Transportation Research Part C 145 (2022) 103910

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Traffic-light control in urban environment exploiting drivers' reaction to the expected red lights duration ☆,☆☆

Matteo Scandella [a,*], Arnob Ghosh [b,1], Michelangelo Bin [a], Thomas Parisini [a,c,d]

[a] Department of Electrical and Electronic Engineering, Imperial College London, London, United Kingdom
[b] The Ohio State University, Columbus, OH, USA
[c] KIOS Research and Innovation Center of Excellence, University of Cyprus, Cyprus
[d] Department of Engineering and Architecture, University of Trieste, Trieste, Italy

ARTICLE INFO

Keywords:
Traffic light control
Traffic network
Congestion

ABSTRACT

Traffic congestion in urban environment is one of the most critical issue for drivers and city planners for both environment and efficiency reasons. Traffic lights are one of the main tools used to regulate traffic by diverting the drivers between different paths. Rational drivers, in turn, react to the traffic light duration by evaluating their options and, if necessary, by changing direction in order to reach their destination quicker. In this paper, we introduce a macroscopic traffic model for urban intersections that incorporates this rational behavior of the drivers. Then, we exploit it to show that, by providing additional information about the expected red-time duration to the drivers, one can decrease the amount of congestion in the network and the overall length of the queues at the intersections. Additionally, we develop a control policy for the traffic lights that exploits the reaction of the drivers in order to divert them to a different route to further increase the performances. These claims are supported by extensive numerical simulations.

1. Introduction

1.1. Motivation

According to the fifth IPCC report on mitigation of climate change (Edenhofer et al., 2014), in 2010 the transport sector was responsible for approximately the 23% of the entire energy-related CO_2 emissions (Edenhofer et al., 2014, Chapter 8). In addition, "Greenhouse gas (GHG) emissions from the transport sector have more than doubled since 1970, and have increased at a faster rate than any other energy end-use sector . . . Around 80% of this increase has come from road vehicles" (Edenhofer et al., 2014, Section 8.1, p. 605). Moreover, road traffic is associated with several other environmental, financial, and social problems due to congestion, delays, and infrastructure maintenance or building. All these negative consequences are exacerbated in presence of high-density traffic and congestion, and road traffic steadily increases every year (ACEA, 2019; UK Department for Transport, 2019; Sprung et al., 2018). Therefore, rethinking the way traffic is managed is necessary to guarantee a sustainable future for transportation, and regulating traffic flows efficiently is becoming an ever more important control challenge.

☆ This work has been supported by European Union's Horizon 2020 research and innovation programme under grant agreement no. 739551 (KIOS CoE) and by the Italian Ministry for Research in the framework of the 2017 Program for Research Projects of National Interest (PRIN), Grant no. 2017YKXYXJ.
☆☆ Part of this work has been presented at the IEEE Conference on Decision and Control (CDC)'2021 (Ghosh et al., 2021).
* Corresponding author.
E-mail address: m.scandella@imperial.ac.uk (M. Scandella).
[1] Part of this work has been done while Arnob Ghosh was at the Imperial College London.

https://doi.org/10.1016/j.trc.2022.103910
Received 4 April 2022; Received in revised form 9 August 2022; Accepted 24 September 2022
Available online 21 October 2022

In this article, we focus on an urban traffic setting and, in particular, on intersection control. For more than 50 years, computer-aided traffic lights have been the standard tool for controlling intersections (McShane, 1999), and quite an extensive literature exists on the topic (see Section 1.2). However, to the best of the authors' knowledge, none of the existing works considers the possibility of showing drivers the information on how much time they have to wait, on an average, to cross the intersection. When such information is provided, drivers can react to it by adjusting their route accordingly. Hence, showing to drivers the *Expected Red Time Duration* (ERTD from now on) may have a considerable impact on congestion, and unveils several control opportunities unexplored so far. In this article, we aim at introducing such information in the control loop, by showing to the drivers queuing at the intersection how much they are estimated to wait for a green light. First, we develop a new macroscopic model capturing the rational reaction of drivers to the ERTD information. The model supports origin–destination pairs and rational planning albeit being macroscopic. Then, we propose a new control policy that exploits the possibility of showing ERTD and a model of the drivers' reaction to control the traffic lights with enhanced performance.

Overall, our results suggest that showing the ERTD information to drivers may lead to considerable advantages in terms of alleviating the congestion also if the traffic-lights control policy is agnostic of it. In addition, if the control policy is designed to exploit such an additional degree of freedom, then better performance can be achieved.

1.2. Related works

Controlling traffic lights to alleviate congestion or maximize the throughput is a classical problem in transportation research. Traffic-light control designs based on dynamic programming or informed by control theory have been proposed, for instance, in Chang and Sun (2004), Zhao et al. (2011), Le et al. (2015), Tettamanti et al. (2014), Fleck et al. (2016), Chiou (2018), Nilsson and Como (2020) and Liu et al. (2020). See also Papageorgiou et al. (2003) and Eom and Kim (2020) for a broad overview. Heuristic solutions using machine learning to cope with complex optimization problems and uncertainties have been proposed in Spall and Chin (1997), Srinivasan et al. (2006), Arel et al. (2010), A and Bhatnagar (2011) and Chu and Wang (2017). See also Araghi et al. (2015) and Eom and Kim (2020) for a review, and Yau et al. (2017) for a recent survey about the use of reinforcement learning techniques.

In Varaiya (2013), a "back-pressure" (or "max-pressure") traffic light control algorithm has been proposed with the aim of stabilizing the queues length at the intersections. Back-pressure algorithms were first introduced as controllers for packet routing in computer networks with the aim of maximizing the throughput (Tassiulas and Ephremides, 1992; Dai and Lin, 2005). Similarly, the algorithm proposed in Varaiya (2013) maximizes the vehicles flow at the intersections, and it can be implemented in a decentralized manner. Other approaches pursuing the same goal in different settings appeared in Gregoire et al. (2015), Wu et al. (2019), Chow et al. (2020) and Al Islam et al. (2020). Similarly, Bani Younes and Boukerche (2014) and Younis and Moayeri (2017) studied a method to compute locally optimal traffic-light laws in a dynamic manner using a vehicular network architecture. Typically, back-pressure approaches guarantee stability of the queues, in the sense that, whenever possible depending on the actual demand, the mean queue length is guaranteed to be bounded. Here, instead, we do not provide such guarantees, and we rather focus on optimality with respect to the queues length. While optimality implies stability (whenever a stabilizing controller of the considered form exists), for implementability reasons we eventually employ approximations that inevitably lead to sub-optimal solutions. Hence, we lose the stability guarantees implied by optimality. In addition, we underline that modeling the drivers' reaction to the ERTD information introduces an additional loop in the model, which consequently becomes a two-level optimization problem, and thus considerably complicates the analysis. Since our main goal is to promote the importance of showing additional information such as the ERTD, in this article we prefer to focus on the qualitative effects of this new feature, and to postpone formal guarantees to future research. Likewise, we do not focus on numerical performance and decentralization, which are again left for future investigations. None of the cited works, however, contemplates the possibility of showing to the drivers ERTD information or something related to the time that they are expected to wait at the intersection.

Showing ERTD information opens a number of additional degrees of freedom for the control designer, among which the possibility of using such information to incentivize drivers to, or prevent them from, choosing a certain route. In this connection, this work indirectly intersects the literature of road pricing, where tolls and incentivization policies are customarily studied. See, in particular (Pigou, 1920; Beckmann et al., 1955; Walters, 1961; Yang and Huang, 2005; Small and Verhoef, 2007; De Palma and Lindsey, 2004) for marginal cost pricing, Verhoef (2002), Yang and Huang (2005) and Small and Verhoef (2007) for second-best approaches, Verhoef et al. (1997), Kockelman and Kalmanje (2005), Friesz et al. (2008), Yang and Wang (2011) and Lessan and Fu (2019) for credit and permits schemes, and Daganzo (1995), Viegas (2001) and Wang et al. (2010) for rationing approaches. Contrary to these works, however, we do not require drivers to pay any toll or to buy any credit or permits, as we only act on traffic lights.

The closest article to this work is Chai et al. (2017), in which drivers exploit vehicle-to-infrastructure (V2I) communication to know the status of the network and the controller at all times. With this information, drivers re-plan their route by using a modified shortest-path algorithm. Simulations employing different heuristic traffic-lights controllers are presented supporting the idea that providing drivers with more information on the network may have beneficial effects on congestion. Unlike Chai et al. (2017), we do not require V2I communication, and we do not assume that drivers know the global status of the network or that of the controller. Rather, drivers gain new information about their expected travel times only when they are in the proximity of the intersection and receive the information about the expected waiting time. Such information, moreover, only refers to that specific intersection, and does not give information about the other nodes of the network. Moreover, we model the drivers' reactions to the displayed information as an optimization problem in which we trade off rationality, randomness, and the natural inertia that drivers experience in changing their path. Finally, contrary to Chai et al. (2017), we propose an optimized traffic lights control policy that intrinsically exploits a model of the drivers' reaction to make better decisions.

1.3. Contributions

We consider an urban traffic network where each intersection may have multiple lanes, and we consider the problem of optimal traffic-lights control with the aim of alleviating congestion. More specifically:

1. We add to the traffic lights system a visual indication showing to the queuing drivers ERTD they will have in the foreseeable future.
2. We model traffic flows by means of a novel macroscopic model that supports origin–destination pairs, finite links capacity, and rational decision-making for the drivers. Specifically, at each crossing, drivers select their next path optimally according to an optimization problem trading off rational decisions relative to the expected travel time to their destination, randomness, and the natural inertia keeping them from changing direction.
3. We evaluate the effect of showing the ERTD numerically through simulations by comparing the performance of an optimal control policy in presence and absence of such information. We find that showing to the drivers the ERTD always carries considerable improvement in the control performances.
4. We introduce a novel control policy for the traffic lights that exploits a model of the drivers' reaction to the new information to optimize the traffic-light schedule. While the model of the drivers' reaction used to simulate their behavior is based on their origin and destination, the model used in the controller is not, since the origin–destination pairs are unknown to the controller.
5. We validate the proposed controller numerically through simulations. In all the considered cases, the controller performs better than the previous optimal control scheme. Overall, the numerical analysis suggests that showing to the drivers the ERTD may carry a considerable advantage in terms of congestion, also if the control scheme is unchanged. Moreover, they also suggest that embedding a model of the drivers' reaction in the controller allows a further enhancement of the performance.

1.4. Context and paper organization

Section 2 illustrates the innovative model of the traffic network where we characterize the evolution of the traffic flow across the network as a function of the green-light duration at the intersections. In Section 3, we model the drivers' behavior governing how they choose their routes to reach the destination in terms of the solution of two convex optimization problems. It is worth noting that Sections 2 and 3 do not yet deal with the control problem, and they are rather long with somewhat cumbersome notations and definitions that are nevertheless unavoidable to construct a model that, albeit being macroscopic, supports origin–destination pairs, rational drivers' behavior, and allow us to include realistic constraint on the capacity of the intersections and on the finite length of the queues. Indeed, Sections 2 and 3 are instrumental for the novel control scheme we propose. Moreover, the traffic model that results is of interest per se, since it blends the simplicity of a macroscopic model and some details of a microscopic one, and we consider it a second important contribution of the article. Section 4 presents the control scheme embedding a model on the drivers' behavior. The control action comes from the solution of a bi-level optimization problem originating from the interconnection of the drivers, which try to minimize on average their travel time, and the controller, which tries to move the traffic equilibrium towards an efficient regime. Finally, in Section 5, we provide extensive numerical results validating the main conceptual idea of showing the ERTD information and the proposed control scheme.

2. The traffic model

In this section, we derive a macroscopic model of the traffic flow in the network. In particular, in Section 2.1 we introduce the road structure and in Section 2.2 we derive the equations governing the traffic dynamics.

2.1. The road network

The road network is modeled by a directed graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Nodes represent intersections, and each edge represents a single road connecting two different intersections. In particular, if $(i, j) \in \mathcal{E}$ then the junctions $i \in \mathcal{V}$ and $j \in \mathcal{V}$ are connected. Fig. 1(a) shows an example of a road network where each node is represented with a gray square and each edge with a purple arrow. For each node $j \in \mathcal{V}$, we define a set $\mathcal{P}_j \subseteq \mathcal{V} \times \{j\} \times \mathcal{V}$ representing all the possible ways in which vehicles can cross the junction j. Specifically, if $(i, j, f) \in \mathcal{P}_j$, then a vehicle that comes from node i can go toward node f passing through node j. For example, each of the green arrows of Fig. 1(a) represents an element of the sets $\mathcal{P}_j$. It is important to notice that $\mathcal{P}_j$ may contain only a subset of the possible paths through j. For instance, in Fig. 1(a), $(\mathsf{H}, \mathsf{D}, \mathsf{A}) \notin \mathcal{P}_{\mathsf{D}}$. It is also possible that $\mathcal{P}_j = \varnothing$ for certain nodes (e.g., nodes a and g in Fig. 1(a)). We define $\mathcal{P} := \bigcup_{j \in \mathcal{V}} \mathcal{P}_j$, and call its elements *paths*.

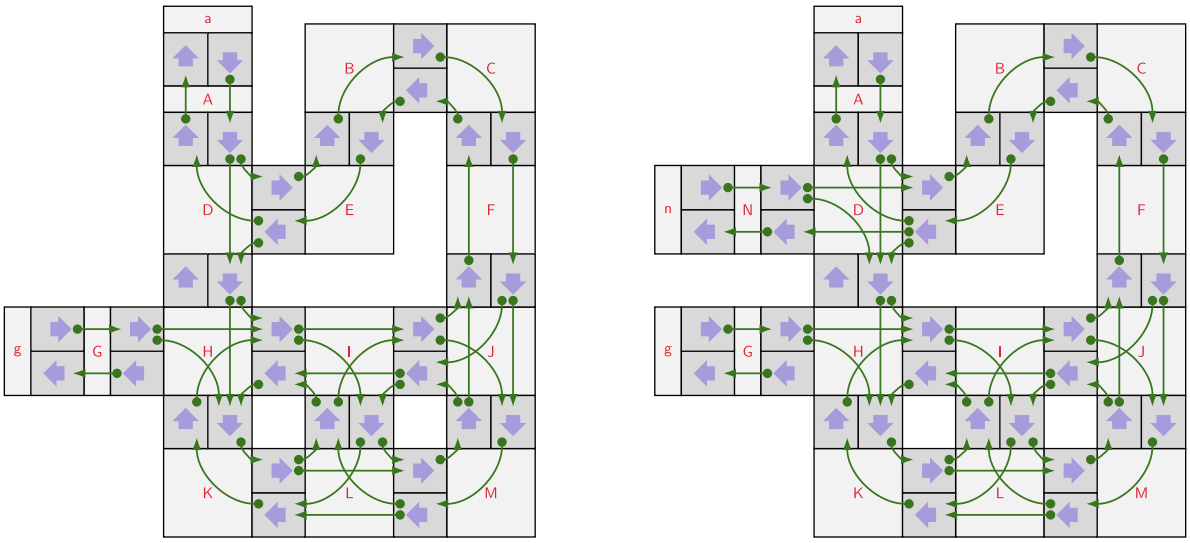
For each edge $(i, j) \in \mathcal{E}$, we define the set

$$\mathcal{O}_{i,j} := \left\{ k \in \mathcal{V} \;\middle|\; (i, j, k) \in \mathcal{P} \right\} \subseteq \mathcal{V},$$

containing all the possible nodes k that are reachable directly from j by vehicles coming from node i. Similarly, for each edge $(i, j) \in \mathcal{E}$, we define

$$\mathcal{I}_{i,j} := \left\{ k \in \mathcal{V} \;\middle|\; (k, i, j) \in \mathcal{P} \right\} \subseteq \mathcal{V}$$

as the set of all the possible nodes k from which vehicles can drive to node j passing through i.



(a) First example. (b) Second example.

Fig. 1. Two examples of networks of urban intersections. The intersections of the network are represented by gray rectangles with their name written in red in the center. The purple arrows represent the flow direction of the roads that connect the various nodes. Finally, the green arrows show the various ways a vehicle can cross the intersection.

We define $\mathcal{L} \subset \mathcal{V}$ as the set of all the *entry nodes* of the road network. Namely, the nodes inside $\mathcal{L}$ are those connected to the outside of the considered road network, from which vehicles can enter. For each entry node $i \in \mathcal{L}$, $\mathcal{P}_i = \emptyset$, and there exists a unique edge $(j, f) \in \mathcal{E}$ such that $(i, j, f) \in \mathcal{P}$. With $i \in \mathcal{L}$ and $(i, j, f) \in \mathcal{P}$, we define $B(i) := (i, j, f)$, which we call the *entry path* of the input node i. For example, a and g are the entry nodes of the road network of Fig. 1(a). Additionally, $B(\mathsf{a}) = (\mathsf{a}, \mathsf{A}, \mathsf{D})$ and $B(\mathsf{g}) = (\mathsf{g}, \mathsf{G}, \mathsf{H})$ are their entry paths. Then, we define the set of all the entry paths as

$$\mathcal{B} := B(\mathcal{L}) = \left\{(i, j, k) \in \mathcal{P} \;\middle|\; i \in \mathcal{L}\right\} \subseteq \mathcal{P}.$$

Finally, we let n_v be the cardinality of $\mathcal{V}$, n_p be the cardinality of $\mathcal{P}$ and n_b be the cardinality of $\mathcal{B}$.

