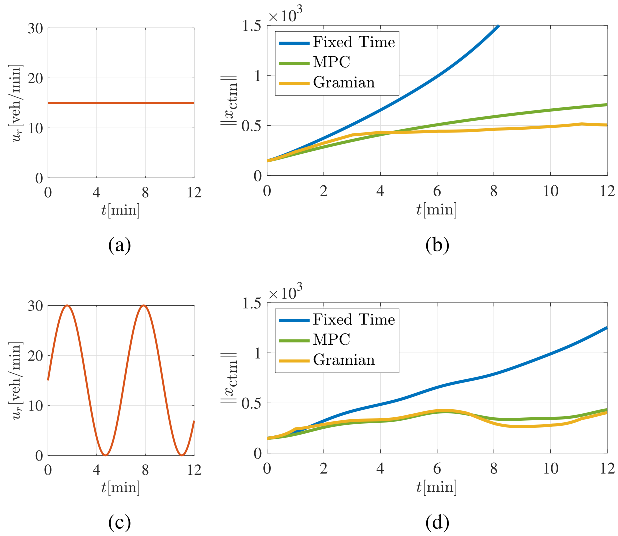


Fig. 6. Performance of the method evaluated on the cell transmission model. (a)-(b) constant network inflows. (c)-(d) time-varying network inflows.

The method discussed in [16] consists in performing network-wide design of the green split times by implementing a Model Predictive Control (MPC) optimization technique that relies on the averaged Cell Transmission Model for state prediction. We remark that the latter technique is adopted here for comparison because of its similarity with our framework in the adoption of a time-averaging method. In particular, MPC is performed by discretizing the averaged Cell Transmission Model and by considering a one-step ahead prediction horizon (representing a full cycle of the intersections, with $T = 100$sec), where the instantaneous values of the network state (density) are sensed from the network at each time step. In particular, we discretized the system using the Euler discretization with sample time $T_s = 10$sec, which satisfies the stability assumption $v_{\rm ff} T_s/\ell < 1$ that guarantees convergence of the model [18].

We report in Fig. 6 a comparison between the time-evolution of the cost of congestion obtained by simulating the averaged Cell Transmission Model for the three controllers under consideration. In particular, we consider two scenarios. First, we let the inflows be constant $u_r = 15$ veh/min for all source roads (Fig. 6(a)-(b)). Second, we let the inflows be time-varying $u_r = 15(1 + \sin(t))$veh/min (Fig. 6(c)-(d)). We observe that in both cases our controller outperforms the two control policies considered in the comparison. In particular, for constant network inflows our controller achieves an improvement of over 80% with respect to fixed-time control, and of about 40% with respect to the MPC-based control technique. We interpret these results by observing that, although our approach is based on a simplified model description of the system, it allows us to take into account larger control horizons with respect to (tractable) MPC control policies, thus resulting in increased network performance in the long term. This observation is further supported by the transient phase of the cost function in Fig. 6(b). In fact, we can observe that during the time interval $0-3$min the MPC-based control method reacts more effectively to changes in inflows thanks to (i) the availability of a more precise model that can capture quickly-varying regimes, and (ii) the shorter update interval. However, the benefits of a faster response degrade over time (see time interval $3 - 12$min) due to the lack in adequate prediction horizons that can capture all the relevant system dynamics. We conclude by observing that in the presence of quickly-varying inflow rates, both methods suffer from the lack of appropriate knowledge in the unknown network inflow rates and, in this case, the performance of the two techniques is comparable.

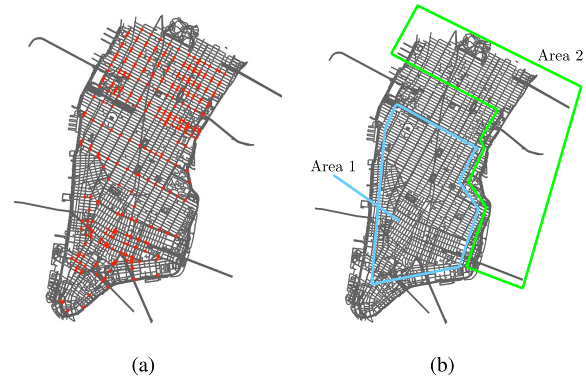


Fig. 7. Urban interconnection of Manhattan, NY, USA. (a) Red dots denote the set of signalized intersections considered in the study. (b) Commute zones.

