Network Traffic Signal Control via Non-convex ADMM Formulations

Stelios Timotheou
Research Associate*
Phone: +357-22-893459
timotheou.stelios@ucy.ac.cy

Christos G. Panayiotou
Associate Professor*
Phone: +357-22-892298
christosp@ucy.ac.cy

Marios M. Polycarpou
Professor and Director*
Phone: +357-22-892252
mpolycar@ucy.ac.cy

*KIOS Research Center for Intelligent Systems and Networks
University of Cyprus
P.O.Box 20537
CY-1678 Nicosia
Cyprus
Fax: +357-22-893455

A Paper Submitted for Presentation at the 2015 Annual Meeting of the Transportation Research Board and Publication in the Transportation Research Record

Total words = 5990 + 250*4 (4 Figures + 0 Table) + 500 (references)= 7490
November 15, 2014
This paper considers the distributed solution of the online network traffic signal control problem. Towards system-wide optimality, the problem is modeled as a large-scale mixed-integer linear program (MILP) with the traffic dynamics captured by the Cell Transmission Model (CTM). The alternating direction method of multipliers (ADMM) is utilized to achieve spatial problem decomposition and design an iterative approach that achieves network-wide solution under a fully distributed architecture where computation, communication and control is performed locally at individual intersections. Two different ADMM-based algorithms are developed based on appropriate problem reformulations, resulting in the solution of convex/non-convex subproblems with distinct properties. Their performance is demonstrated to be close to global optimality and comparative to genetic algorithms, while each algorithm offers a different tradeoff between communication and computation complexity.
1. INTRODUCTION

The ever increasing urbanization and motorization of cities requires the implementation of efficient strategies to alleviate congestion. Given that major road constructions in cities are both difficult and costly, Intelligent Transportation System (ITS) strategies are considered invaluable to improve the capacity of existing infrastructure. Traffic signal control is among the most prominent ITS strategies to reduce traffic congestion, and improve the conditions both for the drivers and the environment. Nevertheless, network/system-wide traffic signal control (NTSC) is a challenging task due to the large-scale nature and complexity of the problem, the uncertain and dynamic behavior of the network (e.g. weather, accidents) and the unpredictable driver behavior. For this reason, several solution techniques have been proposed with different characteristics in terms of adaptability, system architecture, scalability, computational complexity and traffic model consideration.

Fixed or pre-timed signal control strategies optimize offline the split plans, based on historical data so that a fixed signal timing plan is in place for different periods of the day, such as the Traffic Network Study Tool (TRANSYT) (1). These approaches usually rely on the development of mixed-integer linear programming (MILP) formulations using the Cell Transmission Model (CTM) (2, 3) or other traffic models. However, because the formulated problems are usually NP-hard, meta-heuristic techniques such as genetic algorithms (4) or greedy heuristics (5) are often employed to achieve close to optimal and timely solutions. Pre-timed signal control strategies can perform fairly well during peak traffic periods, but their performance deteriorates during off-peak periods or when unexpected events create different traffic conditions than anticipated.

To account for stochastic variations of traffic flows, several online adaptive traffic signal control (ATSC) systems have been developed based on real-time traffic information. Such strategies usually rely on centralized or hierarchical system architectures to adaptively optimize traffic signal plan parameters such as splits, offsets and cycle, (e.g. the Split Cycle and Offset Optimization Technique (SCOOT) system, (6), and the Real-Time Hierarchical Optimized Distributed Effective System (RHODES) (7)) or select the best from a library of pre-calculated signal plans (e.g. the Sydney Coordinated Adaptive Traffic System (SCATS) (8)).

To avoid temporally myopic decision making that relies only on current information, model predictive control (MPC) approaches consider predicted values of future traffic such as in (9) and (10). Although these approaches can address both non-saturated and saturated traffic conditions, to deal with the large-scale nature of the problem in both space and time, they usually rely on a coarse representation of the transportation infrastructure and continuous instead of binary green traffic light allocations. Another noteworthy and practicable attempt to address the problem of network-wide saturated traffic conditions is the traffic-responsive urban control (TUC) strategy that is based on store-and-forward modeling and linear-quadratic regulator theory (11). However, both MPC and store-and-forward approaches rely on centralized architectures.

Recently, it has been theoretically suggested (12) and empirically observed (13) that the average flow-density relationship of homogeneous networks with spatially uniform demands and congestion patterns can be described by a Network Fundamental Diagram (NFD). Based on this observation several strategies have been proposed to control the amount of traffic that resides in regions described by different NFDs via gating measures. Large-scale urban gating strategies have been applied either to protect a central area (14, 15), or multiple regions (16, 17). The later requires apart from gating control, boundary control to regulate traffic exchanged between regions. The NFD concept has also been exploited to control mixed-networks under an MPC framework (18). In (19), a reduced NFD was used for gating based on real-time measurements, to avoid the
extensive amount of measurements needed for the characterization of regions with distinct NFDs. Nonetheless, this class of algorithms also relies on centralized architectures.

Distributed online algorithms dynamically adjust the split plans of multiple intersections based on the network state through local decisions and communication between intersection controllers. Such algorithms usually draw inspiration from artificial and computational intelligence techniques such as reinforcement learning (20) and computational markets (21) or more recently from networked control (22). Due to the complex and large-scale nature of the problem these algorithms often rely on low-complexity but suboptimal solution approaches, and often consider simplistic traffic decisions that do not account for the signal phase sequence or the offset between intersections.

In this paper, two cooperative distributed online algorithms for the NTSC problem using CTM traffic dynamics are proposed. Distributed cooperative optimization is achieved by iteratively solving non-convex one-intersection subproblems via an iterative method that solves optimization problems by breaking them into smaller pieces called alternating direction method of multipliers (ADMM) (23) and exchanging information between adjacent intersection controllers. Online optimization is accomplished by applying traffic signal plans for a certain small time-period (e.g. 10 mins), and proactively computing traffic plans for the next period. In this way, our approach departs from the traditional view of considering optimization techniques for the solution of CTM-based NTSC problems, only for offline centralized systems, and also offers scalability to large networks. The two solution algorithms are comprised of two phases. In the first phase, an initial solution to the problem is obtained via appropriate non-convex ADMM formulations; the resulting solution is improved in the second phase through an appropriate distributed heuristic. Although both algorithms rely on ADMM, they have distinct properties in terms of solution quality, communication and computation. This is a result of the different subproblems that need to be solved; MIQP-ADMM convergences fast but each iteration requires the solution of Mixed-Integer Quadratic Programs (MIQP) which are computational intensive, while SP-ADMM convergences slower but each iteration require the solution of quadratic programming and shortest path (SP) subproblems that are significantly faster to solve.

