

Distributed Traffic Signal Control using the Cell Transmission Model via the Alternating Direction Method of Multipliers

Stelios Timotheou, Christos G. Panayiotou and Marios M. Polycarpou
KIOS Research Center for Intelligent Systems and Networks
University of Cyprus
{timotheou.stelios, christosp, mpolycar}@ucy.ac.cy

Abstract—Traffic signal control is a key ingredient in intelligent transportation systems to increase the capacity of existing urban transportation infrastructure. However, to achieve optimal system-wide operation it is essential to coordinate traffic signals at various intersections. In this paper we model the multiple-intersections traffic signal control problem using the cell transmission model as a mixed integer linear program. The solution of the problem is facilitated by its special structure which allows both temporal and spatial decomposition. Temporal decomposition is employed to reduce problem size by solving subproblems of smaller time-window compared to the original problem. Temporal subproblems can be further spatially decomposed into subproblems associated with different intersections, which are jointly solved by exchanging messages between neighboring intersections. The proposed distributed solution strategy is comprised of two phases. First, the relaxed linear problem is reformulated and distributedly solved via the Alternating Direction Method of Multipliers. Second, two distributed rounding schemes are developed to solve the original problem. Simulation results indicate that the proposed solution strategy is scalable to large transportation topologies, suitable for online execution and provides close to optimal results.

Keywords: intelligent transportation systems, traffic signal control, cell-transmission model, online, distributed, mixed-integer linear programming, alternating direction method of multipliers.

I. INTRODUCTION

Over the past few decades there has been a steady growth of vehicle and total miles traveled leading to an increase in traffic congestion [2]. Given that major road constructions in cities are both difficult and costly, one of the most effective ways to alleviate congestion is to increase the capacity of the existing infrastructure using traffic signal control. Traffic signal control can bring substantial reduction to traffic congestion, leading to improved conditions both for the drivers (better travel times, safety and convenience) and the environment (reduced air pollution and energy consumption). Furthermore, recent advancements in electronics, sensing, and ICT (information and communication technology) allow the real-time collection and processing of traffic data, as well as the deployment of intelligent controllers for the efficient operation of a transportation system.

Nevertheless, controlling the traffic signals of a transportation network constitutes a significant challenge due to the

large-scale nature and complexity of the problem, the uncertain and dynamic behavior of the network (e.g. weather, accidents, events) and the patterns of different driver behaviors. For this reason, several different approaches have been proposed; many strategies apply to single intersections, others use historical data to determine fixed plans, while a family of strategies attempt to dynamically decide on the traffic signal timing plans in a distributed and online manner. Traffic signal variables typically controlled are the cycle length, split plan, and offset. *Cycle length* is the time required for a complete sequence of signal indications. The *split plan* refers to the time assigned to different *phases* (simultaneous movement combinations that have the right-of-way) during a signal cycle. Finally, the *offset* is used to coordinate phases of adjacent intersections to reduce vehicle stops [3].

A large body of literature considers the single intersection traffic signal control problem, neglecting interrelation effects with other intersections. These approaches aim to optimize some measure-of-interest (mean delay, mean stoppage time, throughput) based on the state of the intersection and include mathematical programming [4], stochastic control [5] as well as computational and artificial intelligence techniques (e.g. fuzzy logic [6], expert systems [7], etc.). Nonetheless, by considering intersections atomically, the offset between intersections is not optimized leading to frequent vehicle stops. Also measures-of-interest are optimized locally instead of globally and may lead to poor global performance.

The majority of techniques consider the multiple intersections traffic signal control (MITSC) problem. *Fixed* or *pre-timed* signal control strategies optimize *offline* the signal timing plans based on historical data so that fixed signal programs are applied for different periods of the day. Fixed-time MITSC methods either attempt to adjust the offset between adjacent intersections so as to maximize progression along multiple corridors using MILP methods, e.g. in MULTIBAND [8], and global optimization techniques [9], or optimize split plans and cycle according to some measure of effectiveness (MOE) that combines different traffic metrics such as delay, minimum number of stops and throughput, e.g. TRANSYT [10]. This is accomplished using some traffic model (e.g. TRANSYT-13 uses both the platoon dispersion model and the cell transmission model (CTM)) to capture traffic dynamics and different optimization techniques to optimize signal programs. CTM-based optimization approaches usually rely on the development of MILP formulations [11], [12] which are usually NP-hard; hence, meta-heuristic techniques such as

A preliminary version of this work appears in [1].

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

genetic algorithms [13] and greedy heuristics [14] 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 (e.g. accidents, weather conditions, socio-cultural events).