2.2. The traffic dynamics

In this section, we derive a discrete-time model of the traffic dynamics. At each time instant, vehicles enter the network from the entry nodes $\mathcal{L}$ and move along the road network toward a specific node representing their final destination. When a vehicle reaches its destination, it leaves the network. We assume that, with each path $(i, j, f) \in \mathcal{P}$, there is associated a unique traffic light regulating the traffic on it. Hence, with each path there is associated a unique queue of vehicles waiting to cross the intersection. With slight abuse of notation, we use (i, j, f) also to denote such a queue. We denote by $N^{\max}_{i,j,f} \in (0, \infty]$ the maximum length of the queue (i, j, f). We set $N^{\max}_{i,j,f} = \infty$ for every entry path $(i, j, f) \in \mathcal{B}$. In this way, vehicles can enter the network also if all the queues are completely full. This is important as it guarantees that the vehicles influx is exogenous. The (unbounded) queues of the entry paths can be thus interpreted as queues of vehicles that want to enter the network, but that do not have the space to do it because of the traffic inside.

The model evolves in discrete time, with constant sampling time $h \in (0, +\infty)$. In particular, the tth interval is denoted by $T_t := [ht, h(t+1))$. As a simplifying assumption, we suppose that, given a time instant $t \in \mathbb{N}$ and a path $(i, j, f) \in \mathcal{P}$, all the vehicles that can depart during the time interval T_t from the queue (i, j, f) depart at the start of the time interval T_t and arrive at the next queue at the end of the time interval T_t or, equivalently, at the start of the subsequent time interval T_{t+1}. This assumption implies that the discretization time h is also the time needed to travel along an entire edge. We stress that this is a rather standard assumption widely used in this type of models (see, e.g., Ghosh et al. (2021), Chang and Sun (2004), Zhao et al. (2011), Le et al. (2015), Tettamanti et al. (2014), Gregoire et al. (2015), Varaiya (2013), Wu et al. (2019) and Miller (1963)). Moreover, this assumption is particularly motivated in our context because, in an urban scenario, edges are typically short and similar, and the traffic speed is small and homogeneous. Additionally, it is possible to model an edge with a larger travel time by splitting it into several edges. For example, in Fig. 1(a), the nodes E, B, C, and F are used to create a longer road connecting nodes D and J.

Given a path $(i, j, f) \in \mathcal{P}$, a node $q \in \mathcal{V}$, and a time $t \in \mathbb{N}$, we define the following three quantities:

- The *queue length* $N^t_{i,j,f,q} \in [0, N^{\max}_{i,j,f}]$ is the amount of vehicles waiting in the queue (i, j, f) at the start of the time interval T_t that have q as final destination.

- The *outgoing flow* $M^t_{i,j,f,q} \in [0,\infty)$ is the amount of vehicles that depart from the queue (i,j,f) at the start of the time interval T_t and that have q as final destination.
- The *inbound flow* $\Lambda^t_{i,j,f,q} \in [0,\infty)$ is the amount of vehicles that arrive at the queue (i,j,f) at the end of the time interval T_t and that have q as final destination.

We allow the possibility that drivers decide to change lane or direction once they have joined a queue. We assume that all these changes take place at the start of the time intervals. Given a path $(i,j,f) \in \mathcal{P}$, a node $q \in \mathcal{V}$, and a time $t \in \mathbb{N}$, we define the *post-change queue length* $\widetilde{N}^t_{i,j,f,q} \in [0, N^{\max}_{i,j,f}]$ as the amount of vehicles waiting in the queue (i,j,f) with destination q after all vehicles have finalized their lane changing. If no driver in (i,j,f) decides to change direction, then the post-change queue length is the same as the queue length, i.e., $\widetilde{N}^t_{i,j,f,q} = N^t_{i,j,f,q}$, for all $q \in \mathcal{V}$. The values of $\widetilde{N}^t_{i,j,f,q}$ are computed by solving an optimization problem taking into account: **(i)** the waiting time perceived by the drivers, **(ii)** their final destination, **(iii)** the network structure, and **(iv)** the maximum queue's lengths. This optimization problem and its solution are discussed later in Section 3.2. We stress that $N^t_{i,j,f,q}$, $\widetilde{N}^t_{i,j,f,q}$, $M^t_{i,j,f,q}$, and $\Lambda^t_{i,j,f,q}$ are real numbers.

For each $(i,j,f) \in \mathcal{P}$ and $t \in \mathbb{N}$, we define the following "aggregated" quantities

$$N^t_{i,j,f} := \sum_{q\in\mathcal{V}} N^t_{i,j,f,q}, \qquad \widetilde{N}^t_{i,j,f} := \sum_{q\in\mathcal{V}} \widetilde{N}^t_{i,j,f,q}, \qquad M^t_{i,j,f} := \sum_{q\in\mathcal{V}} M^t_{i,j,f,q}, \qquad \Lambda^t_{i,j,f} := \sum_{q\in\mathcal{V}} \Lambda^t_{i,j,f,q}.$$

Every traffic light alternates a green and a red light in each time interval. Given $(i,j,f) \in \mathcal{P}$ and $t \in \mathbb{N}$, we define $g^t_{i,j,f}$ as the *duty cycle* of the traffic light operating at (i,j,f). Namely, $g^t_{i,j,f}$ is the fraction of time in which the light is green in the time interval T_t. Therefore, a vehicle in the path (i,j,f) is allowed to cross for $hg^t_{i,j,f}$ time units during the time interval T_t, and it is blocked for the rest of the interval. To avoid situations where a path is completely blocked, the green light duration is supposed to be lower-bounded by a common bound $g^{\min} \in (0,1)$. Hence, $g^t_{i,j,f} \in [g^{\min}, 1]$ for all $t \in \mathbb{N}$ and all $(i,j,f) \in \mathcal{P}$. As explained in detail in Section 4, at each $t \in \mathbb{N}$, the tuple $g^t = (g^t_{i,j,f})_{(i,j,f)\in\mathcal{P}}$ is the *control variable* that a controller decides in order to manage traffic and influence the drivers' decision.

Given a path $(i,j,f) \in \mathcal{P}$, a node $q \in \mathcal{V}$ and a time $t \in \mathbb{N}$, the queue length at the end of the time interval T_t and at the start of the time interval T_{t+1} is obtained by adding the inbound flow $\Lambda^t_{i,j,f,q}$ and removing the outgoing flow $M^t_{i,j,f,q}$ from the post-change queue lengths $\widetilde{N}^t_{i,j,f,q}$. Therefore, we have

$$N^{t+1}_{i,j,f,q} = \widetilde{N}^t_{i,j,f,q} + \Lambda^t_{i,j,f,q} - M^t_{i,j,f,q}. \tag{1}$$

Next, we define the inbound flows. To this aim, we first consider a non-entry path $(i,j,f) \in \mathcal{P} \setminus \mathcal{B}$. Given a node $q \in \mathcal{V}$ and a time $t \in \mathbb{N}$, we define

$$\Lambda^t_{i,j,f,q} = \alpha_{i,j,f,q} \sum_{k\in\mathcal{I}_{i,j}} M^t_{k,i,j,q}.$$

In the previous equation, the summation term gives the amount of vehicles that depart from the downstream nodes at the start of the time interval T_t and travel along the edge (i,j) with final destination q. The term $\alpha_{i,j,f,q} \in [0,1]$, instead, is the fraction of vehicles with final destination q joining the queue (i,j,f). The value of the coefficients $\alpha_{i,j,f,q}$ is computed by solving an optimization problem that takes into account the final destination of the drivers and the network structure. This algorithm is discussed in detail later in Section 3.1. Since the vehicles that reach their destination are supposed to leave the network, if $j = q$, we have $\alpha_{i,j,f,q} = 0$.

We now consider the entry paths. If $(i,j,f) \in \mathcal{B}$, then $\mathcal{P}_i = \varnothing$. Hence, $\mathcal{I}_{i,j} = \varnothing$. Therefore, there is no flow of vehicles arriving toward the path (i,j,f). Instead, we define the incoming vehicles at $i \in \mathcal{L}$ during the interval T_t in terms of an exogenous variable $(\zeta^t_{i,q})_{q\in\mathcal{V}}$, where each term $\zeta^t_{i,q}$ equals the amount of vehicles entering the network at i with destination $q \in \mathcal{V}$. Hence, we set $\Lambda^t_{i,j,f,q} = \zeta^t_{i,q}$. In summary, we have

$$\Lambda^t_{i,j,f,q} = \begin{cases} \alpha_{i,j,f,q} \displaystyle\sum_{k\in\mathcal{I}_{i,j}} M^t_{k,i,j,q} & \text{If } i \in \mathcal{V}\setminus\mathcal{L}, \\ \zeta^t_{i,q} & \text{If } i \in \mathcal{L}. \end{cases} \tag{2}$$

The value of the outgoing flows are defined as the solution of an optimization problem that searches the maximum feasible outgoing flow for each queue and destination. Here, we are assuming that the outgoing flows are always equal to the maximum feasible value and, therefore, if a driver has time to depart, it does.

A value is feasible if it satisfies the following constraints:

- All the outgoing flows have to be non-negative. Hence,

$$M^t_{i,j,f,q} \geq 0, \qquad \forall (i,j,f) \in \mathcal{P},\ \forall q \in \mathcal{V}. \tag{3}$$

- Only the vehicles that are queuing at the start of the time interval T_t can cross the path. Thus,

$$M^t_{i,j,f,q} \leq \widetilde{N}^t_{i,j,f,q}, \qquad \forall (i,j,f) \in \mathcal{P},\ \forall q \in \mathcal{V}. \tag{4}$$

- Every link has a *maximum capacity* $v_{i,j,f} \in (0, +\infty)$, representing the maximum amount of vehicles that can cross (i, j, f) in any time interval if the traffic light is always green, and an *effective capacity* given by $v_{i,j,f}\ g^t_{i,j,f}$ for each $t \in \mathbb{N}$, which takes into account the green-light duration. Then, assuming that the vehicles with different destinations are distributed evenly in the queue, the effective capacity is subdivided among the possible destinations in the same way as the post-change queue length. Therefore,

$$M^t_{i,j,f,q} \le \frac{\widetilde{N}^t_{i,j,f,q}}{\widetilde{N}^t_{i,j,f}}\, v_{i,j,f}\ g^t_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P},\ \forall q \in \mathcal{V}. \tag{5}$$

- The queue length at the start of the next time interval T_{t+1} cannot exceed the maximum queue length. Hence,

$$N^{t+1}_{i,j,f} = \sum_{q \in \mathcal{V}} N^{t+1}_{i,j,f,q} \le N^{\max}_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P} \setminus \mathcal{B}, \tag{6}$$

  where the relation between the queue lengths and the outgoing flows is defined in (1) and (2). Notice that this constraint is not defined for the entry paths because, in these cases, $N^{\max}_{i,j,f} = \infty$. By excluding the entry paths, this constraint can always be satisfied because the vehicles can accumulate in the entry paths of the network if the downstream paths are congested.

Therefore, given $t \in \mathbb{N}$, the value of the outgoing flows is defined as the solution of the optimization problem

$$\mathscr{M}(t): \begin{cases} \max \displaystyle\sum_{(i,j,f) \in \mathcal{P}} \sum_{q \in \mathcal{V}} M^t_{i,j,f,q} \\ \text{Subject to (1), (2), (3), (4), (5), and (6).} \end{cases} \tag{7}$$

in the optimization variables $(M^t_{i,j,f,q})_{(i,j,f) \in \mathcal{P}, q \in \mathcal{V}}$. This is a linear optimization problem that can be solved using a suitable solver. This definition of the outgoing flows, together with (1) and (2), describes the time evolution of the traffic flow in the network.

3. Models of drivers' behavior

Drivers' decisions enter the traffic model presented in the previous Section 2 in two points: **(i)** In the definition of the coefficients $\alpha_{i,j,f,q}$ in (2), and **(ii)** in the definition of the post-change queue length variables $\widetilde{N}^t_{i,j,f,q}$ in (1). Conceptually, the determination of $\alpha_{i,j,f,q}$ and $\widetilde{N}^t_{i,j,f,q}$ are different tasks in that the former comes from a prior planning that only depends on the drivers' origin–destination pairs and the topology of the road network, while the latter also depends on real-time traffic data and on the value of the duty cycles $g^t_{i,j,f}$ decided by the controller. Hence, we treat the two tasks separately in the forthcoming sections.

3.1. Drivers' a priori route selection

In this section, we detail the procedure used to define the coefficients $\alpha = (\alpha_{i,j,f,q})_{(i,j,f) \in \mathcal{P}, q \in \mathcal{V}}$. These coefficients are determined in terms of the drivers' origin–destination pairs, and are tuned on an expected nominal value for the traffic lights, fixed a priori and independent of the control policy actually implemented. Hence, α represents the subdivision of the traffic flow at each intersection corresponding to an a priori optimistic plan, where each driver selects the shortest path to its destination and does not take traffic into account.

As a first step, we derive the expected time necessary to reach a given destination from a certain queue. When a vehicle moves along the network, it follows a sequence of consecutive paths, here formalized in terms of routes. A *route* of length $n \in \mathbb{N} \setminus \{0\}$ is a tuple of paths $((i_z, j_z, k_z))_{z=1}^n \in \mathcal{P}^n$ such that, either $n = 1$, or $i_z = j_{z-1}$ and $j_z = k_{z-1}$ for all $z \in \{2, \dots, n\}$. We let $\mathcal{R}_n \subset \mathcal{P}^n$ be the set of all the routes of length n, and $\mathcal{R} = \bigcup_{n=1}^{+\infty} \mathcal{R}_n$ be the set of all routes. We identify length-1 routes with their only element, e.g., by writing (i, j, k) in place of $((i, j, k))$. Moreover, with $r \in \mathcal{R}^n$ and $p \in \mathcal{R}^m$, we denote by $r \oplus p \in \mathcal{R}^{n+m}$ their concatenation whenever it is well-defined. Given a route $r \in \mathcal{R}$, we denote by $\ell(r) \in \mathbb{N} \setminus \{0\}$ its length. The time needed to follow a route $r = ((i_z, j_z, k_z))_{z=1}^n \in \mathcal{R}$ without traffic is given by

$$\tau(r) := \sum_{z=1}^{\ell(r)} h + \sum_{z=1}^{\ell(r)} \frac{h}{v_{i_z,j_z,k_z} \gamma_{i_z,j_z,k_z}} = h \left(\ell(r) + \sum_{z=1}^{\ell(r)} \frac{1}{v_{i_z,j_z,k_z} \gamma_{i_z,j_z,k_z}} \right), \tag{8}$$

in which:

- The first summation term is the time needed to cross all the edges in the route, and it is based on the assumption that the time needed to travel one edge is equal to the discretization step h (see Section 2).
- The second summation is the time needed to cross all the junctions in the route assuming that the green light duration of the traffic light at junction $(i, j, f) \in \mathcal{P}$ equals $\gamma_{i,j,f}$, where $\gamma_{i,j,f} \in (0, 1)$ is some value that drivers expect, a priori, for the duty cycle $g^t_{i,j,f}$.

The formula (8) builds on the assumption that drivers know the topology of the network and that they have a reasonable guess $\gamma_{i,j,f}$ for the duty cycles. In particular, $\gamma_{i,j,f}$ is the average value drivers assume for duty cycles when they have no information about the status of the network. For example, looking at the network in Fig. 1(a), we may take $\gamma_{\mathsf{E},\mathsf{B},\mathsf{C}} = \gamma_{\mathsf{C},\mathsf{B},\mathsf{E}} = 1$, since all the paths in node B are non-conflicting, and thus it is licit to assume that drivers presume they can always cross. Instead, it is reasonable to take

$\gamma_{i,j,f} = 1/3$, for all $(i,j,f) \in \mathcal{P}_{\mathrm{I}}$. In fact, the crossing at I has three conflicting phases, and it is licit to assume drivers assign to them the same duration a priori.

With $(i,j) \in \mathcal{E}$ and $q \in \mathcal{V}$, we define the set

$$\Phi_{i,j,q} := \left\{ ((i_z, j_z, k_z))_{z=1}^{n} \;\middle|\; i_1 = i \wedge j_1 = j \wedge k_n = q \right\} \subseteq \mathcal{R},$$

which contains all the routes connecting the edge (i,j) to the destination q. Notice that $\Phi_{i,j,q} = \varnothing$ if and only if it is not possible to reach the destination q from the edge (i,j). With the previous definitions in mind, we define the *optimal route* from an edge $(i,j) \in \mathcal{V}$ to a node $q \in \mathcal{V}$ as[2]

$$\varphi_{i,j,q} := \underset{r \in \Phi_{i,j,q}}{\arg\min} \{\tau(r)\} \in \mathcal{R}. \tag{9}$$

Consider now a vehicle with final destination $q \in \mathcal{V}$ that is traveling on the edge $(i,j) \in \mathcal{E}$ and needs to choose the next queue to join among the possible ones $\{(i,j,k)\}_{k \in \mathcal{O}_{i,j}}$. We can assign a weight to each possible decision depending on the time gain or loss that it leads to. In particular, we define the weights

$$\rho_{i,j,f,q} := \begin{cases} 0 & \text{If } j = q \\ \tau\big((i,j,f)\big) & \text{If } j \neq q \wedge f = q \\ +\infty & \text{If } j \neq q \wedge f \neq q \wedge \Phi_{j,f,q} = \varnothing \\ \tau\big((i,j,f) \oplus \varphi_{j,f,q}\big) & \text{If } j \neq q \wedge f \neq q \wedge \Phi_{j,f,q} \neq \varnothing. \end{cases} \tag{10}$$

The third case of (10) happens when it is not possible to reach the destination q by selecting the path (i,j,f). We stress that $\rho_{i,j,f,q}$ is an element of the extended real numbers because it can be equal to $+\infty$. For the remainder of the manuscript, we set by convention that $0(\pm\infty) = 0$.