The remainder of this paper is organized as follows. Section 2 outlines CTM and ADMM to facilitate the reader’s understanding. Section 3 explains temporal decomposition and presents the formulation of the NTSC problem. In Section 4 spatial decomposition is explained, while in Section 5 the two distributed ADMM algorithms are developed, along with a solution improving heuristic. Section 6, discusses the simulation results for the two proposed ADMM-based algorithms, while the final section summarizes the paper and discusses directions for future work.

**Notation:** All boldface letters indicate vectors (lower case) or matrices (upper case), while calligraphic letters denote sets. Superscript $(\cdot)^T$ denotes the matrix transpose. $||z||_2$ denotes the Euclidean norm of a vector $z$. Operators $A \cup B$, $A \cap B$, and $A \setminus B$, denote the union, intersection and set difference of sets $A$ and $B$, respectively, while $|A|$ denotes the cardinality of set $A$. $|x|$ denotes the absolute value of variable $x$. $h_C(x)$ is the indicator function which is equal to zero if $x \in C$ and $+\infty$ otherwise.

**2. PRELIMINARIES**

**2.1 Cell Transmission Model**

CTM (24) is a discrete analog in both space and time of the well-known first-order Lighthill-Whitham-Richards (LWR) continuum flow model. In CTM, each road segment is divided into
homogeneous sections called cells, while time is partitioned so that one vehicle takes one time-unit to travel through one cell at free-flow speed. Assuming a piecewise-linear trapezoidal flow-density fundamental diagram, CTM equations for ordinary cell $c \in \mathcal{E}$ are given by:

$$y_{c,t} = \min(n_{c,t}, Q_{c,t}, Q_{c+1,t}, W_{c+1,t}(N_{c+1,t} - n_{c+1,t}))$$ (1)

$$n_{c,t+1} = n_{c,t} + y_{c-1,t} - y_{c,t}$$ (2)

In the above equations, $y_{c,t}$ represents the number of vehicles that leave cell $c$ and enter $c+1$ at time-interval $[t, t+1)$, while $n_{c,t}$ is the number of vehicles inside $c$ at the beginning of time-step $t$. Important cell parameters are the maximum number of vehicles that can flow through ($Q_{c,t}$) or reside into cell $c$ ($N_{c,t}$), and the ratio between the shock-wave propagation and free-flow speeds ($W_{c,t}$) at time-interval $[t, t+1)$. In homogeneous networks, it is true that quantities $Q_{c,t} = Q$, $N_{c,t} = N$ and $W_{c,t} = W$ are constant for all cells. Eq. (1) indicates that the number of vehicles leaving cell $c$ is limited either by the number of vehicles in the cell, the flow capacity at the boundary between cells $c$ and $c+1$ and the space left in the successor cell when a queue is forming. Eq. (2) ensures flow conservation at cell $c$.

Apart from ordinary cells, other cell types are necessary to model the traffic dynamics. In origin cells $c \in \mathcal{O}$, a number of vehicles, $D_{c,t}$, enters from the outside of the network rather than from predecessor cells ($D_{c,t}$ replaces $y_{c-1,t}$ in eq. (2)). Destination cells $c \in \mathcal{D}$, have no successor cells and eq. (1) becomes $y_{c+1,t} = n_{c,t}$. Finally, intersection cells $c \in \mathcal{I}$ have variable capacity $q_{c,t}$ equal to $Q_{c,t}$ if the vehicles in the cell have the right-of-way and 0 otherwise.

### 2.2 Alternating Direction Method of Multipliers

ADMM is a powerful method for solving mathematical optimization problems of the form

$$\min_{x} \ f(x) + g(z)$$ (3a)

subject to $Ax + Bz = d$ (3b)

$$x \in \mathcal{C}_x, \ z \in \mathcal{C}_z$$ (3c)

where $x \in \mathcal{C}_x \subseteq \mathbb{R}^{M_x \times 1}$, $z \in \mathcal{C}_z \subseteq \mathbb{R}^{M_z \times 1}$, $A \in \mathbb{R}^{M_d \times M_x}$, $B \in \mathbb{R}^{M_d \times M_z}$ and $d \in \mathbb{R}^{M_d \times 1}$, $f(x)$ and $g(z)$ are convex functions and $\mathcal{C}_x, \mathcal{C}_z$ are closed convex sets. ADMM is capable of providing fast and often distributed solutions to convex problems through appropriate decomposition of the considered problem into simpler ones (23). For the solution of problem (3), ADMM uses the scaled augmented Lagrangian form, $\mathcal{L}_\rho(x, z, u)$:

$$\mathcal{L}_\rho(x, z, u) = f(x) + g(z) + h_{\mathcal{C}_x}(x) + h_{\mathcal{C}_z}(z) + (\rho/2)\|Ax + Bz - d + u\|^2_2$$

where $u = \omega/\rho$ are the scaled dual variables, $\omega \in \mathbb{R}^{M_d \times 1}$ are the dual variables or Lagrange multipliers and $\rho \in \mathbb{R}$ is a penalty constant, while $h_{\mathcal{C}_x}(x)$ and $h_{\mathcal{C}_z}(z)$ are indicator functions. Starting from initial values $\bar{x}^0$ and $\bar{u}^0$, ADMM iteratively minimizes $\mathcal{L}_\rho(x, z, u)$ with respect to $x$ and $z$ followed by an update of the scaled dual variables in three consecutive steps:
Step 1: \[ x^{k+1} = \arg\min_x \mathcal{L}_\rho(x, z^k, u^k) \]
Step 2: \[ z^{k+1} = \arg\min_z \mathcal{L}_\rho(x^{k+1}, z, u^k) \]
Step 3: \[ u^{k+1} = Ax^{k+1} + Bz^{k+1} - d + u^k \]

The procedure continues until a stopping criterion is satisfied. Contrary to other decomposition methods that impose strong convergence conditions, ADMM enjoys the superior convergence properties of the method of multipliers with mild technical convergence conditions. The same procedure can be utilized when \( C_x \) and \( C_z \) are non-convex sets. Nonetheless, in this case even when the resulting subproblems can accurately be solved, ADMM need not converge, and when it does it need not converge to the globally optimal solution. Employing ADMM for non-convex problems, is done with the hope that it will possibly give better performance than other algorithms in terms of convergence or optimality. 