To account for stochastic variations of traffic flows, several *online* adaptive traffic signal control (ATSC) systems have been developed. These approaches collect information from different sources on-demand, and use them to adaptively optimize traffic signal plan parameters such as splits, offsets and cycle, (e.g. SCOOT, [15], MOTION [16]) or select the best from a library of pre-calculated signal plans (e.g. SCATS [17]). Early commercially deployed systems usually rely on centralized architectures where one computing unit decides for all intersections (e.g. SCOOT, SCATS) or hierarchical architectures where one part of the decision is central and another part is decentralized. For example, in MOTION the cycle time and offsets are centrally decided every 10-15 minutes for the whole network, while traffic signal splits, phase sequence and small cycle modifications are determined at individual intersections based on real-time traffic fluctuations. RHODES system [18] is also hierarchical with local and network decision modules, but contrary to older ATSC systems that re-actively consider real-time information, it predicts future traffic flows for proactive decision making. Model predictive control approaches are also centralized proactive ATSC strategies that attempt to determine system-wide optimal signal plans in a rolling horizon fashion [19], [20]. To deal with the large-scale nature of the problem in both space and time, these approaches usually rely on a coarse representation of the transportation infrastructure and continuous instead of binary green traffic light allocations. Coarse infrastructure representation means that large road segments between adjacent intersections are modeled with single links, failing to capture rapidly evolving traffic phenomena, while by avoiding the explicit modeling of red/green switching for consecutive time-units suboptimal solutions are derived.

Fully distributed online algorithms dynamically adjust the signal parameters of multiple intersections based on the network state without relying on a centralised computing unit. Due to the complex and large-scale nature of the problem these algorithms often rely on low-complexity but suboptimal artificial and computational intelligence techniques. One such technique is reinforcement learning (RL), which offers the ability to learn relationships between observed states and actions in an uncertain and dynamic environment by maximizing a value function. Thorpe [21] considered the traffic lights as the acting agents and developed an approach which constructed a traffic-light-based value function which approximates the aggregate waiting time of cars (this can result in a huge number of states), while Wiering et al. [22] considered car-based value functions for the estimation of waiting times, while the traffic signal plans were generated by car voting. A distributed computational market-based technique that also considers vehicles - intersection agent cooperation is presented in [23]; it was suggested to form computational markets at each intersection

so that vehicles agents can trade the use of capacity. This approach can be difficult to implement in practice as it requires vehicle drivers to monetize access to road resources on-the-fly. Multiagent learning with cooperation between neighboring intersections has also been considered in [24], [25]; these approaches adaptively improve the system-wide performance of the network using game-theoretic and reinforcement learning methods, but do not provide optimal results. Recently, a control approach based on backpressure routing, adopted from communication networks, has been proposed [26]. This approach eliminates the aforementioned problems by allowing each intersection to decide on the right-of-way based only on local information about the traffic states and queue lengths. The authors prove that the proposed algorithm attains maximum network throughput for each time instance. Nevertheless, the approach relies only on queue lengths and neglects vehicle delay, while the information utilized are not sufficient to optimize the offset between intersections.

In this paper a cooperative distributed online¹ algorithm for system-wide optimization of traffic signal control based on CTM is proposed. This approach departs from the traditional view of considering optimization techniques for the solution of MITSC using fine-grained macroscopic models, such as CTM, only for offline centralised systems. The algorithm is based on spatially and temporally decomposing the problem and iteratively solving the produced subproblems by individual intersection controllers. *Spatial decomposition* is achieved by dividing the considered transportation topology into single-intersection areas; *temporal decomposition* is achieved by separating the time-horizon considered into small time-windows (e.g. 10 mins). In this way, signal timing plans are adapted online every one temporal time-window, while during this time the subsequent signal plans are computed. Because the subproblem solution presumes information associated with neighboring intersections, intersection controllers cooperate with each other by locally exchanging appropriate data. To achieve this, the centralized problem is reformulated in a way that facilitates decomposition, and the alternating direction method of multipliers (ADMM) [27] is employed to arrive at a distributed solution to the relaxed linear problem. After the relaxed solution is obtained, distributed rounding schemes are employed at each intersection to derive appropriate binary values for the decision variables which determine the traffic signal split plans. The fully distributed nature of the proposed algorithm, overcomes the main disadvantages of centralized and hierarchical systems such as SCOOT, SCATS, MOTION and RHODES, that require the deployment of an expensive communication network, are not scalable, while failure of the centralized computer can bring down the entire traffic signal network. In addition, contrary to other distributed algorithms that myopically control traffic signals based on the current network state, our approach proactively optimizes system-wide

¹The key defining characteristic of *online adaptation* is that soon after input information is received regarding the state of the system, decision making takes place resulting in the adaptation of traffic signal plans. The proposed system has this feature as it adapts every few minutes based on short-term prediction of the system state; nonetheless, it is not a *real-time system* in the sense that adaptation does not occur on a second-by-second basis.

performance for a look-ahead time-window using the CTM.