TABLE II
MANHATTAN NETWORK INFLOW RATES

Time [sec] (From - To)	Area 1 Inflow [veh/h]	Area 2 Inflow [veh/h]
$0 - 2500$	4000	0
$2500 - 4000$	0	0

C. Microscopic Simulations

We consider a test case scenario inspired by the area of Manhattan, NY (Fig. 7), which features $n_{\rm r} = 958$ roads and $n_{\mathcal{I}} = 332$ signalized intersections. We replicate a daily-commute scenario, where sources of traffic $\mathcal{S}$ are uniformly distributed in the central area of the island (Area 1), and routing is chosen so that traffic flows are departing from the city, that is, destinations $\mathcal{D}$ are uniformly distributed within Area 2. Inflow rates used in the simulations are illustrated in Table II. To estimate the network turning rates, we set the simulator so that each vehicle follows the shortest path between its source and destination, and derive the turning rates $\varphi(r_i, r_k)$ for every pair of roads by computing the fraction of traffic flow on every route. Although in our simulations we make the assumption that the traffic patterns are known, in many practical scenarios the turning preferences are typically inferred from measured or historical traffic data [15].

We consider three control policies, described next.

1) Gramian-Optimization Settings: We model the network by means of the technique discussed in Section II, where we let $h = 0.1$mi and associate $\sigma_i = \lceil \ell_i/h \rceil \geq 1$ states to each section of road interconnecting two signalized intersections. We observe that, for the Manhattan interconnection shown in Fig. 7, we obtain $n = \sum_{i=1}^{n_{\rm r}} \sigma_i = 3091$, which corresponds to an average of approximately 3 states associated with each link. We solve the optimization (11) with cycle time $T = 100$sec (corresponding to $\text{Error}_{\%} \approx 3\%$, see Fig. 5). Moreover, the solution to (10) is re-computed with updated network conditions A_{av} and x_0 every $T_{\text{update}} = 500$sec by sensing these parameters from the microscopic simulation. The implementation of the gradient-descent algorithm was performed in Python, and the computation of the descent direction (Lemma 2) was performed using tools from the NumPy library. Finally, in order to emphasize the benefits of our optimization method and to make the results independent on the offset optimization algorithm adopted, we performed no offset optimization to decide the specific sequence of phases at the intersections. Thus, our simulation results represent a lower bound on the performance that can be achieved when offset optimization is applied to the output of our optimization.

2) Max-Pressure Settings: The Max-Pressure [1] is a controller that can be distributively implemented at the single-intersection level, and that requires only local information concerning the instantaneous traffic densities in the roads that are adjacent to that intersection. In particular, at each intersection and at each time step, the controller computes the difference between the number of vehicles waiting (on each road) and the number of vehicles at their downstream road, and activates the phase associated to the road with the largest difference value ("pressure") for a fixed time interval. In the simulation, the Max-Pressure is implemented through the Sumo TraCI tool, by associating a set of four phases to each intersection, where the activation time of each phase is set to $T/4 = 25$sec (thus, similarly to the Gramian-based optimization settings, the cycle time is $T = 100$sec).

D. Fixed-Time Settings

Fixed-time control is widely-adopted policy in practice thanks to its simplicity [12]. In this policy, the activation time of each phase is constant and proportional to the average traffic flow in the upstream roads, which are inferred from historical data. To implement this policy we estimated the average traffic flows by combining the network demand with the vehicles routing policy, and used $T = 100$sec.

Fig. 8(a) shows a comparison between the cost functions resulting from the three policies considered, while Fig. 8(b) shows the time evolution of the total number of vehicles in the network (network occupancy). The plots demonstrate the benefits of using the optimization (7): the improvement in cost function is of almost 60% with respect to fixed-time control, and of about 46% with respect to Max-Pressure. Moreover, Fig. 8(b) demonstrate the effectiveness of the cost function in (7) to capture the network congestion. In fact, control policies that minimize the cost (7) result in networks with reduced overall congestion (i.e. total number of vehicles in the network), and thus in increased network throughput.

The benefits of more efficient control policies on the network overall congestion can be further visualized by means of the illustration in Fig. 9. The figure illustrates the time evolution of the network congestion (measured as [veh/mi]) in the simulation for two control policies: Gramian-Based and Max-Pressure. The graphic shows that in the absence of external inflows, the network is evacuated faster when the former control technique is adopted, supporting our claim that a global model description results in increased network performance as compared to distributed techniques that rely on local knowledge of the traffic dynamics.

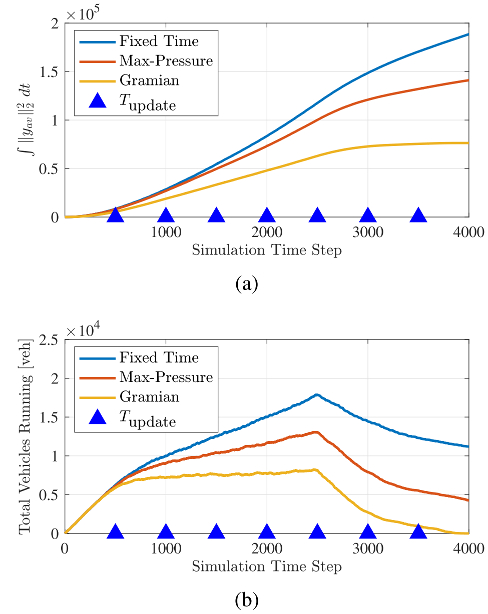


Fig. 8. Network performance of the Manhattan interconnection assessed via a microscopic simulator for three control policies. Blue triangles denote the instants when the optimal solution is recomputed with updated A_{av} and x_0.