3. CENTRALIZED PROBLEM FORMULATION

We consider the problem of jointly optimizing the traffic signal plans of a transportation network with multiple intersections over a time-horizon \( T_h \). Optimization is performed for the mean/total vehicle delay but other measures of interest such as stoppage time and throughput can also be employed. Traffic dynamics are incorporated into the optimization problem through CTM. To simplify the problem, we separate \( T_h \) into smaller time-windows, \( T_w \), and optimize the problem sequentially over those periods. The optimization period considered for each subproblem \( T_w \), is larger than \( T_w \) to avoid the effects of short-term planning, while generated traffic is considered for \( T_w \) time units. In fact, when online decision making is sought, traffic signal plans must be updated every \( T_w \) time units, using only the solution corresponding to the first \( T_w \) time units.

We consider a grid transportation network with two way streets and \( N_I \) intersections, as shown in Fig. 1. To model the traffic dynamics four types of CTM cell sets are utilized: ordinary (\( E \)), origin (\( O \)), destination (\( D \)) and intersection (\( I \)), with \( A = E \cup O \cup D \cup I \). We assume that traffic lights have two phases so that cells of intersection \( i \in R = \{1, \ldots, N_I\} \) receive a green light either at phase 1 or phase 2 (sets \( I_{1,i} \) and \( I_{2,i} \), respectively).

Assuming that decision variables \( w_{i,j} \), indicate whether phase 1 (\( w_{i,j} = 1 \)) or phase 2 (\( w_{i,j} = 0 \)) has the right-of-way at intersection \( i \in R \) at time \( t \in T = \{1, \ldots, T_o\} \), the NTSC problem can be formulated as:
\( \mathcal{NTSC} \): \( \text{min} \sum_{c \in \mathcal{A}} \sum_{t \in \mathcal{T}} n_{c,t} - \sum_{c \in \mathcal{A}} \sum_{t \in \mathcal{T}} y_{c,t} \)  
\text{s.t. } \begin{align*}
0 & \leq y_{c,t} \leq q_{c,t}, y_{c,t} \leq n_{c,t}, \quad c \in \mathcal{A}, t \in \mathcal{T}, \\
y_{c,t} & \leq W_{c+1,t} \bar{n}_{c+1,t}, \quad c \in \mathcal{A}, t \in \mathcal{T}, \\
n_{c,t+1} & = n_{c,t} + \bar{y}_{c,t} - y_{c,t}, \quad c \in \mathcal{A}, t \in \mathcal{T}, \\
q_{c,t} & = \begin{cases} 
Q_{c,t}, & c \in \mathcal{E} \cup \mathcal{O}, t \in \mathcal{T}, \\
(1-w_{i,t})Q_{c,t}, & i \in \mathcal{R}, c \in \mathcal{I}_{1,i}, t \in \mathcal{T}, \\
\infty, & c \in \mathcal{D}, t \in \mathcal{T}, 
\end{cases} \\
\bar{n}_{c+1,t} & = \begin{cases} 
N_{c+1,t} - n_{c+1,t}, & c \in \mathcal{A} \setminus \mathcal{D}, t \in \mathcal{T} \\
\infty, & \text{otherwise}
\end{cases} \\
\bar{y}_{c,t} & = \begin{cases} 
y_{c-1,t}, & c \in \mathcal{A} \setminus \mathcal{O}, t \in \mathcal{T} \\
D_{c,t}, & \text{otherwise}
\end{cases} \\
w_{i,t} - w_{i,t-1} & \leq u_{i,t} \leq w_{i,t} + w_{i,t-1}, \quad i \in \mathcal{R}, t \in \mathcal{T}, \\
-w_{i,t} + w_{i,t-1} & \leq u_{i,t} \leq 2 - w_{i,t} - w_{i,t-1}, \quad i \in \mathcal{R}, t \in \mathcal{T}, \\
\sum_{t=1}^{t+G_{\text{min}}} u_{i,t} & \leq 1, \quad i \in \mathcal{R}, t = 1, \ldots, T_0 - G_{\text{min}}, \\
\sum_{t=1}^{t+G_{\text{max}}-1} w_{i,t} & \leq G_{\text{max}}, i \in \mathcal{R}, t = 1, \ldots, T - G_{\text{max}} + 1, \\
\sum_{t=1}^{t+G_{\text{max}}-1} w_{i,t} & \geq 1, \quad i \in \mathcal{R}, t = 1, \ldots, T - G_{\text{max}} + 1, \\
w_{i,t} & = w_{i,t}^{\text{init}}, \quad i \in \mathcal{R}, t = -G_{\text{max}} + 1, \ldots, 0, \\
n_{c,1} & = n_{c,1}^{\text{init}}, \quad c \in \mathcal{A}, \\
w_{i,t} & \in \{0,1\}, 0 \leq u_{i,t} \leq 1, \quad i \in \mathcal{R}, t \in \mathcal{T}.
\end{align*} \tag{4a-4o} 

In the above formulation, (4a) minimizes the total delay, as \( n_{c,t} - y_{c,t} \) denotes the delay experienced by vehicles in cell \( c \) at time \( t \) in time units, since each vehicle that is inside cell \( c \) at the beginning of time-step \( t \) and does not exit during period \( [t, t+1] \) experiences one-unit of delay. Note that minimizing the total delay is equivalent to minimizing the total travel time (TTT), because in CTM vehicles either travel at free-flow speed or are motionless due to congestion.

Constraints (4b)-(4d), represent the fundamental CTM dynamics, while (4e)-(4g), define cell parameters \( q_{c,t}, \bar{n}_{c,t} \) and \( \bar{y}_{c,t} \), respectively for different types of cells. Inequalities (4h)-(4i) are equivalent to \( u_{i,t} = |w_{i,t} - w_{i,t-1}| \), when \( w_{i,t} \in \{0,1\} \); hence \( u_{i,t} \) indicates whether a signal change has occurred between \( t - 1 \) and \( t \) (green-to-red or red-to-green). Inequalities (4j)-(4l) ensure that the minimum and maximum duration of each phase is \( G_{\text{min}} \) and \( G_{\text{max}} \), respectively, while (4m)-(4n) are initial conditions that need to be respected regarding the state of the network and the traffic signals. Without loss of generality, we will assume for the rest of the paper that the network is homogeneous. More details regarding the centralized formulation can be found in (25, 26).
4. SPATIAL DECOMPOSITION

The distributed solution of the NTSC problem is motivated by the special structure of transportation networks that allow appropriate spatial decomposition, as explained through the example topology in Fig. 1, which depicts the CTM of a 2x2 grid topology with four intersections and two-way arterials. In the figure, the dashed lines indicate a possible spatial decomposition of the network which partitions the topology into four areas $A_i$, $i = 1, \ldots, 4$ so that each area is associated to exactly one intersection. Note that under such a decomposition, interactions between adjacent intersections only take place through boundary cells, while if the optimal incoming boundary flows are known for each intersection, the optimal solution to the global problem can be obtained by independently solving single-intersection subproblems.