The contributions of this paper are the following:

- Introduction of a spatial and temporal decomposition framework for CTM which can be used in several transportation problems.
- Development of a distributed online cooperative solution procedure of the MITSC Problem using CTM; the procedure can also be employed in a parallel fashion on a single computing platform (e.g. multi-core computer or computer-cluster at a traffic operations center) to speedup the solution procedure.
- Efficient approximation of the optimal solution through the development of two distributed solution rounding schemes.

The remainder of this paper is organized as follows. Section II-A, outlines the cell transmission model which captures the traffic dynamics. Section III explains temporal decomposition and presents the centralized formulation of the multiple-intersection traffic signal control problem. In section IV, the spatial decomposition framework is explained, while in section V the distributed ADMM solution to the relaxed problem and the two distributed rounding schemes are developed. Section VI, describes the performance evaluation of the proposed solution approach in terms of convergence, optimality and scalability. Finally, section VII 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. The superscripts $(\cdot)^T$ and $(\cdot)^{-1}$, denote the transpose and the matrix inverse respectively. $\|\mathbf{z}\|_2$ denotes the Euclidean norm of a vector \mathbf{z} . Operators $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cap \mathcal{B}$, and $\mathcal{A} \setminus \mathcal{B}$, denote the union, intersection and set difference of sets \mathcal{A} and \mathcal{B} , respectively, while $|\mathcal{A}|$ denotes the cardinality of set \mathcal{A} . $|x|$ denotes the absolute value of variable x . $h_C(\mathbf{x})$ is the indicator function which is equal to zero if $\mathbf{x} \in C$ and $+\infty$ otherwise.

II. PRELIMINARIES

A. Cell Transmission Model

The Cell Transmission Model (CTM) [28] is a discrete analog of the well-known first-order Lighthill-Whitham-Richards (LWR) continuum flow model [29], [30] which is based on the fundamental relationship for the conservation of flow and supplemented by the assumption that traffic flow, at road point x at time t is only a function of traffic density. The form of this flow-density function is specified using a flow-density model (e.g. Greenshields, Pipes, or Van Aerde models [31]) and calibrated by estimating macroscopic traffic flow parameters (such as the *free-flow speed*, *jam-density*, *maximum flow* or *capacity* and *speed-at-capacity*) for a specific road configuration.

In the CTM both time and space are discrete. Each road segment is divided into homogeneous sections called *cells*, while time is partitioned in a way that one vehicle takes one time-unit to travel through one cell at free-flow speed (the speed of vehicles when density is zero). When CTM assumes a piecewise-linear flow-density relationship, the LWR

model is simplified to the following difference equations which constitute the fundamental relationships of CTM.

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

$$\bar{y}_{c+1,t} = y_{c,t} \quad (2)$$

$$n_{c,t+1} = n_{c,t} + \bar{y}_{c,t} - y_{c,t} \quad (3)$$

In the above equations, $\bar{y}_{c,t}$, $y_{c,t}$ and $n_{c,t}$ represent the number of vehicles entering cell c , leaving cell c and are inside cell c respectively, at time $[t, t+1)$. $Q_{c,t}$ and $N_{c,t}$ represent the maximum number of vehicles that can flow through and reside into cell c at time t , respectively. $W_{c,t}$ is the ratio between the shock-wave propagation speed and the free-flow speed and indicates how fast a vehicle queue is formed. 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 vehicle leaving cell c is limited either by the number of vehicles in the cell, the capacity of the cell for outflow vehicles, the capacity of the successor cell for inflow vehicles and the space left in the successor cell when a queue is forming. Eqs. (2) - (3) ensure flow conservation at cell c .

The popularity of CTM is based on its simplicity, its capacity to capture phenomena that are found in first-order continuum flow models and its ability to model different boundary phenomena such as origin, destination, merge, diverge and general intersection cells [32]. Although, the standard CTM model cannot capture platoon dispersion, it is quite useful in modeling the spatial extent of queues and hence is more appropriate for signalized networks with closely spaced intersections as in urban environments [33]. For this reason, it has been adopted in TRANSYT-13 as an alternative to the platoon dispersion model [10]. In section III, CTM is utilized to derive a MILP formulation for the optimal operation of traffic light signals.