Higher weights correspond to worse decisions. However, a completely rational flow is a rather strong assumption. This motivates defining the coefficients $\alpha = \left(\alpha_{i,j,f,q}\right)_{(i,j,f) \in \mathcal{P}, q \in \mathcal{V}}$ as the ones minimizing the following the cost function

$$J_a(\alpha) := \mu \sum_{(i,j,f) \in \mathcal{P}} \sum_{q \in \mathcal{V}} \alpha_{i,j,f,q}\ \rho_{i,j,f,q} + \sum_{(i,j,f) \in \mathcal{P}} \sum_{q \in \mathcal{V}} \alpha_{i,j,f,q}\ \log \alpha_{i,j,f,q}, \tag{11}$$

representing a compromise between rationality and randomness. Specifically, the cost function J_a is the sum of two terms. The first weights the time needed to reach the destination, where $\mu \in (0, +\infty)$ is a design parameter representing the drivers' *value of time*, i.e., the importance that they give to the time needed to reach their destination. Minimizing this term leads to the most rational decision in terms of value of time. The second term is a negative entropy term. As explained later, minimizing this implies maximizing dispersion, and it is thus associated with irrational behavior.

The negative entropy term is common in the context of learning theory, where it is used as a regularization term (Chen et al., 2016; Mertikopoulos et al., 2018; Neu et al., 2017). Here, it is used to model the fact that not all drivers perfectly plan their route in order to obtain the fastest travel time. For example, some vehicles may have only incomplete information about the road structure, may be distracted at the time of decision, or may have other objectives than minimizing the travel time. The parameter μ enters as a multiplier in the first term of (11). Hence, it can be used to adjust the trade-off between the rational and irrational behavior.

Since not all possible values of α are feasible, the optimization problem is constrained. In particular, a value of α is feasible if and only if it satisfies the following conditions:

- The coefficients need to be non-negative. Hence,

$$\alpha_{i,j,f,q} \geq 0, \qquad \forall (i,j,f) \in \mathcal{P},\ \forall q \in \mathcal{V}. \tag{12}$$

- Vehicles can only leave the network at their destination, and can enter only at the entry nodes. Thus,

$$\sum_{f \in \mathcal{O}_{i,j}} \alpha_{i,j,f,q} = 1, \qquad \forall (i,j) \in \mathcal{E},\ \forall q \in \mathcal{V} \setminus \{j\}, \tag{13}$$

  which expresses the conservation of the flow of vehicles with destination q at nodes $j \neq q$.
- Finally, vehicles must leave the network once at the destination node. Hence,

$$\alpha_{i,j,f,j} = 0, \qquad \forall (i,j,f) \in \mathcal{P}. \tag{14}$$

Therefore, the value of α is defined as the solution of the optimization problem

$$\mathscr{A} : \begin{cases} \min J_a(\alpha) \\ \text{Subject to (12), (13), and (14).} \end{cases} \tag{15}$$

It turns out that $\mathscr{A}$ can be solved in closed form, as stated by Lemma 1 below (the proof of Lemma 1 is in Appendix A).

[2] For simplicity, we suppose that the set of minimizers in (9) is a singleton. Otherwise, we can define the quantity $\varphi_{i,j,q}$ as an arbitrary element of it.

Lemma 1. *Problem $\mathscr{A}$ has a unique solution given by*

$$\alpha_{i,j,f,q} = \begin{cases} 0 & \text{If } j = q \\ \dfrac{\exp\left(-\mu\rho_{i,j,f,q}\right)}{\sum_{k\in\mathcal{O}_{i,j}} \exp\left(-\mu\rho_{i,j,k,q}\right)} & \text{If } j \neq q, \end{cases} \tag{16}$$

for all $(i,j,f) \in \mathcal{P}$ and all $q \in \mathcal{V}$.

We remark that $\alpha_{i,j,f,q} = 0$ whenever $\Phi_{j,f,q} = \varnothing$ because, in this case, $\rho_{i,j,f,q} = +\infty$. This makes sure that vehicles with destination q cannot end up in roads from which q is not reachable.

3.2. The driver's reaction to the waiting time

In this section, we detail a model of the drivers' reaction to the displayed waiting times and to the congestion at the junctions, thereby defining a procedure to compute the post-change queue lengths $(\widetilde{N}^t_{i,j,f,q})_{(i,j,f)\in\mathcal{P},q\in\mathcal{V}}$ at each time t. The waiting time perceived by a vehicle in a queue depends on its position in the queue and on the information it has about the duty cycle of the traffic light. The duty cycles $g^t_{i,j,f}$, in particular, may be either known to the drivers, if such information is displayed as we propose, or not, if this feature is disabled. To capture both cases, given $(i,j,f) \in \mathcal{P}$ and $t \in \mathbb{N}$, we define the *perceived duty cycle* as

$$\widehat{g}^t_{i,j,f} := \begin{cases} g^t_{i,j,f} & \text{If the drivers know } g^t_{i,j,f} \\ \gamma_{i,j,f} & \text{Otherwise.} \end{cases}$$

Hence, if the traffic light on path (i,j,f) displays the information about the duty cycle, then the drivers in the queue (i,j,f) know $g^t_{i,j,f}$, and they can use it to make an informed decision. Otherwise, they rely on the value $\gamma_{i,j,f}$ expected a priori (see Section 3.1).

By using the perceived duty cycle $\widehat{g}^t_{i,j,f}$, it is possible to define the *perceived waiting time* for a vehicle in position $x \in [0, N^t_{i,j,f}]$ in the queue $(i,j,f) \in \mathcal{P}$ at the start of the time interval T_t as

$$\widehat{w}^t_{i,j,f}(x) := \frac{hx}{v_{i,j,f}\,\widehat{g}^t_{i,j,f}}, \tag{17}$$

where we assume that $x = 0$ is the front of the queue and $x = N^t_{i,j,f}$ the back. The *perceived traveling time* for a vehicle with destination $q \in \mathcal{V}$ is then defined as the sum $\widehat{w}^t_{i,j,f} + \rho_{i,j,f,q}$ of the expected a priori crossing time $\rho_{i,j,f,q}$, defined in (10), and the perceived waiting time in the queue $\widehat{w}^t_{i,j,f}$.

Consider a vehicle with final destination $q \in \mathcal{V}$ that is in position $x \in [0, N^t_{i,j,k}]$ in the queue $(i,j,k) \in \mathcal{P}$ at the start of the time interval T_t. The vehicle must choose whether to stay in the same queue, or to change lane to one among the possible alternatives $\{(i,j,f)\}_{f\in\mathcal{O}_{i,j}}$. By following the same reasoning used in Section 3.1, we assign a weight to each possible decision. In particular, for each possible alternative queue (i,j,f), we define the weight

$$\widehat{\omega}^t_{i,j,k,f,q}(x) := \begin{cases} \xi\left(\widehat{w}^t_{i,j,f}(x) + \rho_{i,j,f,q}\right) - \sigma & \text{If } k = f \\ \xi\left(\widehat{w}^t_{i,j,f}(x+\eta) + \rho_{i,j,f,q}\right) & \text{If } k \neq f \wedge x + \eta < N^t_{i,j,f} \\ \xi\left(\widehat{w}^t_{i,j,f}\left(N^t_{i,j,f}\right) + \rho_{i,j,f,q}\right) & \text{If } k \neq f \wedge x + \eta \geq N^t_{i,j,f}, \end{cases} \tag{18}$$

in which $\xi, \sigma, \eta \in [0, +\infty)$ are parameters to be specified. The first case, in which $k = f$, corresponds to the decision not to change queue. This term is the sum of a rational term, given by the perceived traveling time associated with the present queue (scaled by ξ), and a negative term $-\sigma$ modeling the inertial preference toward the decision of not changing lane. The other two cases apply to the lanes different from the present one. Here, we assume that a vehicle that changes queue cannot maintain its position x in the new queue, but rather it joins the new queue in position $x + \eta$. Therefore, in the second case, the weight is given by the perceived traveling time in the selected queue in position $x + \eta$, weighted by ξ. In the third case, instead, the selected queue is shorter than the penalized position $x + \eta$. Thus, the vehicle joins the new queue at the end.

Since the weights depend on the position of the vehicle in the queue, in order to ease their computation, we subdivide the queue into n sections of equal length. In particular, we denote the zth section, $z \in \{1, \dots, n\}$, of the queue $(i,j,k) \in \mathcal{P}$ by

$$S^t_{i,j,k,z} := \left[(z-1)\frac{N^t_{i,j,k}}{n}, z\frac{N^t_{i,j,k}}{n}\right). \tag{19}$$

Then, given $q \in \mathcal{V}$ and $f \in \mathcal{O}_{i,j}$, we define $\widehat{\omega}^t_{i,j,k,f,q,z}$ as the mean weight among the vehicles with destination q that are in the queue section $S^t_{i,j,k,z}$. In particular, we have

$$\widehat{\omega}^t_{i,j,k,f,q,z} := \frac{1}{\frac{N^t_{i,j,k}}{n}} \int_{S^t_{i,j,k,z}} \widehat{\omega}^t_{i,j,k,f,q}(x)\ dx$$

$$= \begin{cases} \dfrac{\xi h}{v_{i,j,f}\widehat{g}^t_{i,j,f}} \dfrac{N^t_{i,j,k}(2z-1)}{2n} + \xi\rho_{i,j,f,q} - \sigma & \text{If } f = k \\ \dfrac{\xi h}{v_{i,j,f}\widehat{g}^t_{i,j,f}} \left(\dfrac{N^t_{i,j,k}(2z-1)}{2n} + \eta \right) + \xi\rho_{i,j,f,q} & \text{If } f \neq k \wedge z\dfrac{N^t_{i,j,k}}{n} + \eta < N^t_{i,j,f} \\ \dfrac{\xi h}{v_{i,j,f}\widehat{g}^t_{i,j,f}} N^t_{i,j,f} + \xi\rho_{i,j,f,q} & \text{If } f \neq k \wedge (z-1)\dfrac{N^t_{i,j,k}}{n} + \eta \geq N^t_{i,j,f} \\ \dfrac{\xi h}{v_{i,j,f}\widehat{g}^t_{i,j,f}} \left(z\left(N^t_{i,j,f} - \eta\right) + \eta - \dfrac{(z-1)^2 N^t_{i,j,k}}{2n} - \dfrac{n\left(N^t_{i,j,f} - \eta\right)^2}{2N^t_{i,j,k}} \right) + \xi\rho_{i,j,f,q} & \text{Otherwise.} \end{cases}$$

Now, if we denote by $\beta^t_{i,j,k,f,q,z} \in [0,1]$ the fraction of vehicles in section $S^t_{i,j,k,z}$, $(i,j,k) \in \mathcal{P}$ and $z \in \{1,\dots,n\}$, with destination $q \in \mathcal{V}$ that decide to join the queue (i,j,f) at time t, for some $f \in \mathcal{O}_{i,j}$, the post-change queue lengths are given as follows

$$\widetilde{N}^t_{i,j,f,q} = \sum_{k \in \mathcal{O}_{i,j}} \sum_{z=1}^{n} \beta^t_{i,j,k,f,q,z} \frac{N^t_{i,j,k,q}}{n} = \frac{1}{n} \sum_{k \in \mathcal{O}_{i,j}} \sum_{z=1}^{n} N^t_{i,j,k,q} \beta^t_{i,j,k,f,q,z}, \tag{20a}$$

$$\widetilde{N}^t_{i,j,f} = \sum_{q \in \mathcal{V}} \widetilde{N}^t_{i,j,k,q} = \frac{1}{n} \sum_{q \in \mathcal{V}} \sum_{k \in \mathcal{O}_{i,j}} \sum_{z=1}^{n} N^t_{i,j,k,q} \beta^t_{i,j,k,f,q,z}. \tag{20b}$$

In the remainder of the section, we define the values of the coefficients $\beta^t = \left(\beta^t_{i,j,k,f,q,z}\right)_{(i,j,k)\in\mathcal{P}, f\in\mathcal{O}_{i,j}, q\in\mathcal{V}, z\in\{1,\dots,n\}}$ on the basis of the previous definitions concerning the expected traveling times. In particular, by following the same line of reasoning of the definition of the coefficients α in Section 3.1, we choose β^t to minimize the following cost function

$$J_b\left(\beta^t, t\right) = \sum_{(i,j,k)\in\mathcal{P}} \sum_{f\in\mathcal{O}_{i,j}} \sum_{q\in\mathcal{V}} \sum_{z=1}^{n} \beta^t_{i,j,k,f,q,z} \log \beta^t_{i,j,k,f,q,z} + \sum_{(i,j,k)\in\mathcal{P}} \sum_{f\in\mathcal{O}_{i,j}} \sum_{q\in\mathcal{V}} \sum_{z=1}^{n} \beta^t_{i,j,k,f,q,z}\ \widehat{\omega}^t_{i,j,k,f,q,z}. \tag{21}$$

As in (11), the cost function (21) represents a trade-off between randomness and rationality.

As for the choice of α, also β^t is constrained. In particular, the value of β^t is feasible if and only if it satisfies the following conditions:

- The coefficients $\beta^t_{i,j,k,f,q,z}$ must be non-negative, i.e.,

$$\beta^t_{i,j,k,f,q,z} \geq 0 \qquad \forall (i,j,k) \in \mathcal{P},\ \forall f \in \mathcal{O}_{i,j},\ \forall q \in \mathcal{V},\ \forall z \in \{1,\dots,n\}. \tag{22}$$

- The total flow is conserved. Namely,

$$\sum_{f\in\mathcal{O}_{i,j}} \beta^t_{i,j,k,f,q,z} = 1, \qquad \forall (i,j,k) \in \mathcal{P},\ \forall q \in \mathcal{V},\ \forall z \in \{1,\dots,n\}. \tag{23}$$

- The post-change queue lengths cannot exceed their respective maximum queue lengths. Thus,

$$\widetilde{N}^t_{i,j,f} = \frac{1}{n} \sum_{q\in\mathcal{V}} \sum_{k\in\mathcal{O}_{i,j}} \sum_{z=1}^{n} N^t_{i,j,k,q} \beta^t_{i,j,k,f,q,z} \leq N^{\max}_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P}. \tag{24}$$

Therefore, the value of β^t is defined as the solution of the following optimization problem

$$\mathscr{B}(t): \begin{cases} \min J_b\left(\beta^t, t\right) \\ \text{Subject to (22), (23), and (24).} \end{cases} \tag{25}$$

This is a relative entropy optimization problem (Chandrasekaran and Shah, 2016), and it can be solved with a suitable solver. However, it is possible to notice that, if we relax the constraint (24), $\mathscr{B}(t)$ presents a closed-form solution given by Lemma 2 (the proof of Lemma 2 is in Appendix B). Furthermore, constraint (24) is active only when the queues are almost full. For these reasons, it is computationally convenient to solve $\mathscr{B}(t)$ in two steps. Firstly, we use Lemma 2 to find the solution $\widehat{\beta}^t$ by ignoring the constraint (24). Then, if $\widehat{\beta}^t$ is not feasible, instead of solving the relative entropy optimization problem $\mathscr{B}(t)$, we simply select the closest value to $\widehat{\beta}^t$ that is feasible. Using this two-steps procedure, we substitute a relative entropy optimization problem with a simpler optimization problem that is solved only when the queues are almost full.

In more detail, we propose to solve $\mathscr{B}(t)$ using the following procedure:

1. Find the solution $\widehat{\beta}^t$ of a relaxed optimization problem only subject to the constraints (22) and (23). Namely, $\widehat{\beta}^t$ is the solution of the optimization problem

$$\mathscr{B}_1(t): \begin{cases} \min J_b\left(\widehat{\beta}^t, t\right) \\ \text{Subject to (22), and (23).} \end{cases} \tag{26}$$

The optimization problem $\mathscr{B}_1(t)$ has a closed form solution as stated in Lemma 2 below. Similarly to the coefficients α, notice that $\widehat{\beta}^t_{i,j,k,f,q,z} = 0$ whenever $\boldsymbol{\Phi}_{j,f,q} = \varnothing$ because, in this case, $\rho_{i,j,f,q} = +\infty$. This makes sure that vehicles with destination q cannot end up in roads from which q is not reachable.

2. If the solution $\widehat{\beta}^t$ respects the constraint (24), then set $\beta^t = \widehat{\beta}^t$. Otherwise, select β^t by finding the closest feasible value to $\widehat{\beta}^t$. Namely, by solving the optimization problem

$$\mathscr{B}_2(t): \begin{cases} \min \sum_{(i,j,k)\in\mathcal{P}} \sum_{f\in\mathcal{O}_{i,j}} \sum_{q\in\mathcal{V}} \sum_{z=1}^{n} \left(\widehat{\beta}^t_{i,j,k,f,q,z} - \beta^t_{i,j,k,f,q,z}\right)^2 \\ \text{Subject to (22), (23), and (24).} \end{cases} \tag{27}$$

in the decision variable β^t. The optimization problem $\mathscr{B}_2(t)$ is a constrained quadratic optimization problem that can be solved efficiently. After solving $\mathscr{B}_2(t)$, if $\widehat{\beta}^t_{i,j,k,f,q,z} = 0$ then $\beta^t_{i,j,k,f,q,z} = 0$. In this way, it is guaranteed that the vehicle will not join queues from which it is impossible to reach their destination.

In summary, we first compute the coefficients β^t by solving the problem (25) (either directly or through the previous two-step procedure). This problem and its solution β^t depend on the expected waiting times as specified previously. Then, we compute the post-change queue lengths $\widetilde{N}^t_{i,j,f,q}$ according to (20). We close the section with the following lemma providing the solution of the step-one sub-problem (26).