This can be understood by observing the characteristic CTM equations (4b) - (4g), from which it is clear that the computation of variables ($y_{c,t}, n_{c,t+1}$) for ordinary cells only depends on variables associated with cells $c - 1$, $c$ and $c + 1$. This important observation, implies that the cell states can evolve independently for each area, apart from boundary cells which require information about variables that belong to predecessor or successor cells outside the particular area.

This discussion gives rise to two different types of boundary cells: (a) input and (b) output. An input boundary cell $c \in B^I_{i}$ is any boundary cell of area $A_i$ that receives inflow traffic from a cell of a neighboring area (e.g. cell 705). Similarly, an output boundary cell $c \in B^O_{i}$ is any boundary cell of area $A_i$ that sends outflow traffic to a cell of a neighboring area (e.g. cell 105). We also define the extended input and output boundary cell sets of area $A_i$, $B^IE_{i}$ and $B^OE_{i}$ respectively, which include all input/output cells associated with area $A_i$ and input/output cells that are directly adjacent to $A_i$, as well as the set of all boundary cells of area $A_i$, $B_i = B^I_{i} \cup B^O_{i}$ and the corresponding extended set $B^E_{i} = B^IE_{i} \cup B^OE_{i}$. For instance, for area $A_1$ we have that $B^I_{1} = \{705, 506\}$, $B^O_{1} = \{105, 305\}$, $B_{1} = \{705, 506, 105, 305\}$, $B^IE_{1} = \{705, 306, 506, 106\}$, $B^OE_{1} = \{105, 305, 505, 704\}$ and $B^E_{1} = \{705, 306, 506, 106, 105, 305, 505, 704\}$.

These definitions facilitate the distributed solution of the NTSC problem discussed next. Note that similar decomposition could be useful for solving other transportation problems using CTM such as the freeway ramp metering and the dynamic traffic assignment.

5. $\mathcal{NTSC}$ DISTRIBUTED FORMULATION AND SOLUTION

In this paper we consider the solution of the $\mathcal{NTSC}$ problem assuming a fully distributed architecture with each signalized intersection equipped with an intelligent controller that has sensing, communication, computation and control capabilities. Attaining a fully distributed solution approach has several advantages compared to centralized solutions in terms of scalability, robustness, and cost, as distributed systems are cheaper and easier to deploy, less susceptible to communication and hardware failures and scale better with problem size compared to centralized ones.

Both developed algorithms adhere to the following methodological stages:

1. Reformulation of NTSC problem to allow distributed solution via ADMM (see eqs. (6) and (12))

2. Definition and solution of resulting ADMM subproblems (algorithm 1)

3. Distributed improvement of obtained ADMM solution (algorithm 2)

Based on these stages, two ADMM-based algorithms are introduced for the distributed solution of $\mathcal{NTSC}$. Both algorithms are iterative and exploit the concept of boundary cells and the
structure of ADMM to decompose NTSC into easier to solve single-intersection subproblems and information exchange between neighboring intersections in each iteration. Nonetheless, the two algorithms differ in the subproblem structure considered in each iteration resulting in different requirements in terms of computation and communication. The first, MIQP-ADMM, reformulates the problem into one MIQP and two analytically solvable subproblems for each iteration of ADMM. The second, SP-ADMM, reformulates the problem into a constrained quadratic program, an MIQP which can be solved via a SP reformulation, and an analytically solvable subproblem. Because both MIQP-ADMM and SP-ADMM attempt to solve non-convex subproblems, convergence of ADMM cannot be theoretically guaranteed for neither case. Nonetheless, as discussed in Section 6, the empirical convergence observed in simulation results is satisfactory for both algorithms.

5.1 MIQP-ADMM Solution Algorithm

To derive a distributed formulation for \( NTSC \), we need to convert the problem into an iterative procedure composed of single-intersection subproblems, so that each intersection controller (IC) only solves problems involving local variables. Towards this direction, we rewrite the centralized \( NTSC \) problem into the following form that distinguishes decoupled constraints for each intersection (Eqs. (5b), (5c), (5d) and (5e)), from coupled constraints that involve variables from more than one intersections (Eqs. (5f) and (5g));
\[
\min \sum_{i=1}^{N_I} f_i^T x_i \quad \text{(5a)}
\]

\[
\text{s.t. } A_i x_i = a_i, \quad i \in \mathcal{R}, \quad \text{(5b)}
\]

\[
B_i x_i \leq b_i, \quad i \in \mathcal{R}, \quad \text{(5c)}
\]

\[
x_{i,t} \leq x_i \leq x_{i,u}, \quad i \in \mathcal{R}, \quad \text{(5d)}
\]

\[
x_{c,t}^w \in \{0,1\}, \quad i \in \mathcal{R}, \quad \text{(5e)}
\]

\[
x_{c,t}^x + x_{c,t}^y = WN - W x_{c+1,t}^n, \quad c \in \mathcal{B}_i^O, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \quad \text{(5f)}
\]

\[
x_{c,t+1}^n = x_{c,t}^n + x_{c-1,t}^y - x_{c,t}^y, \quad c \in \mathcal{B}_i^I, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \quad \text{(5g)}
\]

where \( x_i = [x_i^n, x_i^y, x_i^w, x_i^s] \) denotes the vector of variables of area \( A_i \), \( x_i^n \in \mathbb{R}^{[A_i] (T_0+1) \times 1} \).

\( x_i^y \in \mathbb{R}^{[A_i] T_0 \times 1}, x_i^w \in \mathbb{R}^{T_0 \times 1}, x_i^s \in \mathbb{R}^{(T_0-1) \times 1} \) are vectorized versions of variables \( x_{c,t}^n \equiv n_{c,t}, x_{c,t}^y \equiv y_{c,t} \),

\( x_{i,t}^y \equiv w_{i,t} \) and \( x_{i,t}^s \equiv u_{i,t} \), \( c \in \mathcal{A}_i, \quad i \in \mathcal{R}, \quad t \in \mathcal{T} \), \( x_{i,t}^y \) and \( x_{i,t}^s \) are the lower and upper bounds of the corresponding variables, \( \) while slack variables \( x_{i,t}^s \in \mathbb{R}^{(T_0-1) \times 1} \) are introduced to convert inequalities (4c) for output boundary cells of area \( A_i \) into equalities\(^1\). Expression (5a) captures the linear objective function, where \( f_i \) denotes the cost associated with variables \( x_i \), while Eqs. (5b) and (5c) capture all decoupled equality and inequality constraints whose variables belong to the same area \( \text{i.e.} \) Eqs. (4b), (4e)-(4o), (4c) excluding output boundary cells, and (4d) excluding input boundary cells. The excluded constraints are represented by the coupled equalities (5f) and (5g) which involve variables from two different areas and hence cannot be handled directly by the same IC. Note that any initialization constraints have been included in the above formulation as bound constraints.