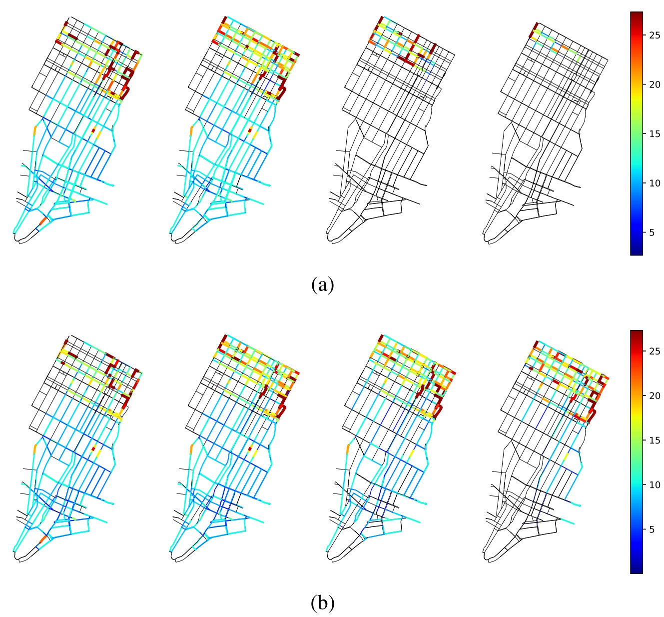


Fig. 9. Time evolution of the network state [×10veh/mi] for two control policies. (a) Solution to (7). (b) Max-Pressure policies. Simulation time from left to right is 1000sec, 2000sec, 3000sec, and 4000sec.

VI. Conclusion

This paper describes a simplified model to capture the overall dynamics of urban traffic networks. We formalize the goal of optimizing network congestion as an optimization problem that aims at minimizing a metric of controllability of the associated dynamical system. We adapt our technique to fit distributed and/or parallel implementations, and provide an efficient way to optimize large groups of intersections under the practical assumption that each agent has only local knowledge on the infrastructure. Our results show that the availability of an global, although approximate, model of the system interconnection can considerably improve the network efficiency, and allows for a more efficient and tractable analysis compared to traditional models. We envision that our model of traffic network and the proposed optimization framework will be useful in future research targeting design of traffic networks, control, and security analysis. Interesting topics for future works include a performance evaluation with respect to emerging machine learning intersection control techniques.

References

[1] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 177–195, Nov. 2013.

[2] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," *Control Eng. Pract.*, vol. 10, no. 2, pp. 183–195, Feb. 2002.

[3] G. Como, E. Lovisari, and K. Savla, "Convexity and robustness of dynamic traffic assignment and freeway network control," *Transp. Res. B, Methodol.*, vol. 91, pp. 446–465, Sep. 2016.

[4] Q. Ba, K. Savla, and G. Como, "Distributed optimal equilibrium selection for traffic flow over networks," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Osaka, Japan, Dec. 2015, pp. 6942–6947.

[5] Y. Zhang and P. A. Ioannou, "Combined variable speed limit and lane change control for highway traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1812–1823, Jul. 2017.

[6] G. Gomes and R. Horowitz, "Optimal freeway ramp metering using the asymmetric cell transmission model," *Transp. Res. C, Emerg. Technol.*, vol. 14, no. 4, pp. 244–262, Aug. 2006.

[7] S. Coogan, E. Kim, G. Gomes, M. Arcak, and P. Varaiya, "Offset optimization in signalized traffic networks via semidefinite relaxation," *Transp. Res. B, Methodol.*, vol. 100, pp. 82–92, Jun. 2017.

[8] P. B. Hunt, D. I. Robertson, R. D. Bretherton, and M. C. Royle, "The SCOOT on-line traffic signal optimisation technique," *Traffic Eng. Control*, vol. 23, no. 4, pp. 190–192, Apr. 1982.

[9] P. Mirchandani and L. Head, "A real-time traffic signal control system: Architecture, algorithms, and analysis," *Transp. Res. C, Emerg. Technol.*, vol. 9, no. 6, pp. 415–432, Dec. 2001.