Lemma 2. *Given $t \in \mathbb{N}$, the optimization problem $\mathscr{B}_1(t)$ has a unique solution given by*

$$\widehat{\beta}^t_{i,j,k,f,q,z} = \frac{\exp\left(-\widehat{\omega}^t_{i,j,k,f,q,z}\right)}{\sum_{p\in\mathcal{O}_{i,j}} \exp\left(-\widehat{\omega}^t_{i,j,k,p,q,z}\right)}, \qquad \forall (i,j,k)\in\mathcal{P},\ \forall f\in\mathcal{O}_{i,j},\ \forall q\in\mathcal{V},\ \forall z\in\{1,\dots,n\}.$$

4. Design of traffic-light controllers

In this section, we design two control schemes to control the traffic-lights duty cycles. First, we design a control scheme which is optimal with respect to a quadratic cost function weighting the queues length, but that does not use any model of the drivers' behavior (Section 4.4). This corresponds to a classic traffic-light control problem already well-investigated in the literature (see Section 1.2), and in the context of this article it serves as a mean for comparison (see Section 5). Next, in Section 4.5, we design a novel controller that embeds a model of the drivers' behavior. The two controllers share the same measurements, the same cost function, the same constraints, and the same basic traffic model. In turn, they only differ for a modification of the queues length taking into account the drivers' changes due to the waiting times information. Therefore, before introducing the controllers, we first present the common parts in Sections 4.1–4.3.

4.1. Measurements and cost

At each time instant $t \in \mathbb{N}$, and for each path $(i,j,f) \in \mathcal{P}$ and entry node $s \in \mathcal{L}$, both controllers measure:

- The aggregated queue length $N^t_{i,j,f}$ of the path (i,j,f) at the start of the time interval T_t.
- The aggregated entry flow at the entry node s at the end of the time interval T_t, which is given by

$$\breve{\zeta}^t_s = \sum_{q\in\mathcal{V}} \zeta^t_{s,q}.$$

- The fraction $\breve{\alpha}^t_{i,j,f}$ of the vehicles traveling along the edge (i,j) during the time interval T_t that join the path (i,j,f) at the end of the time interval T_t, which is given by

$$\breve{\alpha}^t_{i,j,f} = \begin{cases} 1 & \text{If } \sum_{k\in\mathcal{I}_{i,j}} \Lambda^t_{k,i,j} = 0 \\ \dfrac{\Lambda^t_{i,j,f}}{\sum_{k\in\mathcal{I}_{i,j}} \Lambda^t_{k,i,j}} & \text{Otherwise.} \end{cases}$$

We underline that all available measurements are aggregated because, unless an ad hoc technology is used, it is not possible to know the final destination of the vehicles. However, it is possible to measure the local decision of the drivers at each junction. Therefore, the values of $\breve{n}^t_{i,j,f}$, for all $(i,j,f) \in \mathcal{P}$ and $t \in \mathbb{N}$, are available. Since both controllers have access to all the measurements in the network, they are centralized and select the value of all the green light duty cycles g^t at the start of the time interval T_t using the measurements available.

As for what concerns the control objective, the proposed controllers are both based on (finite-horizon) Model Predictive Control (MPC) (García et al., 1989; Kouvaritakis and Cannon, 2016). At each decision time $t_c \in \mathbb{N}$, the duty cycles $g^{t_c} = (g^{t_c}_{i,j,k})_{(i,j,k)\in\mathcal{P}}$ are chosen with the aim of minimizing the cost function

$$J_c(g^{t_c}, t_c) = \sum_{t=t_c+1}^{t_c+m} \sum_{(i,j,f)\in\mathcal{P}} \left(N^t_{i,j,f}\right)^2, \tag{28}$$

in the variable g^{t_c} and subject to

$$g^t_{i,j,k} = g^{t_c}_{i,j,k} \qquad \forall (i,j,f) \in \mathcal{P},\ \forall t \in \{t_c+1, \dots, t_c+m-1\} \tag{29}$$

in which $m \in \mathbb{N}$ is the time horizon of the MPC controller, and the dependence of the terms $N^t_{i,j,f}$ from g^t is made explicit in Section 4.3 below. The constraint (29) is introduced to limit the dimension of the optimization variables in (28), as otherwise each term $N^t_{i,j,f}$ in the sum would depend on a different variable g^t. The decision times t_c, at which the duty cycles are updated, are a degree of freedom. In the following, we shall assume they are periodic with period $\Delta_c \in \mathbb{N}$, which is referred to as the *control period*. The constraint (29) suggests $\Delta_c \geq m$, although this is not necessary. In any case, taking Δ_c sufficiently large avoids changing the duty cycles too frequently, which is desirable for computational reasons and for the drivers' comfort.

The duty cycles must satisfy some additional constraints guaranteeing that vehicles driving on intersecting roads do not collide, and that the problem (28) can be solved by using the available information only. These constraints are discussed below in Sections 4.2 and 4.3 respectively.

4.2. Constraints for collisions avoidance

The traffic lights operating at the same junction are not independent to one another. Indeed, vehicles on potentially colliding paths cannot cross the intersection at the same time. For example, in the network in Fig. 1(a), the traffic lights at paths $(\mathsf{H},\mathsf{I},\mathsf{J})$ and $(\mathsf{L},\mathsf{I},\mathsf{J})$ cannot be green at the same time. Instead, the traffic lights at paths $(\mathsf{H},\mathsf{I},\mathsf{J})$ and $(\mathsf{J},\mathsf{I},\mathsf{H})$ can, since the two paths do not collide. In this section, we formalize the constraints on the values of g^t making sure that collisions are avoided.

First, we subdivide the paths into sets in such a way that each set contains only non-conflicting paths. Specifically, given a node $j \in \mathcal{V}$, the set $\mathcal{C} \subseteq \mathcal{P}_j$ is a *non-conflicting set* of the node j if and only if:

- *(consistency)* For all $(i_1,j,k_1), (i_2,j,k_2) \in \mathcal{C}$, it is possible for a flow of vehicles to travel along (i_1,j,k_1) and (i_2,j,k_2) at the same time.
- *(maximality)* For all $(i_1,j,k_1) \in \mathcal{P}_j \setminus \mathcal{C}$ and all $(i_2,j,k_2) \in \mathcal{C}$, it is not possible for a flow of vehicles to travel along (i_1,j,k_1) and (i_2,j,k_2) at the same time.

Then, we define $\mathcal{U}_j$ as the set of all the non-conflicting sets of the node j, and we denote by d_j its cardinality. Additionally, $\mathcal{U} = \bigcup_{j\in\mathcal{V}} \mathcal{U}_j$ and d is the cardinality of $\mathcal{U}$. For example, with reference to the network in Fig. 1(a), $\{(\mathsf{K},\mathsf{L},\mathsf{I}), (\mathsf{K},\mathsf{L},\mathsf{M}), (\mathsf{M},\mathsf{L},\mathsf{K})\}$ and $\{(\mathsf{K},\mathsf{L},\mathsf{I}), (\mathsf{I},\mathsf{L},\mathsf{M}), (\mathsf{I},\mathsf{L},\mathsf{K})\}$ are two valid collisions set of the node L. We underline that the same path can be an element of more than one non-conflicting set. Namely, the non-conflicting sets of a node j need not being disjoint. For example $(\mathsf{K},\mathsf{L},\mathsf{I})$ is an element of both the aforementioned non-conflicting sets. Additionally, there are $d_{\mathsf{L}} = 4$ non-conflicting sets of the node L.

Next, for each junction $j \in \mathcal{V}$, we divide the available green time into d_j slots, one for each non-conflicting set in $\mathcal{U}_j$. This requires the introduction of a further set of auxiliary variables. In particular, for each $\mathcal{C} \in \mathcal{U}_j$ and each decision time $t_c \in \mathbb{N}$, we let $b^{t_c}_{\mathcal{C}} \in [0,1]$ be the fraction of time assigned to the non-conflicting set $\mathcal{C}$ at time t_c. These variables must satisfy the following constraints

$$\sum_{\mathcal{C}\in\mathcal{U}_j} b^{t_c}_{\mathcal{C}} \leq 1, \qquad \forall j \in \mathcal{V}, \tag{30a}$$

$$\sum_{\mathcal{C}\in\mathcal{U}_j:(i,j,f)\in\mathcal{C}} b^{t_c}_{\mathcal{C}} \geq g^{t_c}_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P}, \tag{30b}$$

$$b^{t_c}_{\mathcal{C}} \geq 0, \qquad \forall \mathcal{C} \in \mathcal{U}_j,\ \forall j \in \mathcal{V}, \tag{30c}$$

$$g^{t_c}_{i,j,f} \geq g^{\min}, \qquad \forall (i,j,f) \in \mathcal{P}. \tag{30d}$$

The constraint (30a) guarantees that the sum of the fractions of time assigned to the non-conflicting sets does not exceed the maximum available time. The constraint (30b) guarantees that the duty cycle of a traffic light does not exceed the total amount of time allocated to the non-conflicting sets to which it belongs. The last two constraint bound the variables from below. Here, $g^{\min}$ is the same as in Section 2.2.

4.3. Dynamics constraints

In this section, we establish a relation between the duty cycles g^{t_c} and the future queue lengths needed for the definition of the cost function (28). Since the final destination of the drivers is not measurable and all the available measurements are aggregated, it is not possible to use the model described in Section 2 for this aim. Instead, the controllers employ a different model relating the aggregated queue lengths directly to the control variable without resorting to the disaggregated values used in the model of Section 2. This guarantees that the controllers are implementable by using only the available measurements specified in Section 4.1.

First, we assume the post-change queue lengths $\widetilde{N}^t_{i,j,f}$ are given, and that they are derived from them the other needed quantities. The computation of the post-change queue lengths is the main difference between the two controllers considered here, and it is where the model of the drivers' behavior enters into play. We postpone the discussion on their computation to Sections 4.4 and 4.5, where the two controllers are finally presented.

Firstly, given a path $(i,j,f) \in \mathcal{P}$ and a time $t \in \mathbb{N}$, the aggregated queue length at the end of the time interval T_t is obtained by adding the inbound flow $\Lambda^t_{i,j,f}$ and removing the outgoing flow $M^t_{i,j,f}$ from the post-change queue lengths $\widetilde{N}^t_{i,j,f}$. Therefore, we have

$$N^{t+1}_{i,j,f} = \widetilde{N}^t_{i,j,f} + \Lambda^t_{i,j,f} - M^t_{i,j,f}, \qquad \forall\,(i,j,f) \in \mathcal{P},\ \forall t \in \{t_c, \dots, t_c + m - 1\}. \tag{31}$$

Note that, for $t = t_c$, the value of $N^{t_c}_{i,j,f}$ is obtained from the available measurements. Next, we use the available measurements to define the aggregated inbound flow as

$$\Lambda^t_{i,j,f} = \begin{cases} \overline{\alpha}^t_{i,j,f} \displaystyle\sum_{k \in \mathcal{I}_{i,j}} M^t_{k,i,j} & \text{If } i \in \mathcal{V} \setminus \mathcal{L} \\ \overline{\zeta}^t_i & \text{If } i \in \mathcal{L} \end{cases} \qquad \forall\,(i,j,f) \in \mathcal{P},\ \forall t \in \{t_c, \dots, t_c + m - 1\}, \tag{32}$$

where $\overline{\alpha}^t_{i,j,f}$ and $\overline{\zeta}^t_i$ are estimates of the values of $\alpha^t_{i,j,f}$ and ζ^t_i, respectively. In particular, for simplicity, we assume that both values vary slowly in the m time intervals considered by the controllers, and for $(i,j,f) \in \mathcal{P}$, $s \in \mathcal{L}$, and $t \in \{t_c, \dots, t_c + m\}$, we set

$$\overline{\alpha}^t_{i,j,f} = \frac{1}{m} \sum_{\tau = t_c - m + 1}^{t_c} \breve{\alpha}^\tau_{i,j,f}, \qquad \overline{\zeta}^t_s = \frac{1}{m} \sum_{\tau = t_c - m + 1}^{t_c} \breve{\zeta}^\tau_s.$$

However, it is possible to employ more advanced predictors that consider more complex dynamics in the traffic behavior and the known typical periodicity of the traffic.

Similarly to Section 2.2, the values of the aggregated outgoing flows $M^t = (M^t_{i,j,f})_{(i,j,f) \in \mathcal{P}}$ are defined as the solution of an optimization problem that searches the maximum feasible value. In particular, a value is feasible if it satisfies the following constraints:

- All the aggregated outgoing flows have to be non-negative. Hence,

$$M^t_{i,j,f} \geq 0, \qquad \forall (i,j,f) \in \mathcal{P}. \tag{33}$$

- Only the vehicles that are queuing at the start of the time interval T_t can cross the path. Thus,

$$M^t_{i,j,f} \leq \widetilde{N}^t_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P}. \tag{34}$$

- The outgoing flow cannot exceed the effective capacity of the path. Therefore,

$$M^t_{i,j,f} \leq \upsilon_{i,j,f}\ g^{t_c}_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P}. \tag{35}$$

- The queue length at the start of the next time interval T_{t+1} cannot exceed the maximum queue length. Hence,

$$N^{t+1}_{i,j,f} \leq N^{\max}_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P} \setminus \mathcal{B}, \tag{36}$$

  where the relation between $N^{t+1}_{i,j,k}$ and M^t is defined in (31) and (32). Notice that this constraint is not defined for the entry paths because, in these cases, $N^{\max}_{i,j,f} = \infty$.

Therefore, we define the value of the aggregated outgoing flows as the solution of the optimization problem

$$\breve{\mathscr{M}}(t): \begin{cases} \max \displaystyle\sum_{(i,j,f) \in \mathcal{P}} M^t_{i,j,f} \\ \text{Subject to (33), (34), (35), and (36).} \end{cases} \tag{37}$$

This is a linear optimization problem that is always feasible because the solution $M^t_{i,j,f} = 0$, $\forall (i,j,f) \in \mathcal{P}$, always satisfies the constraints. Hence, the solution is a function of g^{t_c}, $\widetilde{N}^t = (\widetilde{N}^t_{i,j,f})_{(i,j,f)\in\mathcal{P}}$ and t. Denoting this function with Y, we have[3]

$$M^t = Y\left(g^{t_c}, \widetilde{N}^t, t\right) \qquad \forall t \in \{t_c, \dots, t_c + m - 1\}, \tag{38}$$

4.4. Control without a model for the drivers' behavior

In this section, we present the first control scheme, which optimizes the cost function (28) without using any model of the drivers' behavior. In this case, therefore, we simply have

$$\widetilde{N}^t_{i,j,f} = N^t_{i,j,k}, \qquad \forall (i,j,f) \in \mathcal{P},\ \forall t \in \{t_c, \dots, t_c + m - 1\}, \tag{39}$$

which completes the equations presented in previous Section 4.3 for the first controller. Then, the controller is obtained as a solution to the following optimization problem, with cost function (28) and collecting all the constraints introduced in Sections 4.1, 4.2, 4.3, and 4.4.

$$\mathscr{N}'\left(t_c\right) : \begin{cases} \min J_c\left(g^{t_c}, t_c\right) \\ \text{Subject to (29), (30), (31), (32), (38), and (39).} \end{cases} \tag{40}$$

in the variables g^{t_c} and $b^{t_c} = (b^{t_c}_C)_{C\in\mathcal{U}_j, j\in\mathcal{V}}$. Here, we can observe that the constraint (38) is the only nonlinear constraint. For computational convenience, we consider the following linear constraints

$$M^t_{i,j,f} \geq 0, \qquad \forall (i,j,f) \in \mathcal{P},\ \forall t \in \{t_c, \dots, t_c + m - 1\}, \tag{41a}$$

$$M^t_{i,j,f} \leq \widetilde{N}^t_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P},\ \forall t \in \{t_c, \dots, t_c + m - 1\}, \tag{41b}$$

$$M^t_{i,j,f} \leq v_{i,j,f}\ g^{t_c}_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P},\ \forall t \in \{t_c, \dots, t_c + m - 1\}, \tag{41c}$$

$$N^t_{i,j,f} \leq N^{\max}_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P} \setminus \mathcal{B},\ \forall t \in \{t_c + 1, \dots, t_c + m\}. \tag{41d}$$

Then, we define the following relaxed optimization problem

$$\mathscr{N}\left(t_c\right) : \begin{cases} \min J_c\left(g^{t_c}, t_c\right) - \varepsilon \displaystyle\sum_{t=t_c}^{t_c+m-1} \sum_{(i,j,f)\in\mathcal{P}} M^t_{i,j,f}, \\ \text{Subject to (29), (30), (31), (32), (39), and (41).} \end{cases} \tag{42}$$

in the decision variables g^{t_c}, b^{t_c} and $(M^t)_{t\in\{t_c,\dots,t_c+m-1\}}$. In $\mathscr{N}\left(t_c\right)$, the nonlinear constraint (38) is replaced with the linear constraints (41) and the cost function is obtained by summing to the cost function (28) a quadratic term that minimize the difference between the outgoing flows and their respective upper bounds weighted with the parameter $\varepsilon \in [0, \infty)$. Therefore, $\mathscr{N}\left(t_c\right)$ aims to maximize the outgoing flows subject to their feasibility constraints and to minimize the cost function (28).

Since the constraints on g^{t_c} and b^{t_c} of $\mathscr{N}'\left(t_c\right)$ and $\mathscr{N}\left(t_c\right)$ are the same, a feasible point (g^{t_c}, b^{t_c}) for $\mathscr{N}\left(t_c\right)$ is also feasible for $\mathscr{N}'\left(t_c\right)$ and vice versa. However, an optimal solution of $\mathscr{N}\left(t_c\right)$ is not necessary an optimal solution of $\mathscr{N}'\left(t_c\right)$. But $\mathscr{N}\left(t_c\right)$ is a convex quadratic optimization problem with linear constraints that can be solved efficiently using standard solvers.