To be able to solve the problem in a distributed manner via the ADMM algorithm, we introduce auxiliary variables \( z \) and rewrite (5) into the equivalent form:

\[
\min \sum_{i=1}^{N_I} f_i^T x_i \quad \text{(6a)}
\]

\[
\text{s.t. Constraints \( (5b), (5c), (5d) \) and \( (5e) \)} \quad \text{(6b)}
\]

\[
x_{c,t}^x + x_{c,t}^y = WN - W x_{c+1,t}^n, \quad c \in \mathcal{B}_i^O, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \quad \text{(6c)}
\]

\[
x_{c,t+1}^n = x_{c,t}^n + x_{c-1,t}^y - x_{c,t}^y, \quad c \in \mathcal{B}_i^I, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \quad \text{(6d)}
\]

\[
x_{c+1,t}^n = x_{c,t}^n, \quad c \in \mathcal{B}_i^O, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \quad \text{(6e)}
\]

\[
x_{c,t}^y - x_{c-1,t}^y, \quad c \in \mathcal{B}_i^I, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \quad \text{(6f)}
\]

For better clarity, variables \( z_{c,t}^n \) correspond to variables \( n_{c,t} \), similar to \( x_{c,t}^n \)\(^\dagger\), the same applies to the other variable sets. \( c \) also (6e) and (6f) can be equivalently expressed as \( z_{c,t}^n = n_{c,t} \),

\[\text{c} \in \mathcal{B}_i^I, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \quad \text{and} \quad z_{c,t}^y = y_{c,t}, \quad c \in \mathcal{B}_i^O, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \quad \text{respectively.}\]

The above formulation adheres to ADMM formulation (3) with \( x = [x_1^T, ..., x_{N_I}^T]^T \) and \( z = [z_1^T, ..., z_{N_I}^T]^T \), with \( z_i \) denoting the vector of elements \( c_{c,t}^n, \quad c \in \mathcal{B}_i^O, \quad t \in \mathcal{T} \) and \( c_{c,t}^y, \quad c \in \mathcal{B}_i^I, \quad t \in \mathcal{T} \).

\(^1\text{Area-related sets will be denoted with the same symbol and a subscript index to indicate the area. For example, sets \( \mathcal{E} \) and \( \mathcal{E}_i \) denote the set of ordinary cells in the whole topology and area} \mathcal{A}_i \text{, respectively.}\)
Convex set $C_z$ is empty, while $C_x = \{x|\text{Constraints (5b), (5c) (5d) and (5e)} \}$. Constraints (6c)-(6f) are coupling constraints between $x$ and $z$, representing eq. (3b) of ADMM formulation.

According to the ADMM algorithm we rewrite the problem into the augmented Lagrangian form to obtain:

$$
\min \mathcal{L}(x, z, \lambda, \mu, \nu) = \sum_{i=1}^{N_I} (f_i^T x_i + h_{C_i}(x_i)) + \frac{\rho}{2} \left( \sum_{c \in B_1^I} \sum_{t=1}^{T_c} \left( x_{c,t}^y + s_{c,t} + W_z z_{c+1,t} - WN + \lambda_{c,t} \right)^2 
+ \sum_{c \in B_1^I} \sum_{t=1}^{T_c} \left( x_{c,t}^n - x_{c+1,t}^n + x_{c,t}^y - z_{c-1,t}^y + \mu_{c,t} \right)^2 + \|x_i - z_i + \nu_i\|_2^2 \right),
$$

where $\nu, \lambda,$ and $\mu$ are the scaled Lagrange multipliers of constraints (6e)-(6f), (6c), and (6d), respectively. Hence, ADMM consists of iteratively solving $\mathcal{L}(x, z, \lambda, \mu, \nu)$ over $x$ (step 1), then over $z$ (step 2), and then updating $\lambda, \mu, \nu$ (step 3). After the $k$th iteration of the ADMM algorithm, vectors $x_i^k, z_i, \lambda_i^k, \mu_i^k, v_i^k$ have been attained; next, we show how the problems that appear in iteration $k + 1$ can be solved.

Step 1 of MIQP-ADMM consists of solving the problem over $x$. The problem to be solved is decomposable for each intersection; for the $i$th intersection the problem is $x_i^{k+1} = \arg \min \mathcal{L}(x_i, z_i^k, \lambda_i^k, \mu_i^k, v_i^k)$. which can be defined as:

$$
\min \frac{2}{\rho} f_i^T x_i + \sum_{c \in B_1^I} \sum_{t=1}^{T_c} \left( x_{c,t}^y + s_{c,t} + W_z z_{c+1,t} - WN + \lambda_{c,t} \right)^2 
+ \sum_{c \in B_1^I} \sum_{t=1}^{T_c} \left( x_{c,t}^n - x_{c+1,t}^n + x_{c,t}^y - z_{c-1,t}^y + \mu_{c,t} \right)^2 + \|x_i - z_i + \nu_i\|_2^2
$$

s.t. Constraints (5b), (5c), (5d) and (5e),

Formulation (7) is a MIQP which can be solved with MIQP solvers. The only limitation in (7) is that the values of constants $z_{c+1,t}, c \in B_1^I$ and $z_{c-1,t}, c \in B_1^I, t \in T$ are not locally available; however, these quantities can easily be communicated to $IC_i$ as they belong to immediate neighboring intersections.

Step 2 of ADMM consists of solving the problem over $z$. The problem to be solved is $z_i^{k+1} = \arg \min \mathcal{L}(x_i^{k+1}, z_i, \lambda_i^k, \mu_i^k, v_i^k)$ which can be decomposed similarly to step 1 of ADMM, so as $IC_i, i \in R$ has to solve the following unconstrained QP:

$$
\min \|x_i^{k+1} - z_i + v_i\|^2 + \sum_{c \in B_1^I} \sum_{t=1}^{T_c} \left( x_{c-1,t}^y + s_{c-1,t} + W_z z_{c,t} - WN + \lambda_{c-1,t} \right)^2 
+ \sum_{c \in B_1^I} \sum_{t=1}^{T_c} \left( x_{c+1,t}^n - x_{c+1,t+1}^n + x_{c+1,t}^y - z_{c,t} + \mu_{c+1,t} \right)^2
$$

Formulation (8) is separable for each $z$ variable, hence the solution for the above problem can be analytically obtained through the solution of one-dimensional problems of the form
Algorithm 1: Distributed MIQP-ADMM algorithm for the solution of $N'TSC$ problem

for (each intersection controller $IC_i$) do

Init.: Set $z_i^0 = \lambda_i^0 = \mu_i^0 = \nu_i^0 = 0$.