[10] N. H. Gartner, F. J. Pooran, and C. M. Andrews, "Implementation of the OPAC adaptive control strategy in a traffic signal network," in *Proc. IEEE Intell. Transp. Syst.*, Oakland, CA, USA, Aug. 2001, pp. 195–200.

[11] P. Lowrie, "Scats, sydney co-ordinated adaptive traffic system: A traffic responsive method of controlling urban traffic," Traffic Authority NSW, Darlinghurst, NSW, Australia, Tech. Rep. 00772163, 1990.

[12] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proc. IEEE*, vol. 91, no. 12, pp. 2043–2067, Dec. 2003.

[13] K. Aboudolas, M. Papageorgiou, and E. Kosmatopoulos, "Store-and-forward based methods for the signal control problem in large-scale congested urban road networks," *Transp. Res. C, Emerg. Technol.*, vol. 17, no. 2, pp. 163–174, Apr. 2009.

[14] L. B. de Oliveira and E. Camponogara, "Multi-agent model predictive control of signaling split in urban traffic networks," *Transp. Res. C, Emerg. Technol.*, vol. 18, no. 1, pp. 120–139, Feb. 2010.

[15] S. Coogan, C. Flores, and P. Varaiya, "Traffic predictive control from low-rank structure," *Transp. Res. B, Methodol.*, vol. 97, pp. 1–22, Mar. 2017.

[16] P. Grandinetti, C. C.-de Wit, and F. Garin, "Distributed optimal traffic lights design for large-scale urban networks," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 950–963, May 2019.

[17] J. Vanbiervliet, B. Vandereycken, W. Michiels, S. Vandewalle, and M. Diehl, "The smoothed spectral abscissa for robust stability optimization," *SIAM J. Optim.*, vol. 20, no. 1, pp. 156–171, 2009.

[18] C. F. Daganzo, "The cell transmission model, part II: Network traffic," *Transp. Res. B, Methodol.*, vol. 29, no. 2, pp. 79–93, 1995.

[19] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin, Germany: Springer, 2013.

[20] M. Lighthill and G. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proc. Roy. Soc. London A, Math. Phys. Sci.*, vol. 229, no. 1178, pp. 317–345, 1955.

[21] G. Bianchin and F. Pasqualetti, "A network optimization framework for the analysis and control of traffic dynamics and intersection signaling," in *Proc. IEEE Conf. Decis. Control (CDC)*, Miami Beach, FL, USA, Dec. 2018, pp. 1017–1022.

[22] C. C. De Wit. *An Average Study of the Signalized Cell Transmission Model*. Accessed: Oct. 1, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01635539/document

[23] W. Xia and M. Cao, "Analysis and applications of spectral properties of grounded Laplacian matrices for directed networks," *Automatica*, vol. 80, pp. 10–16, Jun. 2017.

[24] D. Luenberger and Y. Ye, *Linear Nonlinear Programming*. New York, NY, USA: Springer, 1984.

[25] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation $AX + XB = C$ [F4]," *Commun. ACM*, vol. 15, no. 9, pp. 820–826, Sep. 1972.

[26] F. Pasqualetti, R. Carli, and F. Bullo, "Distributed estimation via iterative projections with application to power network monitoring," *Automatica*, vol. 48, no. 5, pp. 747–758, 2012.

[27] S. Lämmer and D. Helbing, "Self-control of traffic lights and vehicle flows in urban road networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 4, Apr. 2008, Art. no. P04019.

[28] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO–simulation of urban mobility: An overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simulation*, 2011, pp. 63–68.

[29] G. Bianchin and F. Pasqualetti. *Gramian-Based Optimization*. Accessed: Oct. 1, 2018. [Online]. Available: https://github.com/gianlucaBianchin/Gramian-Based-Traffic-Optimization

Gianluca Bianchin (S'15) received the Laurea degree in information engineering and the Laurea Magistrale degree *(summa cum laude)* in controls engineering from the University of Padua, Padua, Italy, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree in mechanical engineering with the University of California at Riverside. In 2018, he was a Research Intern with the Pacific Northwest National Laboratory. His main research interests are in the modeling, analysis, and control of large-scale interconnected systems, with a focus on transportation networks.

Fabio Pasqualetti received the Laurea degree (B.Sc. equivalent) in computer engineering and the Laurea Magistrale degree (M.Sc. equivalent) in automation engineering from the University of Pisa, Italy, in 2004 and 2007, respectively, and the Doctor of Philosophy degree in mechanical engineering from the University of California at Santa Barbara, in 2012. He is currently an Assistant Professor with the Department of Mechanical Engineering, University of California at Riverside. His main research interests include secure control systems, with application to multi-agent networks, distributed computing, power networks, computational neuroscience, vehicle routing, and combinatorial optimization.