4.5. Control that considers a model for the drivers' behavior

In this section, we describe the second control scheme that optimizes the cost function (28) by considering a model of the drivers' reaction to the length of the queues. Since it is impossible to discern the destination of the drivers in the queues, we cannot use the same model described in Section 3.2. In particular, it is not possible to assess the crossing time defined in (10) and, therefore, the traveling time. For this reason, in this context, we derive a model where the drivers only react to the waiting time in the queues.

Consider a vehicle that is in position $x \in [0, N^t_{i,j,k}]$ in the queue $(i,j,k) \in \mathcal{P}$ at the start of the time interval T_t. As explained in Section 3.2, the vehicle have to select one queue among the possible alternatives $\{(i,j,f)\}_{f\in\mathcal{O}_{i,j}}$. Following the same reasoning of Section 3.2, we assign a weight to each possible decision that only depends on the waiting time of the queues. In particular, for each possible alternative queue (i,j,f), we define the weight

$$\breve{\omega}^t_{i,j,k,f}(x) := \begin{cases} \breve{\xi}\widehat{w}^t_{i,j,f}(x) - \breve{\sigma} & \text{If } k = f \\ \breve{\xi}\widehat{w}^t_{i,j,f}(x + \breve{\eta}) & \text{If } k \neq f \wedge x + \breve{\eta} < N^t_{i,j,f} \\ \breve{\xi}\widehat{w}^t_{i,j,f}\left(N^t_{i,j,f}\right) & \text{If } k \neq f \wedge x + \breve{\eta} \geq N^t_{i,j,f}, \end{cases} \tag{43}$$

where $\widehat{w}_{i,j,f}$ is the perceived waiting time of the queue (i,j,f), as defined in (17), and $\breve{\xi}, \breve{\sigma}, \breve{\eta} \in [0, +\infty)$ are three tuning knobs of the controller that have the same meaning as the three parameters ξ, σ and η, respectively.

[3] For simplicity, we assume that the solution of $\mathscr{M}(t)$ is always unique. Otherwise, the function Y returns one of the possible solutions arbitrarily.

Then, given $f \in \mathcal{O}_{i,j}$, we define $\breve{\omega}^t_{i,j,k,f,z}$ as the mean weight among the vehicles that are in the queue section $S^t_{i,j,k,z}$, as defined in (19). In particular, we have

$$\breve{\omega}^t_{i,j,k,f,z} := \frac{1}{\frac{N^t_{i,j,k}}{n}} \int_{S^t_{i,j,k,z}} \breve{\omega}^t_{i,j,k,f}(x)\ dx$$

$$= \begin{cases} \dfrac{\breve{\xi} h}{v_{i,j,f}\, \widehat{g}^t_{i,j,f}} \dfrac{N^t_{i,j,k}\,(2z-1)}{2n} - \breve{\sigma} & \text{If } f = k \\ \dfrac{\breve{\xi} h}{v_{i,j,f}\, \widehat{g}^t_{i,j,f}} \left(\dfrac{N^t_{i,j,k}\,(2z-1)}{2n} + \breve{\eta} \right) & \text{If } f \neq k \wedge z \dfrac{N^t_{i,j,k}}{n} + \breve{\eta} < N^t_{i,j,f} \\ \dfrac{\breve{\xi} h}{v_{i,j,f}\, \widehat{g}^t_{i,j,f}} N^t_{i,j,f} & \text{If } f \neq k \wedge (z-1) \dfrac{N^t_{i,j,k}}{n} + \breve{\eta} \geq N^t_{i,j,f} \\ \dfrac{\breve{\xi} h}{v_{i,j,f}\, \widehat{g}^t_{i,j,f}} \left(z \left(N^t_{i,j,f} - \breve{\eta} \right) + \breve{\eta} - \dfrac{(z-1)^2\, N^t_{i,j,k}}{2n} - \dfrac{n \left(N^t_{i,j,f} - \breve{\eta} \right)^2}{2 N^t_{i,j,k}} \right) & \text{Otherwise.} \end{cases}$$

Now, if we denote by $\breve{\beta}^t_{i,j,k,f,z} \in [0,1]$ the fraction of vehicles in section $S^t_{i,j,k,z}$, $(i,j,k) \in \mathcal{P}$ and $z \in \{1, \dots, n\}$, that decide to join the queue (i,j,f) at time t, for some $f \in \mathcal{O}_{i,j}$, the post-change queue lengths are given as follows

$$\widetilde{N}^t_{i,j,f} = \frac{1}{n} \sum_{k \in \mathcal{O}_{i,j}} \sum_{z=1}^{n} N^t_{i,j,k} \breve{\beta}^t_{i,j,k,f,z}, \qquad \forall (i,j,f) \in \mathcal{P},\ \forall t \in \{t_c, \dots, t_c + m - 1\}. \tag{44}$$

Now, following the reason of Section 3.2, we define the values of the coefficients $\breve{\beta}^t = \left(\breve{\beta}^t_{i,j,k,f,z} \right)_{(i,j,k) \in \mathcal{P}, f \in \mathcal{O}_{i,j}, z \in \{1,\dots,n\}}$ as the one that minimize the cost function

$$\breve{J}_b\left(\breve{\beta}^t, t\right) = \sum_{(i,j,k) \in \mathcal{P}} \sum_{f \in \mathcal{O}_{i,j}} \sum_{z=1}^{n} \breve{\beta}^t_{i,j,k,f,z} \log \breve{\beta}^t_{i,j,k,f,z} + \sum_{(i,j,k) \in \mathcal{P}} \sum_{f \in \mathcal{O}_{i,j}} \sum_{z=1}^{n} \breve{\beta}^t_{i,j,k,f,z}\ \breve{\omega}^t_{i,j,k,f,z}. \tag{45}$$

As in (11) and in (21), the cost function (45) represents a trade-off between randomness and rationality. The values of $\breve{\beta}^t$ are constrained with the same reasoning used for constraining β^t. In particular, $\breve{\beta}^t$ has to satisfy the following conditions:

- The coefficients $\breve{\beta}^t_{i,j,k,f,z}$ must be non-negative, i.e.,

$$\breve{\beta}^t_{i,j,k,f,z} \geq 0 \qquad \forall (i,j,k) \in \mathcal{P},\ \forall f \in \mathcal{O}_{i,j},\ \forall z \in \{1, \dots, n\}. \tag{46}$$

- The total flow is conserved. Namely,

$$\sum_{f \in \mathcal{O}_{i,j}} \breve{\beta}^t_{i,j,k,f,z} = 1, \qquad \forall (i,j,k) \in \mathcal{P},\ \forall z \in \{1, \dots, n\}. \tag{47}$$

- The post-change queue lengths cannot exceed their respective maximum queue lengths. Thus,

$$\widetilde{N}^t_{i,j,f} = \frac{1}{n} \sum_{k \in \mathcal{O}_{i,j}} \sum_{z=1}^{n} N^t_{i,j,k,q} \breve{\beta}^t_{i,j,k,f,z} \leq N^{\max}_{i,j,f}, \qquad \forall (i,j,f) \in \mathcal{P}. \tag{48}$$

Therefore, the value of $\breve{\beta}^t$ is defined as the solution of the following optimization problem

$$\breve{\mathscr{B}}(t) : \begin{cases} \min \breve{J}_b\left(\breve{\beta}^t, t\right) \\ \text{Subject to (46), (47), and (48).} \end{cases} \tag{49}$$

This is a convex optimization problem with linear constraints, and therefore it has a unique solution. Hence, the value of the unique solution $\breve{\beta}^t$ is a function of g^{t_c}, $N^t = (N^t_{i,j,k})_{(i,j,k) \in \mathcal{P}}$ and t. Denoting this function with Ψ, we have

$$\breve{\beta}^t = \Psi\left(g^{t_c}, N^t, t\right) \qquad \forall t \in \{t_c, \dots, t_c + m - 1\}, \tag{50}$$

which completes the equations presented in Section 4.3 for the second controller. Then, the second controller is obtained by minimizing the cost function (28) constrained to all the constraints described in Sections 4.1, 4.2, 4.3, and 4.5.

$$\breve{\mathscr{W}}'\left(t_c\right) : \begin{cases} \min J_c\left(g^{t_c}, t_c\right) \\ \text{Subject to (29), (30), (31), (32), (38), (44), and (50).} \end{cases} \tag{51}$$

in the decision variables g^{t_c} and b^{t_c}.

The constraint (50) is nonlinear and the function Ψ is only defined as the solution of the optimization problem $\breve{\mathscr{B}}(t)$. Therefore, it is not trivial to find the solution of the optimization problem $\breve{\mathscr{W}}'\left(t_c\right)$. In order to find this solution, it is useful to define two additional optimization problems:

- A simplified version of $\mathscr{W}'(t_c)$ where the value of $\breve{\beta} = (\breve{\beta}^t)_{t\in\{t_c,\dots,t_c+m-1\}}$ is given instead of being constrained to the other variables. Hence, we define

$$\overline{\mathscr{W}}'(\breve{\beta}, t_c) : \begin{cases} \min J_c\left(g^{t_c}, t_c\right) \\ \text{Subject to (29), (30), (31), (32), (38), and (44).} \end{cases} \tag{52}$$

  in the decision variables g^{t_c}, b^{t_c} and $(M^t)_{t\in\{t_c,\dots,t_c+m-1\}}$. We define Φ as the function that, given a value of $\breve{\beta}$ and t_c, returns the optimal g^{t_c} according to the optimization problem $\overline{\mathscr{W}}'(\breve{\beta}, t_c)$.
- An optimization problem whose solution is the value of $\breve{\beta}$ given a certain value of g^{t_c}. Therefore, we have

$$\overline{\mathscr{B}}(g^{t_c}, t_c) : \begin{cases} \min \sum_{t=t_c}^{t_c+m-1} \breve{J}_b\left(\breve{\beta}^{t_c}, t_c\right) \\ \text{Subject to (31), (32), (38), (44), (46), (47), and (48).} \end{cases} \tag{53}$$

  in the decision variables $\breve{\beta}$. This optimization problem merges the m optimization problems $\breve{\mathscr{B}}(t)$, with $t \in \{t_c, \dots, t_c + m - 1\}$, where the value of the queue lengths are constrained by the dynamic model described in (31), (32), (38) and (44). We define Γ as the function that, given a value of g^{t_c} and t_c, returns the optimal $\breve{\beta}$ according to the optimization problem $\overline{\mathscr{B}}(g^{t_c}, t_c)$.

Now, it is possible to note that the optimization problem $\mathscr{W}'(t_c)$ is a bi-level optimization problem that interconnects $\overline{\mathscr{W}}'(\breve{\beta}, t_c)$ and $\overline{\mathscr{B}}(g^{t_c}, t_c)$. Therefore, an optimal solution $g_\star^{t_c}$ of $\mathscr{W}'(t_c)$ satisfies

$$g_\star^{t_c} = \Phi\left(\Gamma\left(g_\star^{t_c}, t_c\right), t_c\right).$$

Therefore, $g_\star^{t_c}$ is a fixed-point of the function $\Phi\left(\Gamma\left(\cdot, t_c\right), t_c\right)$. Hence, it is possible to use a fixed-point algorithm to find the solution of $\mathscr{W}'(t_c)$. In particular, starting from an arbitrary initial condition $\breve{\beta}$:

FP1. We solve $\overline{\mathscr{W}}'(\breve{\beta}, t_c)$, and we assign $g_\star^{t_c} \leftarrow \Phi(\breve{\beta}, t_c)$;
FP2. We solve $\overline{\mathscr{B}}(g_\star^{t_c}, t_c)$, and we assign $\breve{\beta} \leftarrow \Gamma(g_\star^{t_c}, t_c)$;
FP3. If a stopping criterion is not satisfied, return to **FP1**, otherwise, the optimal solution is $g_\star^{t_c}$.

The step **FP2** can be achieved by sequentially solving the optimization problems $\breve{\mathscr{B}}(t)$, with $t \in \{t_c, \dots, t_c + m - 1\}$, by using the dynamic model imposed by the constraints (31), (32), (38) and (44). Furthermore, the optimization problem $\breve{\mathscr{B}}(t)$ can be solved using a similar two steps procedure to the one proposed in Section 3.2 for the solution of $\mathscr{B}(t)$ as explained in detail in Section 4.6. Instead, the step **FP1** requires solving the nonlinear optimization problem $\overline{\mathscr{W}}'(\breve{\beta}, t_c)$. Since the constraint (38) is the only nonlinear constraint, it is possible to solve $\overline{\mathscr{W}}'(\breve{\beta}, t_c)$ by employing the same reasoning used for the solution of $\mathscr{N}'(t_c)$. Therefore, we define the relaxed optimization problem

$$\overline{\mathscr{W}}(\breve{\beta}, t_c) : \begin{cases} \min J_c\left(g^{t_c}, t_c\right) - \varepsilon \sum_{t=t_c}^{t_c+m-1} \sum_{(i,j,f)\in\mathcal{P}} M_{i,j,f}^t, \\ \text{Subject to (29), (30), (31), (32), (41), and (44).} \end{cases} \tag{54}$$

in the decision variables g^{t_c}, b^{t_c} and $(M^t)_{t\in\{t_c,\dots,t_c+m-1\}}$. This is a convex quadratic optimization problem with linear constraints that can be solved using a suitable solver.

4.6. Computational analysis of the controller

The controller described in Section 4.5 is computed using a fixed-point algorithm. Therefore, its computational complexity depends on two factors: **(i)** the convergence rate of the fixed-point procedure, and **(ii)** the complexity of computing the function $\Phi\left(\Gamma\left(\cdot, t_c\right), t_c\right)$. Regarding the first factor, fixed-point algorithms are well-studied in the literature (Agarwal et al., 2001), and their convergence properties can be characterized under different conditions. For instance, the Banach fixed-point theorem (Agarwal et al., 2001, Theorem 1.1) leads to a simple iterative algorithm that converges if the function $\Phi\left(\Gamma\left(\cdot, t_c\right), t_c\right)$ is a contraction, which is the case, for example, when the functions $\Phi(\cdot, t_c)$ and $\Gamma(\cdot, t_c)$ are Lipschitz and the product of their Lipschitz constants is less than 1. Nevertheless, a formal proof of existence of fixed-point algorithms for the specific case considered here (namely, for the map $\Phi\left(\Gamma\left(\cdot, t_c\right), t_c\right)$) with proved guarantees and convergence rate is still missing, and we leave it for future investigation. In practical implementations, one can set a maximum value for the number of iterations. This leads to suboptimal solutions but avoids unbounded computation times.

The second factor, i.e., the complexity of computing the function $\Phi\left(\Gamma\left(\cdot, t_c\right), t_c\right)$, requires the computation of the function $\Phi\left(\cdot, t_c\right)$ (Step **FP1** above) and the function $\Gamma\left(\cdot, t_c\right)$ (Step **FP2** above). In the remainder of the section, we discuss the complexity of these two steps. First, we focus on Step **FP1**, which requires the solution of the optimization problem $\overline{\mathscr{W}}(\breve{\beta}, t_c)$ in Eq. (54). As explained before, this is a convex quadratic optimization problem with linear equality constraints and linear inequality constraints. This type of quadratic optimization problems are well known and studied in the literature (Boyd and Vandenberghe, 2019; Bomze, 1998; Xu, 2003) and efficient algorithms are available (Coleman and Li, 1996; Andersen et al., 2003; Fletcher, 1971; Ye and Tse, 1989).

Additionally, in this case, there are $(m+1)n_p + d$ optimization variables, $4\ mn_p$ equality constraints and $n_v + d + 2n_p + (4n_p - n_b)m$ inequality constraints. Therefore, the dimension of the problem scales linearly with the number of paths in the network n_p and the MPC time horizon m. Similarly, the number of constraints also scales linearly with the number n_v of junctions and the number d of non-conflicting sets. Therefore, the step **FP1** can be solved efficiently with well known procedures. In particular, there exist algorithm that finds the solution in polynomial time complexity with respect to the decision variable (Ye and Tse, 1989).

Step **FP2** requires the solution of the optimization problem $\overline{\mathscr{B}}\left(g^{t_c}, t_c\right)$ defined in Eq. (53). This optimization problem can be solved iteratively by relying on the value of $\check{\beta}^t$ (and N^t) to compute the value of $\check{\beta}^{t+1}$ (and N^{t+1}). In particular, starting with $t = t_c$, we can use the following procedure:

FP2.A. Compute $\check{\beta}^t$ by solving the optimization problem $\check{\mathscr{B}}(t)$ describe in (49);
FP2.B. Compute $\widetilde{N}^t_{i,j,f}$, for all $(i,j,f) \in \mathcal{P}$, according to (44);
FP2.C. Compute M^t by solving the optimization problem $\check{\mathscr{M}}(t)$ described in (37);
FP2.D. Compute $\Lambda^t_{i,j,f}$, for all $(i,j,f) \in \mathcal{P}$, according to (32);
FP2.E. Compute $N^{t+1}_{i,j,f}$, for all $(i,j,f) \in \mathcal{P}$, according to (31);
FP2.F. If $t < t_c + m$ return to **FP2.A.** with $t \leftarrow t+1$.