repeat

1. Compute $x_i^{k+1}$ by solving problem (7) (Step 1).
2. Communicate to neighboring intersection $j \in N_i$ the values of $x_{c,t}^{n,k+1}, x_{c,t}^{r,k+1}, c \in \{B_i \cap B_j^F\}, t \in \mathcal{T}$.
3. Compute $z_i^{k+1}$ by solving (8) (Step 2).
4. Update the values of dual variables $\lambda_i^{k+1}, \mu_i^{k+1}, \nu_i^{k+1}$ using eqs. (9)-(11). Additionally update $\lambda_{c,t}$ and $\mu_{c,t}$, $c \in \{B_i^F \setminus B_i\}, t \in \mathcal{T}$ using Eqs. (9) and (10) (Step 3).

until (Convergence achieved)

end for

\[
\begin{align*}
\min \{az^2 + bz\}, \text{ whose solution is } z &= -b/(2a). \text{ Similar to step 1, the only limitation is that } \\
x_{c,t}^{n,k+1}, x_{c,t}^{r,k+1}, \lambda_{c,t}^{k+1}, \mu_{c,t}^{k+1}, \nu_{i}^{t+1}, c \in B_i^F \text{ and } x_{c,t}^{n,k+1}, x_{c,t}^{r,k+1}, \mu_{c,t}^{k} \in B_i^O, \text{ are not known to } IC_i \text{ and need to be communicated to } IC_i \text{ from adjacent intersections. Hence, assuming that these quantities are known to appropriate intersections, } z_i^{k+1}, i \in R \text{ can be computed analytically by solving } N_I \text{ problems in parallel, one at each intersection.} \\
& \text{Step 3 of ADMM consists of simply updating the dual variables associated with the coupled equality constraints. } IC_i \text{ needs to perform the following updates, for } t \in \mathcal{T}: \\
\lambda_{c,t}^{k+1} &= x_{c,t}^{n,k+1} + x_{c,t}^{r,k+1} + Wz_{c,t}^{n,k+1} - WN + \lambda_{c,t}^{k}, \quad c \in B_i^O, \\
\mu_{c,t}^{k+1} &= x_{c,t}^{n,k+1} - x_{c,t}^{r,k+1} + y_{c,t}^{k+1} - \lambda_{c,t}^{k+1} + \mu_{c,t}^{k}, \quad c \in B_i^l, \\
\nu_i^{t+1} &= x_{i}^{k+1} - z_i^{k+1} + \nu_i^{k}. \quad (11)
\end{align*}
\]

The computation of updates (11) are based only on local information, while updates (9) and (10) require knowledge of terms $z_{c,t}^{n,k+1}, c \in B_i^O$ and $z_{c,t}^{r,k+1}, c \in B_i^l, t \in \mathcal{T}$ which can be obtained from neighboring intersections.

From the above analysis, it is clear that each ADMM iteration requires three computation and local communication steps. Communication step 1 can be eliminated as values $z_{c,t}^{n,k+1}, c \in B_i^O$ and $z_{c,t}^{r,k+1}, c \in B_i^l, t \in \mathcal{T}$ needed in the $k$th iteration have already been obtained from communication step 3 of the previous iteration. In addition, communication step 3 can be avoided if $IC_i$ explicitly computes the $z$-values it needs for computation step 3. In our case, computing the necessary $z$-values of boundary cells of neighboring intersections carries very little computational cost compared to the overall cost of computation steps 1 and 2, so it is beneficial to trade-off one communication step with some extra computation. To summarize, each iteration of ADMM requires three computation steps and one communication step after computation step 1. Algorithm 1 outlines MIQP-ADMM for the solution of $N'TSC$. The intuition behind the proposed MIQP-ADMM algorithm is that in each iteration single intersection subproblems are optimized, followed by minimization of the error in the estimation of the input/output boundary flows which are associated with neighboring intersections.
5.2 SP-ADMM Solution Algorithm

The main disadvantage of MIQP-ADMM is that the solution of problem (7) cannot be guaranteed in polynomial time as it is MIQP and hence non-convex. To overcome this issue we rewrite (6) in the equivalent form

\[
\min \sum_{i=1}^{N_l} f_i^T x_i \tag{12a}
\]

s.t. Constraints (5b), (5c), (5d), (6c) - (6f)

\[
0 \leq x_{i,t}^w \leq 1, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \tag{12b}
\]

\[
z_{i,t}^w = x_{i,t}^w, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \tag{12c}
\]

\[
z_{i,t}^w - z_{i,t-1}^w \leq u_{i,t}^w \leq z_{i,t}^w + z_{i,t-1}^w, \quad i \in \mathcal{R}, \quad t \in \mathcal{T}, \tag{12d}
\]

\[
-z_{i,t}^w + z_{i,t-1}^w \leq u_{i,t}^w \leq 2 - z_{i,t}^w - z_{i,t-1}^w, \quad i \in \mathcal{R}, \quad t \in \mathcal{T}, \tag{12e}
\]

\[
\sum_{t = 1}^{t + G_{\text{min}}} u_{i,t} \leq 1, \quad i \in \mathcal{R}, \quad t = 1, \ldots, T_0 - G_{\text{min}}, \tag{12f}
\]

\[
\sum_{t = 1}^{t + G_{\text{max}} - 1} z_{i,t}^w \leq G_{\text{max}}, \quad i \in \mathcal{R}, \quad t = 1, \ldots, T - G_{\text{max}} + 1, \tag{12g}
\]

\[
\sum_{t = 1}^{t + G_{\text{max}} - 1} z_{i,t}^w \geq 1, \quad i \in \mathcal{R}, \quad t = 1, \ldots, T - G_{\text{max}} + 1, \tag{12h}
\]

\[
z_{i,t}^w = z_{i,t}^\text{init}, \quad i \in \mathcal{R}, \quad t = -G_{\text{max}} + 1, \ldots, 0, \tag{12i}
\]

\[
z_{i,t}^w \in \{0, 1\}, \quad t \in \mathcal{T}, \quad i \in \mathcal{R}, \tag{12j}
\]

where the binary variables and associated constraints are transferred from the \(x\)-variables to the \(z\)-variables. Applying ADMM to reformulation (12) and optimizing with respect to \(x\)-variables (Step 1) yields:

\[
\min \frac{2}{\rho} f_i^T x_i + \sum_{c \in \mathcal{B}_l^T} \sum_{t=1}^{T_n} \left( x_{c,t}^\gamma + x_{c,t}^\nu + W_{c+1}^T - W_{c}^T + \lambda_{c,t}^{\gamma} \right)^2 \\
+ \sum_{c \in \mathcal{B}_l^T} \sum_{t=1}^{T_n} \left( x_{c,t}^\mu - x_{c,t}^\nu + x_{c,t}^\nu - z_{c,t-1}^\nu + \mu_{c,t}^{\nu} \right)^2 + \|x_i - z_i^k + \nu_i^k\|_2^2 \tag{13a}
\]

s.t. Constraints (5b), (5c), (5d) and (12c).

As (13) contains no binary variables, it is a convex QP problem that can easily be solved with standard optimization solvers in polynomial time.

Similar to MIQP-ADMM, Step 2 of SP-ADMM can be decomposed into the same one-dimensional QP problems for \(x_{c,t}^\gamma\), \(c \in \mathcal{B}_l\), and \(z_{c,t}^\nu\), \(c \in \mathcal{B}_0\), \(t \in \mathcal{T}\) variables, however the problem cannot be decoupled for variables \(z_{c,t}^\nu \in \{0, 1\}, \quad t \in \mathcal{T}\) at \(\mathcal{IC}_t\). Step 3 of SP-ADMM is simply to update the Lagrange multipliers similar to step 3 of MIQP-ADMM.

Next, the solution of the \(z_{i,t}^w\) subproblem in step 2 of SP-ADMM is discussed. Because \(z_{i,t}^w\) are decoupled from the other \(z\)-variables, the particular subproblem can be written as:
\[
\begin{align*}
\min_{t \in T} & \sum_{t \in T} (z_{t,i}^w - x_{t,i}^{w,k+1} + \xi_{t,i})^2, \\
\text{s.t. Constraints (12e) - (12k)},
\end{align*}
\]

Although (14) is MIQP it can be transformed into MILP by noting that the objective function (14a) can be written as:

\[
\sum_{t \in T} (z_{t,i}^w f_{1.i,t} + (1-z_{t,i}^w f_{2.i,t}),
\]

where \( f_{1.i,t} = (1-x_{t,i}^{w,k+1} + \xi_{t,i})^2 \) and \( f_{2.i,t} = (0-x_{t,i}^{w,k+1} + \xi_{t,i})^2 \), are the costs of having a green light at intersection \( i \) and time \( t \) for phases 1 and 2 respectively. Despite the problem being MILP and combinatorial in nature, we have developed an optimal solution procedure, which is based on building an appropriate graph and solving a shortest path problem using Dijkstra’s algorithm.

For a two phase intersection, the graph is comprised of vertices \((t, p), t \in T, p \in \{1,2\}\) indicating the possibility that a green light for phase \( p \) starts at time \( t \). The edges of the graph represent the transition between phases, while the associated edge costs indicate the cumulative cost of making the particular transition. For instance, state \((t,1)\) can be reached if the green light of phase 2 started at \( t' \in \{t-G_{\max}, \ldots , t-G_{\min}\} \) and the duration of the green is of length \( t-t' \). In this case the transition cost is \( f_e(t,t'-2) = \sum_{t=1}^{t'-1} f_{2,i,t} \) which indicates the cumulative cost of having \( z_{t,i}^w = \ldots = z_{t,i-1}^w = 1 \). Similarly, edges out of state \((t,1)\) and into state \((t',2)\) can be defined with costs \( f_e(t,t'-1) = \sum_{t'=t}^{t'-1} f_{1,i,t} \), \( t' \in \{t+G_{\min}, \ldots , t+G_{\max}\} \). For example, a transition from state \((10,1)\) to state \((15,2)\) implies a green light of phase 1 for time units 10, 11, 12, 13 and 14 with transition cost \( f_e(10,5,1) = \sum_{t=10}^{14} f_{1,i,t} \). By properly building a graph of traffic light states and phase transitions, the minimal cost of problem (14) is obtained by finding the path with the optimal combination of transitions for the time period considered. This can be achieved by computing the shortest path (e.g. using Dijkstra’s algorithm) from a source node added at the start to a destination node added at the end of the optimization time period. The edges that are visited indicate the best transitions between the two phases of the considered intersection that also respect the minimum and maximum green constraints. Special attention should be paid at the first few time units to integrate any initial traffic light conditions, and also at the end of the considered time to incorporate incomplete transitions. This solution procedure guarantees low polynomial-complexity and optimal solution to (14).

### 5.3 Distributed Cumulative Departure Rate Heuristic

Following the attainment of an initial distributed solution to NTSC from MIQP-ADMM and SP-ADMM, a distributed heuristic is applied to further improve the solution quality for the time-window of interest, i.e. \( T_w \). Towards this direction, the distributed cumulative departure rate heuristic (DCDRH) has been developed, originally proposed as a heuristic for the NTSC problem in a centralized setting (5). The main idea of DCDRH is to attempt to match the cumulative departure rate (CDR) resulting from \( NTSC \), with the CDR obtained from binary decisions for each movement of an intersection. The CDR of phase \( p \) of intersection \( i \) at time \( \tau \) is defined as:
Algorithm 2: DCDRH Scheme

for (each intersection controller IC$_i$, $i \in \mathcal{R}$) do
    \textbf{Init.}: Compute $C_{p,0,i}$, $\forall i$, from MIQP-ADMM and SP-ADMM and set CDR of binary decision to: $C_{p,0,i} = 0$, $p = \{1,2\}$.
    for ($\tau = 1,\ldots,T_w$) do
        Communicate $n_{c,\tau}$, $c \in B^1_i$ to IC$_j$ where $c - 1 \in B^0_j$.
        Wait for reception of $n_{c+1,\tau}$, $c \in B^0_j$ from other ICs.
        if (Constraints (4h) - (4m) are not violated) then
            For $w_{i,\tau} = k$, $k \in \{0,1\}$ compute $y_{c_1,i}^k$ and $y_{c_2,i}^k$, $c_1 \in I_{1,i}$ and $c_2 \in I_{2,i}$, based on Eq. (1).
            Set $C_{p,\tau,i}^k = C_{p,\tau-1,i} + \sum_{c \in D_{p,j}} y_{c,i}^k$, $p = \{1,2\}$.
            Set $w_{i,\tau} = \arg \min_{k \in \{0,1\}} \left\{ \sum_{p=1}^2 |C_{p,\tau,i}^k - \tilde{C}_{p,\tau,i}^k| \right\}$.
        else
            Set $w_{i,\tau}$ to a value that violates no constraint.
        end if
        Update $y_{c,\tau}$, $c \in A_i$ and $C_{p,\tau,i}$ based on $w_{i,\tau}$.
        Communicate $y_{c,\tau}$, $c \in B^0_j$ to IC$_j$ where $c + 1 \in B^1_j$.
        Wait for reception of $y_{c-1,\tau}$, $c \in B^1_j$ from other ICs.
        Update $n_{c,\tau+1}$, $c \in A_i$ using Eq. (2).
    end for
end for

$$C_{p,\tau,i} = \sum_{t=1}^{\tau} \sum_{c \in D_{p,i}} y_{c,t} = C_{p,\tau-1,i} + \sum_{c \in D_{p,i}} y_{c,p,\tau}$$

where $D_{p,i}$ denotes the set of cells immediately downstream all movements of phase $p$ of intersection $i$. DCDRH is outlined in Algorithm 2. Note that DCDRH requires two rounds of communication between adjacent intersections for each time unit which are necessary for the distributed computation of variables $y_{c,t}$ and $n_{c,t}$.