The only two critical steps are **FP2.A.** and **FP2.C.** that require the solution of an optimization problem. The optimization problem $\check{\mathscr{B}}(t)$ is a relative entropy optimization problem (Chandrasekaran and Shah, 2016) similar to the optimization problem $\mathscr{B}(t)$. Hence, it can be substituted by the same two-steps procedure used for the solution of $\mathscr{B}(t)$ detailed below:

1. Find the solution $\hat{\check{\beta}}^t$ of a relaxed optimization problem only subject to the constraints (46) and (47). In particular, $\hat{\check{\beta}}^t$ is the solution of
$$\check{\mathscr{B}}_1(t): \begin{cases} \min \check{J}_b\left(\check{\beta}^t, t\right) \\ \text{Subject to (46), and (47).} \end{cases} \tag{55}$$
Similarly to $\mathscr{B}_1(t)$, $\check{\mathscr{B}}_1(t)$ has a closed form solution as stated by Lemma 3 below.
2. If the solution $\hat{\check{\beta}}^t$ respects the constraint (48), then set $\check{\beta}^t = \hat{\check{\beta}}^t$. Otherwise, select β^t as the closest feasible value to $\widehat{\beta}^t$, which can be obtained by solving the following optimization problem
$$\check{\mathscr{B}}_2(t): \begin{cases} \min \displaystyle\sum_{(i,j,k)\in\mathcal{P}} \sum_{f\in\mathcal{O}_{i,j}} \sum_{z=1}^{n} \left(\hat{\check{\beta}}^t_{i,j,k,f,z} - \check{\beta}^t_{i,j,k,f,z}\right)^2 \\ \text{Subject to (46), (47), and (48).} \end{cases} \tag{56}$$
in the decision variable $\check{\beta}^t$.

As explained in Section 3.2 in the solution of $\mathscr{B}(t)$ (see the discussion below (25)), the constraint (48) is active when the queues are nearly full. If this is not the case, then only the first step of the above procedure suffices to compute $\check{\beta}^t$ and, hence, step **FP2.A.** can be solved in closed form according to Lemma 3. Otherwise, one needs to solve (56). Nevertheless, this is a quadratic optimization problem with linear inequality and equality constraints and whose dimension scales linearly with the number of paths n_p. As explained before, this type of quadratic optimization problems are well known and studied in the literature and efficient solutions exist (Boyd and Vandenberghe, 2019; Bomze, 1998; Xu, 2003; Coleman and Li, 1996; Andersen et al., 2003; Fletcher, 1971; Ye and Tse, 1989).

We conclude the discussion of step **FP2.A.** with the following Lemma which characterizes the closed form solution for $\check{\mathscr{B}}_1(t)$.

Lemma 3. *Given $t \in \mathbb{N}$, the optimization problem $\check{\mathscr{B}}_1(t)$ has a unique solution given by*

$$\hat{\check{\beta}}^t_{i,j,k,f,z} = \frac{\exp\left(-\check{\omega}^t_{i,j,k,f,z}\right)}{\displaystyle\sum_{p\in\mathcal{O}_{i,j}} \exp\left(-\check{\omega}^t_{i,j,k,f,z}\right)}, \qquad \forall (i,j,k) \in \mathcal{P},\ \forall f \in \mathcal{O}_{i,j},\ \forall z \in \{1,\dots,n\}.$$

The second critical step for the computation of **FP2** is **FP2.C.**, which requires the solution of the optimization problem $\check{\mathscr{M}}(t)$. This is linear programming problem with n_p optimization variables and $4n_p - n_b$ inequality constraints. This is a well known problem (Boyd and Vandenberghe, 2019; Bertsimas and Tsitsiklis, 1997), and it can be easily solved using the Simplex method with a polynomial computational complexity (Bertsimas and Tsitsiklis, 1997, Table 3.1). Furthermore, in this particular case, Simplex methods have complexity $O(n_p^2)$.

We conclude this section by remarking that the proposed controller is meant to be used with moderately large periods, e.g., between 30 min and an hour, and it is only meant to be a proof of concept that show the beneficial effects of showing the ERTD to the drivers and by exploiting their rational behavior. Therefore, the computational efficiency of the controller is not a central focus of this work, and improved solutions are left to future developments. In particular, the computational complexity of the proposed method scales with the number of paths n_p in the network. Therefore, it can be greatly improved by decentralizing the procedure in such a way that each junction can act independently of each other like it is commonly done in the literature (Zhao et al., 2011; Le et al., 2015; Eom and Kim, 2020).

5. Numerical results

In this section, we present some numerical simulations illustrating, as a proof of concept, the possible impact of showing to the drivers the expected waiting time and the advantages of using the control policy proposed in Section 4.5 that takes the drivers' reaction into account. In particular, we compare the optimal controller presented in Section 4.4 with **(i)** the same controller where, however, the expected waiting times are shown to the drivers, and **(ii)** the controller proposed in Section 4.5, which displays the expected waiting times and embeds a model of the drivers' behavior. The first comparison allows us to underline the impact of showing the expected waiting times also when using classical control policies. The second, instead, shows the advantages of a controller that embeds a model of the drivers' reaction. In all the simulations, the linear and quadratic programming problems are solved using YALMIP (Löfberg, 2004) equipped with MOSEK (Mosek APS, 2022). Moreover, we use the term **NC** (No Change) to refer to the control policy presented in Section 4.4, and **WC** (With Changes) to refer to the controller presented in Section 4.5. The fixed-point algorithm of controller **WC** stops if the difference between the values of $\check{\beta}$ of the last two steps is smaller than a threshold or, if the procedure do not converge, after 10 iteration.

In the first set of simulations, we consider the network of Fig. 1(a) with maximum queue length $N^{\max}_{i,j,f} = 80$ and maximum capacity $v_{i,j,f} = 20$ for all $(i,j,f) \in \mathcal{P}$ (Section 2.2), and subject to a traffic input equal to

$$\begin{aligned} \zeta^t_{\mathsf{a},\mathsf{J}} &= \zeta^t_{\mathsf{g},\mathsf{J}} = 5 + 15\exp\left(-\frac{(t-90)^2}{150^2}\right) + 20\exp\left(-\frac{(t-250)^2}{60^2}\right) + \sin\left(\frac{2\pi t}{5}\right), \\ \zeta^t_{i,q} &= 0, \qquad \forall (i,q) \in \mathcal{V}^2 \setminus \{(\mathsf{a},\mathsf{J}),(\mathsf{g},\mathsf{J})\}. \end{aligned} \tag{57}$$

The graph of $\zeta^t_{\mathsf{a},\mathsf{J}}$ and $\zeta^t_{\mathsf{g},\mathsf{J}}$ is shown in Fig. 2 (profile 0). This input profile models a slowly increasing traffic flow with a final peak then fading away. Regarding the non-conflicting sets, for each node $j \in \mathcal{V}$, we call two paths $(i,j,k),(i',j,k') \in \mathcal{P}_j$ not compatible if either $k = k'$, or the corresponding arrows in Fig. 1(a) intersect. Moreover, we let $\gamma_{\mathsf{A},\mathsf{D},\mathsf{H}} = \gamma_{\mathsf{E},\mathsf{D},\mathsf{A}} = \gamma_{\mathsf{E},\mathsf{D},\mathsf{H}} = 1/2$, $\gamma_{i,j,k} = 1/3$ for all $(i,j,k) \in \mathcal{P}_j$ with $j \in \{\mathsf{H},\mathsf{I},\mathsf{J},\mathsf{L}\}$, and we set all the remaining values to 1. The traffic evolution is modeled according to Sections 2 and 3, in which we set $h = 1$, $\mu = 5$ in (10), $\xi = 4$, $\sigma = 1/2$, and $\eta = 2$ in (18), and $n = 30$ in (19). The choice $h = 1$ normalizes the time units to the value of the integration step h, which corresponds to about 5 min. In all the simulations, the controllers start working at time $t = 40$.

Fig. 3 shows a comparison between the **NC** controller of Section 4.4 with and without showing the expected waiting times, and the controller **WC** of Section 4.5, which shows the waiting time information and also considers the drivers' reaction to it. For completeness, we also simulate the **NC** controller without waiting-time information with a lager value of $g^{\min}$. Indeed, if the information on the waiting times is not shown, it is licit to expect larger values for $g^{\min}$. As expected, the simulations show a considerable gain in showing the expected waiting times also with the same controller **NC**. Moreover, the simulations also show an additional advantage in using the control policy **WC** that takes the drivers' reaction into account. In particular, the table in Fig. 4 reports the value of the duty cycle, averaged on the interval corresponding to the peak of the incoming traffic, of some crucial traffic lights. As shown by the figure, and with reference to the network of Fig. 1(a), the controller **WC** blocks as much as possible the traffic flow in the path $(\mathsf{G},\mathsf{H},\mathsf{I})$, and diverts the traffic coming from g toward node K. This permits to better handle the traffic coming from node a. In addition, it also sets $g_{\mathsf{A},\mathsf{D},\mathsf{E}} = 1$, thus maximizing the flow taking the alternative road $(\mathsf{a},\mathsf{A},\mathsf{D},\mathsf{E},\mathsf{B},\mathsf{C},\mathsf{F},\mathsf{J})$ to node J. Finally, Fig. 3 also shows that larger values of $g^{\min}$ degrade the performances. This further supports the idea of showing the waiting-time information, as it allows one to use lower values of $g^{\min}$.

Figs. 5, 6, 7, and 8 show the comparison of the same controllers simulated in Fig. 3 in which, however, the inputs $\zeta^t_{\mathsf{a},\mathsf{J}}$ and $\zeta^t_{\mathsf{n},\mathsf{J}}$ equal, respectively, to the profiles 1, 2, 3, and 4 in Fig. 2. Profile 1 models a rush-hour scenario with a large initial peak and a slow decay. Profile 2 models a train of step-wise pulses. Profile 3 models a commuting scenario by means of a large and narrow peak of incoming flow. Finally, profile 4 models a sudden change in the incoming flow followed by a steady decay. In all the simulations, the **WC** controller performs better than the **NC** controller with expected waiting-time information, which, in turn, performs better than the same controller without such information.

Fig. 9 shows the performance of the **WC** controller for different values of the traffic parameters ξ and σ in (18) and with matching coefficients $(\check{\xi},\check{\sigma}) = (\xi,\sigma)$. All the other parameters are the same as specified previously. The simulations show that, when the **WC** controller matches the traffic parameters, the performance of the control policy is not very sensitive to their actual value. Fig. 10 shows the performance of the **WC** controller for different values of $(\check{\xi},\check{\sigma})$ not matching the model's parameters (ξ,σ), kept fixed to $(\xi,\sigma) = (4,1/2)$ as before. The range of values simulated goes from -87% to $+200\%$ of the nominal value for $\check{\xi}$, and from -90% to $+200\%$ of the nominal value of $\check{\sigma}$. In all cases, the performance are better than the **NC** without waiting-times information. Fig. 10 suggests that overestimating ξ is better than underestimating it.

Fig. 11 shows the same controller **WC** of Fig. 3 for different values of $N^{\max}_{i,j,f}$. As $N^{\max}_{i,j,f}$ is taken the same for every path $(i,j,f) \in \mathcal{P}$, we drop the dependency on it. As the figure shows, the sensitivity of the controller with respect to $N^{\max}$ is low, and better performances are obtained for smaller values of $N^{\max}$.

Fig. 12 shows a comparison between the controllers **NC** (showing the waiting-time information) and **WC** for different values of the control horizon m. As shown in the figure, the controller **WC** is not much sensitive to the choice of m, whereas **NC** shows a higher variance in the performance. In particular, the performance of **NC** with $m = 5$ or $m = 10$ are close to that of the same controller without showing the waiting-times information.

Finally, Fig. 13 show a comparison between the same controllers of Fig. 3 for the network shown in Fig. 1(b), in which an additional input flow from node n is added. Also in this case, **WC** outperforms **NC**. Fig. 14 shows the mean value on the interval $[150,350]$ of some relevant traffic-light duty cycles. As shown by the figure, the **WC** controller and the **NC** controller showing the

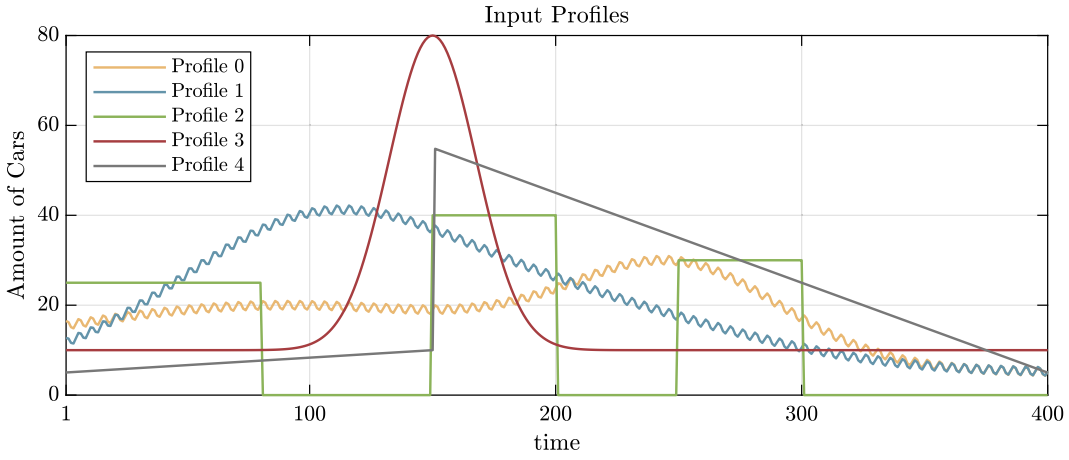


Fig. 2. Traffic input profiles.

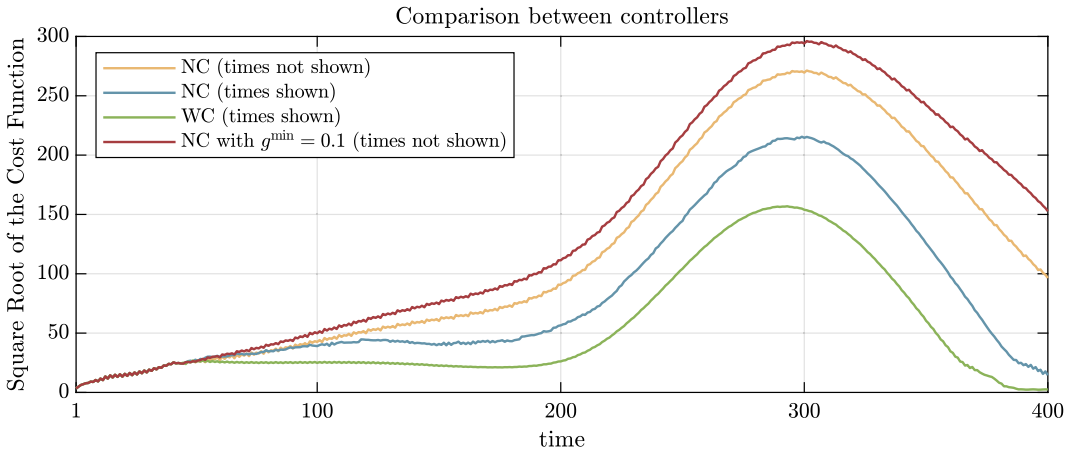


Fig. 3. Comparison between **(i)** controller **NC** with $g^{\min} = 10^{-2}$ and $\widehat{g}^{t}_{i,j,f} = \gamma_{i,j,f}$ for all $(i,j,f) \in \mathcal{P}$ (yellow); **(ii) NC** with $g^{\min} = 10^{-2}$ and $\widehat{g}^{t}_{i,j,f} = g^{t}_{i,j,f}$ for all $(i,j,f) \in \mathcal{P}$ (blue); **(iii) WC** with $g^{\min} = 10^{-2}$, $\widehat{g}^{t}_{i,j,f} = g^{t}_{i,j,f}$ for all $(i,j,f) \in \mathcal{P}$, and $(\check{\xi}, \check{\sigma}, \check{\eta}) = (\xi, \sigma, \eta) = (4, 1/2, 2)$ (green); **(iv) NC** with $g^{\min} = 10^{-1}$ and $\widehat{g}^{t}_{i,j,f} = \gamma_{i,j,f}$ for all $(i,j,f) \in \mathcal{P}$ (red). In ordinate the square root of the cost function (28). For all controllers, $\Delta_c = m = 3$ (Section 4.1). In all the simulations, the input is given by (57) (Profile 0 in Fig. 2). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

waiting-times information direct the traffic coming from n to the path $(\mathsf{n}, \mathsf{N}, \mathsf{D}, \mathsf{H}, \mathsf{I}, \mathsf{J})$, the traffic from a toward the longer path $(\mathsf{a}, \mathsf{A}, \mathsf{D}, \mathsf{E}, \mathsf{B}, \mathsf{C}, \mathsf{F}, \mathsf{J})$, and the traffic coming from g toward node K. In this way, these controllers divide the traffic into three different non-intersecting paths, thus optimizing the flow.

6. Conclusions

In this work, we have presented a proof of concept: showing to the drivers the information about the amount of time they need to wait while they are in a queue can yield significant improvement in the traffic behavior. In particular, we have developed a macroscopic model where the drivers behave rationally and try to reach a specific destination reacting to the information provided in the queues. This new model was employed to show that the additional information about the *Expected Red Time Duration* considerably decreases the length of the queues in the network. Furthermore, we designed a controller that exploits the driver reaction to the provided information obtaining an additional improvement. Therefore, we show that it is possible to decrease the congestion in the traffic network by exploiting the rational behavior of the drivers and by providing them only local information about the traffic lights.

We focused on the specific case where the only information provided to the drivers is local to their position in the network, and they integrate it with their experience in the network to make a decision.

In future development, this proof of concept will be extended to other information structures in which drivers have access to more global information using V2I communication. Furthermore, future research efforts will focus on the important theoretical and practical issues of providing stability guarantees and developing a decentralized implementation.