6. SIMULATION RESULTS

The effectiveness of the proposed distributed algorithms MIQP-ADMM and SP-ADMM is evaluated for a four-intersection and eight-intersection topology of two-way, two-lane alternating direction arterials similar to the one shown in Fig. 1. The free-flow speed is 50km/h and each time-step is equivalent to 5s, resulting in 70m-long cells. Adjacent intersections are 4 cells apart which implies that the distance between them is 280m. The capacity of all cells is $Q = 5$ veh./time unit, and the jam density is $N = 20$ veh./cell. The ratio of shock-wave speed over free flow speed is $W = 0.75$.

The maximum and minimum green times are assumed to be equal to 15s and 40s respectively, i.e. $G_{\text{min}} = 3$ and $G_{\text{max}} = 8$. The time horizon considered is $T_h = 2160$ time units ($3h$), with traffic being generated for 720 time units ($1h$). The time horizon is temporally decomposed into 10min intervals ($T_w = 120$). In the evaluation, we consider four randomly generated traffic scenaria cor-
responding to 10%, 30%, 50% and 70% congestion level\(^2\). Traffic stems from all origin cells and flows across the network to the opposite destination cell both horizontally (e.g. from cell 201 to 210 in Fig. 1) and vertically (e.g. from cell 301 to 309 in Fig. 1) in both directions. In the ADMM algorithm, we set \( \rho = N_I \), while for faster convergence we consider an over-relaxation parameter \( a = 1.6 \) (see (23)). The centralized optimal solution to all optimization problems was obtained using Gurobi (27), while genetic algorithms (GAs) were considered to obtain close to optimal centralized solutions to NTSC using the global optimization toolbox of Matlab; any GA-based demonstrated results reflect the best solution obtained among ten independent runs. Experiments were conducted on a desktop computer with an Intel i5-3470 processor (3.2GHz) with 8GB of DDR3 RAM running 64-bit Windows 7.

In order to investigate ADMM convergence, Fig. 2 depicts the total travel time (TTT) that was obtained for the first time-window from MIQP-ADMM and SP-ADMM respectively, when the ADMM execution is stopped at one particular iteration. The figure indicates that MIQP-ADMM converges within 10 iterations while SP-ADMM requires around 70-80 iterations to converge under the 30% congestion level for both topologies. Similar behavior has been observed in other traffic scenarios; hence, a stopping criterion of 10 and 100 iterations has been adopted for MIQP-ADMM and SP-ADMM, respectively. Having a small number of iterations is important as both algorithms require the same amount of communication in each iteration; hence, in terms of communication MIQP-ADMM is superior to SP-ADMM.

Fig. 3 depicts the total execution time required to compute the traffic signal plans for the next time-window using the two proposed algorithms. It can be seen, that SP-ADMM is superior in terms of execution time despite the fact that it executes ten times more iterations compared to MIQP-ADMM. The reason is that each iteration of SP-ADMM is significantly faster due to the polynomial computational complexity of the involved subproblems. On the contrary, in MIQP-ADMM each IC solves one MIQP subproblem per iteration which is of exponential complexity.

To ensure online execution, each MIQP problem is restricted to run for a maximum of 40s; for

\(^2\)Congestion level is defined as the ratio of the average to the maximum possible arrival rate over the considered traffic generation period at all traffic sources.
FIGURE 3 Total execution time per intersection for different time-windows to evacuate traffic for the different scenarios considered with topology a) 2x2, and b) 4x2.

FIGURE 4 Comparison between the optimal solution of NTSC and the solutions obtained from MIQP-ADMM, SP-ADMM and GA algorithms with topology a) 2x2, and b) 4x2.

this reason, the execution time of MIQP-ADMM is 400s for almost all time periods. Only for the last time-window MIQP-ADMM is able to find optimal solutions to the MIQP problems and finish faster. Note that only the scenario for the 10% congestion level is shown for the MIQP-ADMM algorithm, because all other scenarios gave a very similar execution time pattern. As expected, the execution time of each intersection does not change between the considered two topologies due to the ability of the proposed algorithms to spatially distribute the computation among intersections. In addition, notice that for the 70% congestion level the number of 10-min time-windows required to evacuate traffic are 12 and 16 for the 2x2 and 4x2 topologies, respectively, which indicates that there is significant congestion for the particular scenario; in all other traffic scenarios full evacuation takes place in 7-8 time-windows which indicates that congestion does not heavily build up. Note that both algorithms are suitable for online execution, as they always finish execution before the start of the next time windows (time-limit:600s).
Fig. 4 depicts the relative deviation of the proposed algorithms from the optimal TTT obtained via a MILP solver for the considered traffic scenarios. The figure indicates that SP-ADMM provides better results for the lower congestion scenarios, while for 50% and 70% congestion levels MIQP-ADMM and SP-ADMM have similar performance. Compared to the GAs, both algorithms provide significantly better results for 10-50% congestion levels, but for the 70% congestion level GAs performs better. Nonetheless, on average the proposed algorithms MIQP-ADMM and SP-ADMM outperform GAs as their average deviation from optimality is 12.2% and 10.4%, respectively, whereas GAs deviate 20.8% from the optimal.

7. CONCLUSIONS & FUTURE WORK
In this paper, two distributed online solution strategies for NTSC based on non-convex ADMM formulations have been proposed. Both algorithms on average achieve results close to 10% from optimality, significantly better than a genetic algorithm alternative, and provide a different trade-off between communication and computation. For future work, we intend to examine large-scale topologies and investigate whether the observed properties of the proposed algorithms are maintained. In addition, ways to reduce the execution time of the MIQP subproblems should be investigated, as well as other reformulations that have lower computation and communication requirements. Finally, the performance of the proposed algorithms should be examined under traffic perturbations and compared with other TSC strategies.

ACKNOWLEDGMENTS
This work is partially funded by the European Research Council Advanced Grant FAULT-ADAPTIVE (ERC-2011-AdG-291508).

REFERENCES