Traffic Light Table

	$g_{A,D,H}$	$g_{A,D,E}$	$g_{G,H,I}$	$g_{G,H,K}$	$g_{D,H,I}$	$g_{D,H,K}$
NC (times not shown)	0.99	0.5083	0.5453	0.4086	0.4447	0.09957
NC (times shown)	0.99	0.4427	0.5028	0.5625	0.4872	0.2096
WC (times shown)	0.99	1	0.01999	0.98	0.97	0.01
NC with $g^{\min} = 0.1$ (times not shown)	0.9	0.553	0.4497	0.4379	0.4503	0.1144

Fig. 4. Table showing the mean value in the interval $[200, 350]$ of the duty cycles for some sample lights obtained in the simulations of Fig. 3.

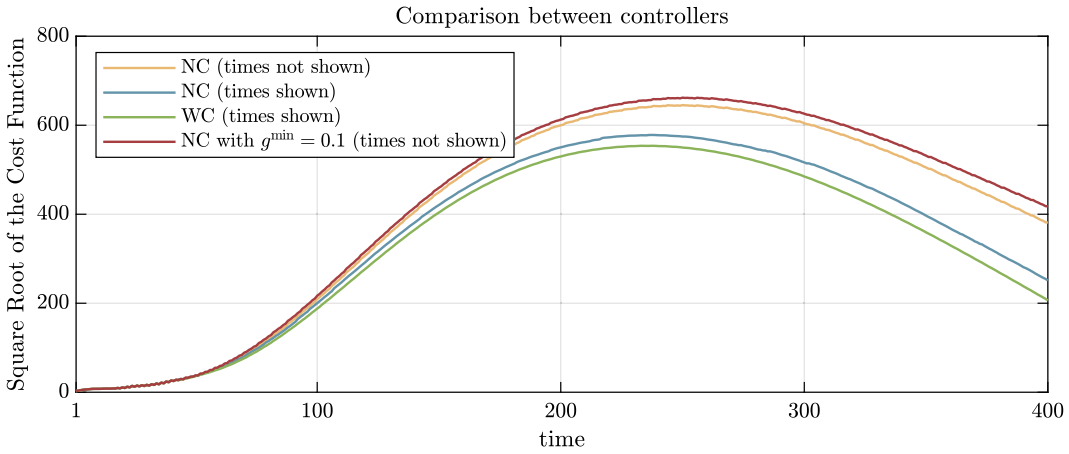


Fig. 5. Comparison between controllers subject to profile 1 in Fig. 2. In ordinate the square root of the cost function (28).

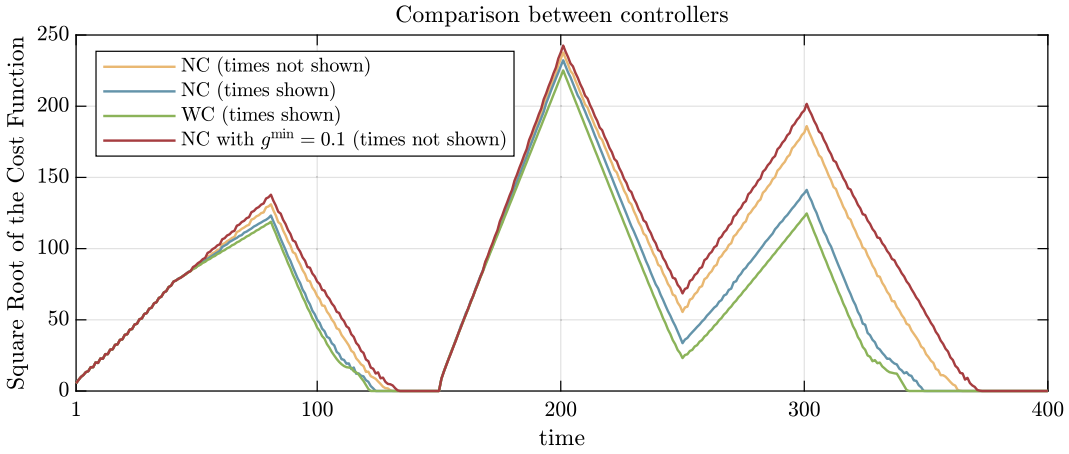


Fig. 6. Comparison between controllers subject to profile 2 in Fig. 2. In ordinate the square root of the cost function (28).

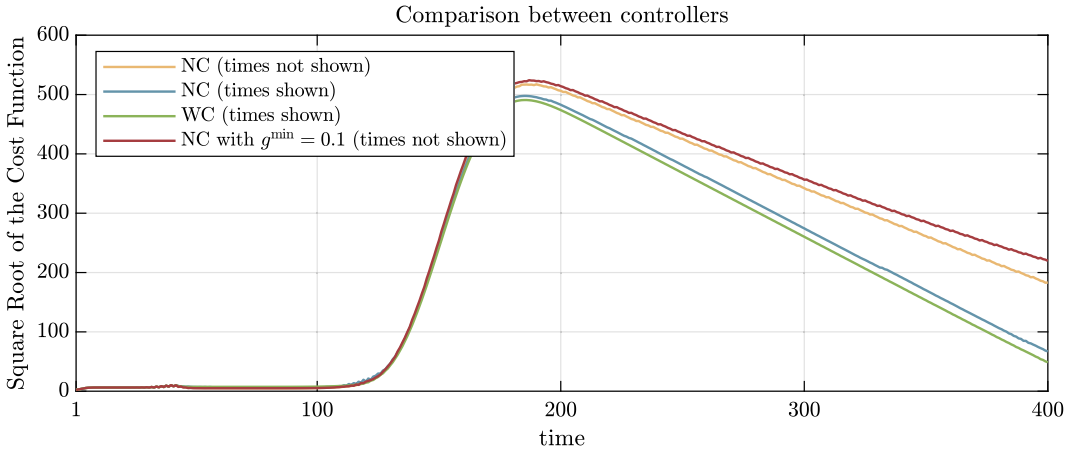


Fig. 7. Comparison between controllers subject to profile 3 in Fig. 2. In ordinate the square root of the cost function (28).

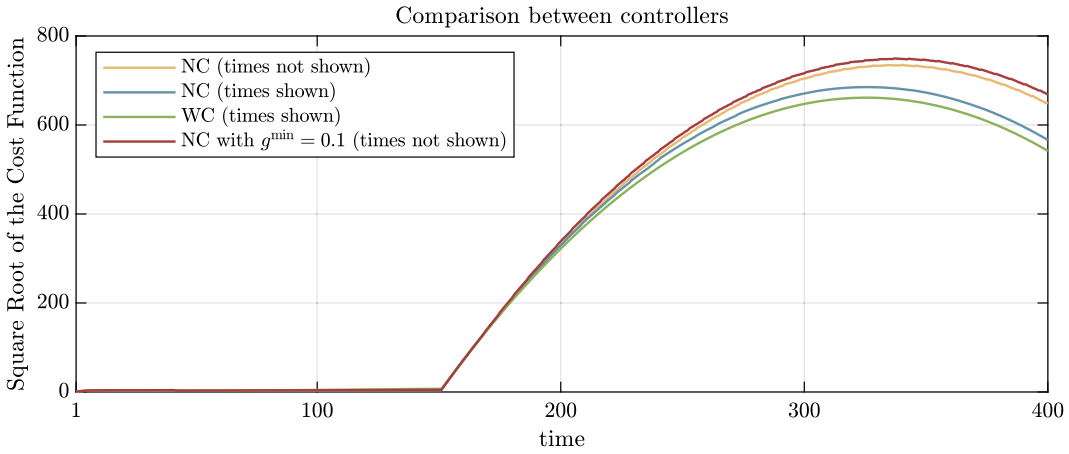


Fig. 8. Comparison between controllers subject to profile 4 in Fig. 2. In ordinate the square root of the cost function (28).

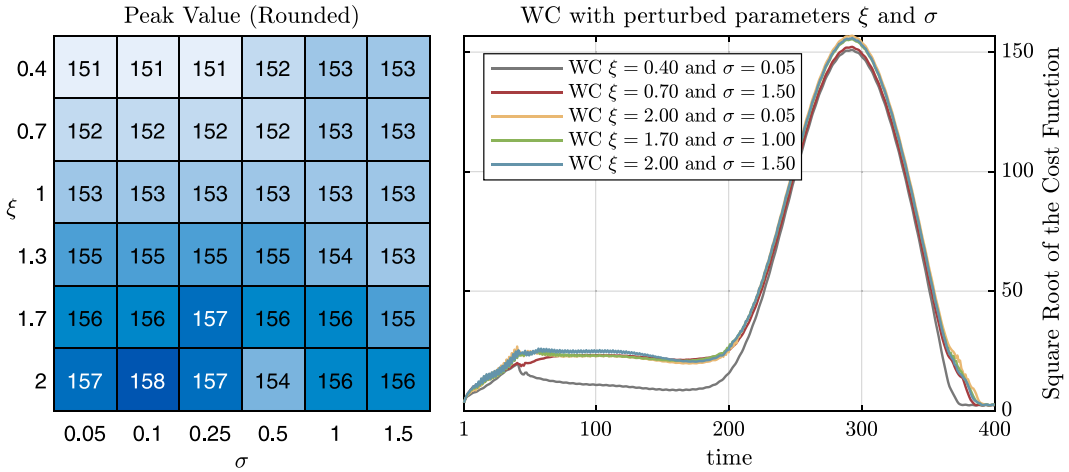


Fig. 9. **WC** controller with $g^{\min} = 10^{-2}$, $\hat{g}^t_{i,j,f} = g^t_{i,j,f}$ for all $(i,j,f) \in \mathcal{P}$, and $(\check{\xi}, \check{\sigma}, \check{\eta}) = (\xi, \sigma, \eta)$, for different values of ξ and σ. In all simulations, the input profile 0 (Eq. (57)) is used. (Left) Heatmap reporting the peak value of the square root of the cost function (28). (Right) Some sample trajectories. In ordinate the square root of the cost function (28).

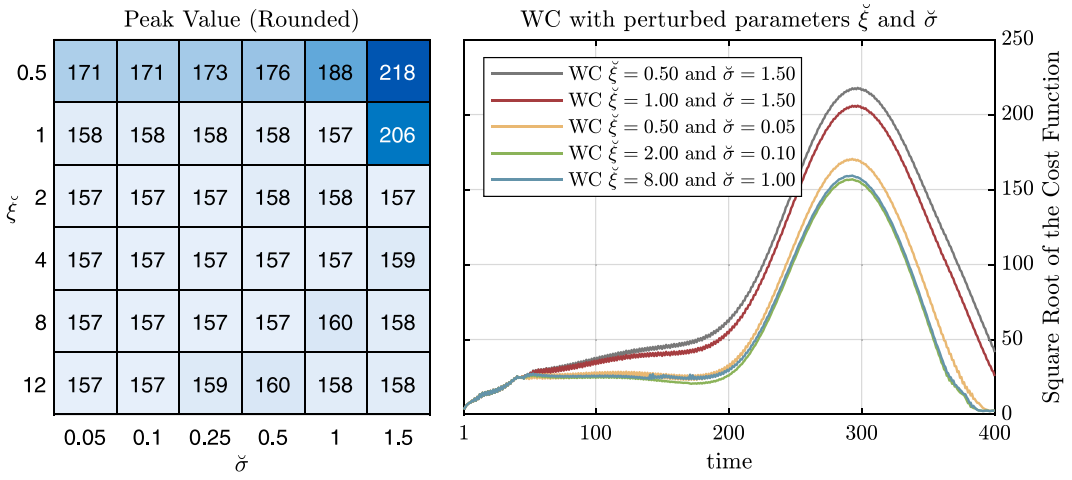


Fig. 10. **WC** controller with $g^{\min} = 10^{-2}$, $\hat{g}^t_{i,j,f} = g^t_{i,j,f}$ for all $(i,j,f) \in \mathcal{P}$, and different values of $\breve{\xi}$ and $\breve{\sigma}$. In all simulations, $(\xi, \sigma) = (4, 1/2)$ in (18), and the input profile 0 (Eq. (57)) is used. In ordinate the square root of the cost function (28).

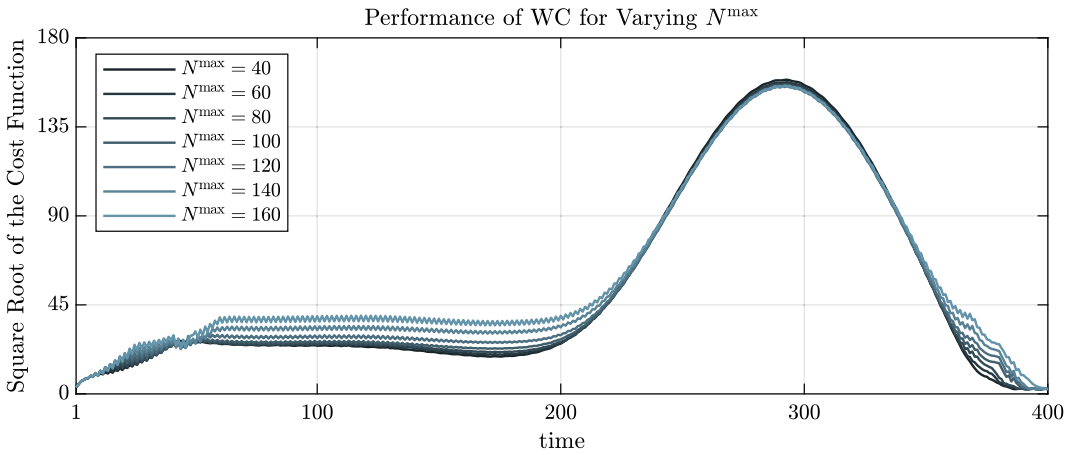


Fig. 11. **WC** controller with $g^{\min} = 10^{-2}$, $\hat{g}^t_{i,j,f} = g^t_{i,j,f}$ for all $(i,j,f) \in \mathcal{P}$, and different values of $N^{\max}$. In all simulations, the input profile 0 (Eq. (57)) is used. In ordinate the square root of the cost function (28).

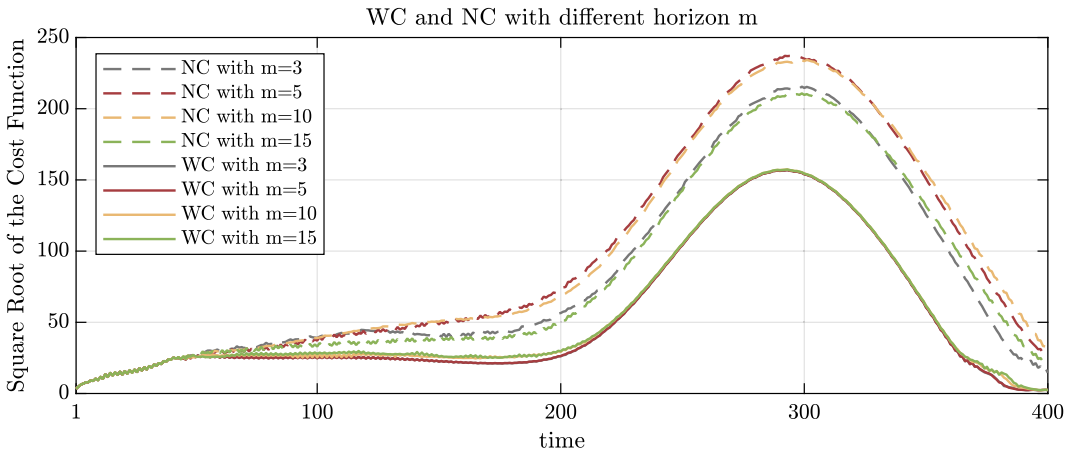


Fig. 12. **WC** and **NC** controllers with $g^{\min} = 10^{-2}$, $\hat{g}^t_{i,j,f} = g^t_{i,j,f}$ for all $(i,j,f) \in \mathcal{P}$, and different values of the control horizon m. In all simulations, the input profile 0 (Eq. (57)) is used. In ordinate the square root of the cost function (28).

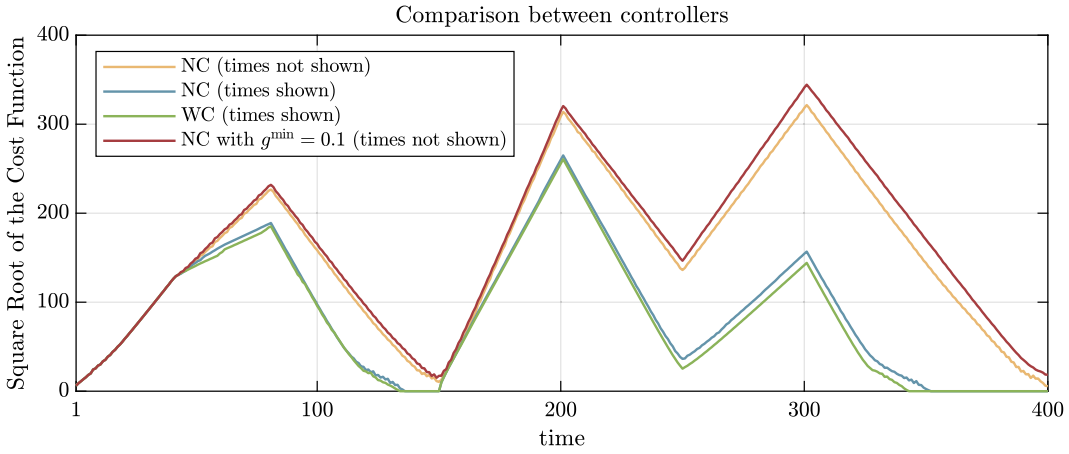


Fig. 13. Comparison between the same controller of Fig. 3 applied to the network of 1(b) in which $\zeta_{a,J}$, $\zeta_{g,J}$, and $\zeta_{i,q}(t) = 0$ for all other pairs (i, q). In ordinate the square root of the cost function (28). For all controllers, $\Delta_c = m = 3$ (Section 4.1).

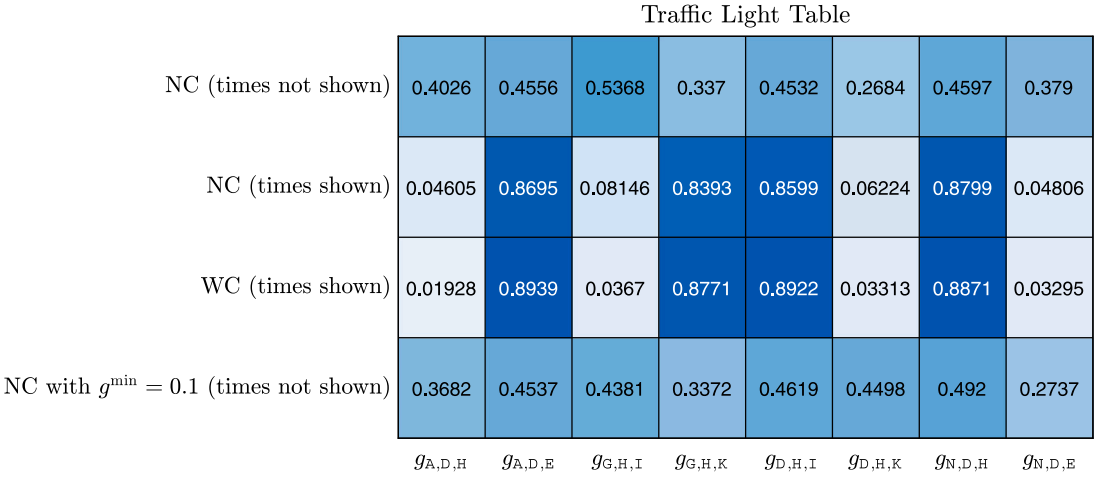


Fig. 14. Table showing the mean value in the interval $[150, 350]$ of the duty cycles for some sample lights obtained in the simulations of Fig. 13.

CRediT authorship contribution statement

Matteo Scandella: Literature review, Model definition, Control scheme development, Software implementation, Writing. **Arnob Ghosh:** Control scheme idea, Literature review, Editing. **Michelangelo Bin:** Conceptualization, Literature review, Editing, Simulations, Visualization. **Thomas Parisini:** Conceptualization, Editing, Supervision.

Appendix A. Proof of Lemma 1

Firstly, we can employ the constraint (14), in order to reduce $\mathscr{A}$ to

$$\mathscr{A}_2 : \begin{cases} \min \sum_{(i,j,f)\in\mathcal{P}} \sum_{q\in\mathcal{V}\setminus\{j\}} J'_a\left(\alpha_{i,j,f,q}\right), \\ \text{Subject to (12), and (13).} \end{cases}$$

where

$$J'_a\left(\alpha_{i,j,f,q}\right) = \mu\ \alpha_{i,j,f,q}\ \rho_{i,j,f,q} + \alpha_{i,j,f,q}\ \log \alpha_{i,j,f,q},$$

and $\alpha_{i,j,f,j} = 0$ for all $(i,j,f) \in \mathcal{P}$ as indicated in the constraint (14). Now, we can focus on the relaxed problem where the inequality constraints (12) are removed. Hence, we want to solve

$$\mathscr{A}_3 : \begin{cases} \min \displaystyle\sum_{(i,j,f)\in\mathcal{P}} \sum_{q\in\mathcal{V}\setminus\{j\}} J'_a\left(\alpha_{i,j,f,q}\right), \\ \text{Subject to (13).} \end{cases}$$

Defining $J'_a(0) = 0$, the cost function and the constraints (13) are continuously differentiable functions in the compact set $[0,1]^{n_p}$. For this reason and noticing that $\mathscr{A}_3$ presents only equality constraints, it is possible to solve $\mathscr{A}_3$ by using the method of Lagrange multipliers. Therefore, we define the Lagrangian function as

$$L(\alpha,\lambda) = \sum_{(i,j,f)\in\mathcal{P}} \sum_{q\in\mathcal{V}\setminus\{j\}} J'_a\left(\alpha_{i,j,f,q}\right) + \sum_{\forall(i,j)\in\mathcal{E}} \sum_{\forall q\in\mathcal{V}\setminus\{j\}} \lambda_{i,j,q}\left(\sum_{k\in\mathcal{O}_{i,j}} \alpha_{i,j,k,q} - 1\right)$$

where $\lambda = \left(\lambda_{i,j,q}\right)_{(i,j)\in\mathcal{E},q\in\mathcal{V}\setminus\{j\}}$ are the Lagrange multipliers. Then, the minimizer can be found by solving the system of equations

$$\begin{cases} \dfrac{\partial}{\partial\alpha_{i,j,f,q}} L(\alpha,\lambda) = 0, & \forall (i,j,f) \in \mathcal{P}, \forall q \in \mathcal{V}\setminus\{j\}, \\ \dfrac{\partial}{\partial\lambda_{i,j,q}} L(\alpha,\lambda) = 0, & \forall (i,j) \in \mathcal{E}, \forall q \in \mathcal{V}\setminus\{j\}. \end{cases}$$

With some mathematical steps, we obtain

$$\begin{cases} 1 + \log\alpha_{i,j,f,q} + \mu\rho_{i,j,f,q} + \lambda_{i,j,q} = 0 & \forall (i,j,f) \in \mathcal{P}, \forall q \in \mathcal{V}\setminus\{j\}, \\ \displaystyle\sum_{k\in\mathcal{O}_{i,j}} \alpha_{i,j,k,q} - 1 = 0, & \forall (i,j) \in \mathcal{E}, \forall q \in \mathcal{V}\setminus\{j\}, \end{cases}$$

whose solution is given by

$$\alpha_{i,j,f,q} = \frac{\exp\left(-\mu\rho_{i,j,f,q}\right)}{\displaystyle\sum_{k\in\mathcal{O}_{i,j}} \exp\left(-\mu\rho_{i,j,k,q}\right)}.$$

Therefore, the solution of $\mathscr{A}_3$ satisfies the constraint (12). Hence, the solution of $\mathscr{A}_3$ is also the solution of $\mathscr{A}_2$. For this reason, the solution of $\mathscr{A}$ is given by (16). ■

Appendix B. Proof of Lemmas 2 and 3

The optimization problems $\mathscr{B}_1(t)$ and $\check{\mathscr{B}}_1(t)$ are very similar to the optimization problem $\mathscr{A}_2$ defined in Appendix A and their solution is obtained using the same process explained in Appendix A. ■

References

A, P.L., Bhatnagar, S., 2011. Reinforcement learning with function approximation for traffic signal control. IEEE Trans. Intell. Transp. Syst. 12 (2), 412–421.

ACEA, 2019. Acea report – vehicles in use, europe 2019. URL https://www.acea.be/publications/article/report-vehicles-in-use-europe-2019 Accessed: October 17th, 2022 [online].

Agarwal, R.P., Meehan, M., O'Regan, D., 2001. Fixed Point Theory and Applications. In: Cambridge Tracts in Mathematics, Cambridge University Press.

Al Islam, S.B., Hajbabaie, A., Aziz, H.A., 2020. A real-time network-level traffic signal control methodology with partial connected vehicle information. Transp. Res. C 121, 102830.

Andersen, E.D., Roos, C., Terlaky, T., 2003. On implementing a primal-dual interior-point method for conic quadratic optimization. Math. Program. 95 (2), 249–277.

Araghi, S., Khosravi, A., Creighton, D., 2015. A review on computational intelligence methods for controlling traffic signal timing. Expert Syst. Appl. 42 (3), 1538–1550.

Arel, I., Liu, C., Urbanik, T., Kohls, A.G., 2010. Reinforcement learning-based multi-agent system for network traffic signal control. IET Intell. Transp. Syst. 4 (2), 128–135.

Bani Younes, M., Boukerche, A., 2014. An intelligent traffic light scheduling algorithm through VANETs. In: 39th Annual IEEE Conference on Local Computer Networks Workshops. pp. 637–642.

Beckmann, M.J., McGuire, C.B., Winsten, C.B., 1955. Studies in the Economics of Transportation. RAND Corporation, Santa Monica, CA.

Bertsimas, D., Tsitsiklis, J.N., 1997. Introduction To Linear Optimization, Vol. 6. Athena Scientific Belmont, MA, p. 608.

Bomze, I.M., 1998. On standard quadratic optimization problems. J. Global Optim. 13 (4), 369–387.

Boyd, S., Vandenberghe, L., 2019. Convex Optimization. Cambridge University Press.

Chai, H., Zhang, H., Ghosal, D., Chuah, C.-N., 2017. Dynamic traffic routing in a network with adaptive signal control. Transp. Res. C 85, 64–85.

Chandrasekaran, V., Shah, P., 2016. Relative entropy optimization and its applications. Math. Program. 161 (1–2), 1–32.

Chang, T.-H., Sun, G.-Y., 2004. Modeling and optimization of an oversaturated signalized network. Transp. Res. B 38 (8), 687–707.

Chen, T., Ardeshiri, T., Carli, F.P., Chiuso, A., Ljung, L., Pillonetto, G., 2016. Maximum entropy properties of discrete-time first-order stable spline kernel. Automatica 66, 34–38.

Chiou, S.-W., 2018. A robust signal control system for equilibrium flow under uncertain travel demand and traffic delay. Automatica 96, 240–252.

Chow, A.H., Sha, R., Li, S., 2020. Centralised and decentralised signal timing optimisation approaches for network traffic control. Transp. Res. C 113, 108–123, ISTTT 23 TR_C-23rd International Symposium on Transportation and Traffic Theory (ISTTT 23).
Chu, T., Wang, J., 2017. Traffic signal control by distributed Reinforcement Learning with min-sum communication. In: 2017 American Control Conference (ACC). pp. 5095–5100.
Coleman, T.F., Li, Y., 1996. A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables. SIAM J. Optim. 6 (4), 1040–1058.
Daganzo, C.F., 1995. A pareto optimum congestion reduction scheme. Transp. Res. B 29 (2), 139–154.
Dai, J.G., Lin, W., 2005. Maximum pressure policies in stochastic processing networks. Oper. Res. 53 (2), 197–218.
De Palma, A., Lindsey, R., 2004. Congestion pricing with heterogeneous travelers: A general-equilibrium welfare analysis. Netw. Spat. Econ. 4 (2), 135–160.
Edenhofer, O., Pichs-Madruga, R., Sokona, Y., Farahani, E., Kadner, S., Seyboth, K., Adler, A., Baum, I., Brunner, S., Eickemeier, P., Kriemann, B., Savolainen, J., Schlömer, S., von Stechow, C., Zwickel, T., Minx, J. (Eds.), 2014. IPCC, 2014: Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA.
Eom, M., Kim, B.-I., 2020. The traffic signal control problem for intersections: a review. Eur. Transp. Res. Rev. 12 (1), 50.
Fleck, J.L., Cassandras, C.G., Geng, Y., 2016. Adaptive quasi-dynamic traffic light control. IEEE Trans. Control Syst. Technol. 24 (3), 830–842.
Fletcher, R., 1971. A general quadratic programming algorithm. IMA J. Appl. Math. 7 (1), 76–91.
Friesz, T.L., Mookherjee, R., Yao, T., 2008. Securitizing congestion: The congestion call option. Transp. Res. B 42 (5), 407–437.
García, C.E., Prett, D.M., Morari, M., 1989. Model predictive control: Theory and practice - A survey. Automatica 25 (3), 335–348.
Ghosh, A., Scandella, M., Bin, M., Parisini, T., 2021. Traffic-light control at urban intersections using expected waiting-time information. In: IEEE 60th Conf. Decis. Control. pp. 1953–1959.
Gregoire, J., Qian, X., Frazzoli, E., de La Fortelle, A., Wongpiromsarn, T., 2015. Capacity-aware backpressure traffic signal control. IEEE Trans. Control Netw. Syst. 2 (2), 164–173.
Kockelman, K.M., Kalmanje, S., 2005. Credit-based congestion pricing: a policy proposal and the public's response. Transp. Res. Part A 39 (7–9), 671–690.
Kouvaritakis, B., Cannon, M., 2016. Model Predictive Control. Springer International Publishing.
Le, T., Kovács, P., Walton, N., Vu, H.L., Andrew, L.L., Hoogendoorn, S.S., 2015. Decentralized signal control for urban road networks. Transp. Res. C 58, 431–450.
Lessan, J., Fu, L., 2019. Credit- and permit-based travel demand management state-of-the-art methodological advances. Transp. A 1–24.
Liu, D., Yu, W., Baldi, S., Cao, J., Huang, W., 2020. A switching-based adaptivecybernetics dynamic programming method to optimal traffic signaling. IEEE Trans. Syst., Man, Cybern.: Syst. 50 (11), 4160–4170.
Löfberg, J., 2004. YALMIP: A toolbox for modeling and optimization in MATLAB. In: Proceedings of the CACSD Conference, Vol. 3. Taipei, Taiwan, pp. 284–289.
McShane, C., 1999. The origins and globalization of traffic control signals. J. Urban Hist. 25, 379–404.
Mertikopoulos, P., Papadimitriou, C., Piliouras, G., 2018. Cycles in adversarial regularized learning. In: Proceedings of the 2018 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). SIAM, pp. 2703–2717.
Miller, A.J., 1963. Settings for fixed-cycle traffic signals. J. Oper. Res. Soc. 14 (4), 373–386.
Mosek APS, 2022. The MOSEK modeling cookbook. Freely Available At https://docs.mosek.com/modeling-cookbook/index.html.
Neu, G., Jonsson, A., Gómez, V., 2017. A unified view of entropy-regularized Markov decision processes. CoRR abs/1705.07798.
Nilsson, G., Como, G., 2020. A micro-simulation study of the generalized proportional allocation traffic signal control. IEEE Trans. Intell. Transp. Syst. 21 (4), 1705–1715.
Papageorgiou, M., Kiakaki, C., Dinopoulou, V., Kotsialos, A., Yibing Wang, 2003. Review of road traffic control strategies. Proc. IEEE 91 (12), 2043–2067.
Pigou, A.C., 1920. The Economics of Welfare. Macmillan and Co., London.
Small, K.A., Verhoef, E.T., 2007. The Economics of Urban Transportation, second ed. Routledge, London.
Spall, J.C., Chin, D.C., 1997. Traffic-responsive signal timing for system-wide traffic control. Transp. Res. C 5 (3), 153–163.
Sprung, M.J., Chambers, M., Smith-Pickel, S., 2018. Transportation statistics annual report 2018. http://dx.doi.org/10.21949/1502596, Accessed: October 17th, 2022 [online].
Srinivasan, D., Choy, M.C., Cheu, R.L., 2006. Neural networks for real-time traffic signal control. IEEE Trans. Intell. Transp. Syst. 7 (3), 261–272.
Tassiulas, L., Ephremides, A., 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Trans. Automat. Control 37 (12), 1936–1948.
Tettamanti, T., Luspay, T., Kulcsar, B., Peni, T., Varga, I., 2014. Robust control for urban road traffic networks. IEEE Trans. Intell. Transp. Syst. 15 (1), 385–398.
UK Department for Transport, 2019. Road traffic estimates in great britain: 2019. URL https://www.gov.uk/government/statistics/road-traffic-estimates-in-great-britain-2019 Accessed: October 17th, 2022 [online].
Varaiya, P., 2013. Max pressure control of a network of signalized intersections. Transp. Res. C 36, 177–195.
Verhoef, E.T., 2002. Second-best congestion pricing in general networks. Heuristic algorithms for finding second-best optimal toll levels and toll points. Transp. Res. B 36 (8), 707–729.
Verhoef, E., Nijkamp, P., Rietveld, P., 1997. Tradeable permits: Their potential in the regulation of road transport externalities. Environ. Plan. B: Plann. Des. 24 (4), 527–548.
Viegas, J.M., 2001. Making urban road pricing acceptable and effective: searching for quality and equity in urban mobility. Transp. Policy 8 (4), 289–294.
Walters, A.A., 1961. The theory and measurement of private and social cost of highway congestion. Econometrica 29 (4), 676–699.
Wang, X., Yang, H., Han, D., 2010. Traffic rationing and short-term and long-term equilibrium. Transp. Res. Rec. 2196 (1), 131–141.
Wu, N., Li, D., Xi, Y., 2019. Distributed weighted balanced control of traffic signals for urban traffic congestion. IEEE Trans. Intell. Transp. Syst. 20 (10), 3710–3720.
Xu, H.K., 2003. An iterative approach to quadratic optimization. J. Optim. Theory Appl. 116 (3), 659–678.
Yang, H., Huang, H., 2005. Mathematical and Economic Theory of Road Pricing. Elsevier, Amsterdam; Boston.
Yang, H., Wang, X., 2011. Managing network mobility with tradable credits. Transp. Res. B 45 (3), 580–594.
Yau, K.-L.A., Qadir, J., Khoo, H.L., Ling, M.H., Komisarczuk, P., 2017. A survey on reinforcement learning models and algorithms for traffic signal control. ACM Comput. Surv. 50 (3), 1–38.
Ye, Y., Tse, E., 1989. An extension of Karmarkar's projective algorithm for convex quadratic programming. Math. Program. 44 (1–3), 157–179.
Younis, O., Moayeri, N., 2017. Employing cyber-physical systems: Dynamic traffic light control at road intersections. IEEE Internet Things J. 4 (6), 2286–2296.
Zhao, L., Peng, X., Li, L., Li, Z., 2011. A fast signal timing algorithm for individual oversaturated intersections. IEEE Trans. Intell. Transp. Syst. 12 (1), 280–283.