B. Alternating Direction Method of Multipliers

The alternating direction method of multipliers (ADMM) is a powerful method for solving mathematical optimization problems of the form

$$\min \quad f(\mathbf{x}) + g(\mathbf{z}) \quad (4a)$$

$$s.t. \quad \mathbf{Ax} + \mathbf{Bz} = \mathbf{d} \quad (4b)$$

$$\mathbf{x} \in \mathcal{C}_x, \mathbf{z} \in \mathcal{C}_z \quad (4c)$$

where $\mathbf{x} \in \mathcal{C}_x \subseteq \mathbb{R}^{M_x \times 1}$, $\mathbf{z} \in \mathcal{C}_z \subseteq \mathbb{R}^{M_z \times 1}$, $\mathbf{A} \in \mathbb{R}^{M_d \times M_x}$, $\mathbf{B} \in \mathbb{R}^{M_d \times M_z}$ and $\mathbf{d} \in \mathbb{R}^{M_d \times 1}$, $f(\mathbf{x})$ and $g(\mathbf{z})$ are convex functions and $\mathcal{C}_x, \mathcal{C}_z$ are closed convex sets. ADMM has been proposed in the 1970's [34] as a method for solving large-scale convex optimization problems by allowing the decomposition of the original problem into simpler and/or smaller subproblems. Recently, ADMM has found a resurgence of interest due to its ability to provide fast, close to optimal and distributed solutions to problems arising in several areas such as machine learning [27] and signal processing [35].

For the solution of problem (4), ADMM uses the scaled augmented Lagrangian form, $\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u})$:

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

where $\mathbf{u} = \omega/\rho$ 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 $h_{C_x}(\mathbf{x})$ and $h_{C_z}(\mathbf{z})$ are indicator functions. Starting from initial values \mathbf{z}^0 and \mathbf{u}^0 , ADMM iteratively minimizes $\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \mathbf{u})$ with respect to \mathbf{x} and \mathbf{z} followed by an update of the scaled dual variables in three consecutive steps:

$$\text{Step 1: } \mathbf{x}^{k+1} = \arg \min_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \mathbf{z}^k, \mathbf{u}^k)$$

$$\text{Step 2: } \mathbf{z}^{k+1} = \arg \min_{\mathbf{z}} \mathcal{L}_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \mathbf{u}^k)$$

$$\text{Step 3: } \mathbf{u}^{k+1} = \mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{d} + \mathbf{u}^k$$

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 and imposes mild technical conditions on the problem under investigation. In the case of problem (4), the necessary and sufficient conditions for optimality reduce to the satisfaction of primal and dual feasibility which measure the extend of constraint satisfaction and objective value suboptimality, respectively. To monitor progress towards optimality, we define the *primal residual* \mathbf{r}_{pr}^k and *dual residual* \mathbf{r}_d^k of iteration k as (see [27]):

$$\mathbf{r}_{pr}^k = \mathbf{A}\mathbf{x}^k + \mathbf{B}\mathbf{z}^k - \mathbf{d} \quad (5)$$

$$\mathbf{r}_d^k = \nabla_x f(\mathbf{x}^k) + \rho \mathbf{A}^T \mathbf{u}^k = \rho \mathbf{A}^T \mathbf{B}(\mathbf{z}^k - \mathbf{z}^{k-1}) \quad (6)$$

As primal and dual feasibility indicate optimality, a reasonable stopping criterion is to ensure that the primal and dual residuals are small. The number of iteration is also used as a stopping criterion, especially when an approximate solution is needed. A comprehensive analysis of ADMM, its convergence properties and stopping conditions appears in [27].

III. CENTRALIZED PROBLEM FORMULATION

This section details the centralized mathematical formulation for the considered multiple intersection traffic signal control (MITSC) problem. MITSC involves the optimization of split plans, cycle lengths and offset of the traffic signals of multiple intersections over a time-horizon T_h . Optimization is performed for some measure-of-interest such as mean/total vehicle delay, stoppage time and throughput. Traffic dynamics are incorporated into the optimization problem through the CTM described in section II-A. Our formulation builds on existing centralized approaches such as [11] and [12], by relaxing the assumption of an initially zero-state system; this allows the temporal decomposition of the original problem into smaller subproblems. This implication is particularly important when one wants to solve problems over a large time period, as such problems are prohibitively large.

Temporal problem decomposition refers to the process of separating the time horizon T_h over which MITSC is optimized into smaller time-windows of generated traffic, T_w , and solving the problem sequentially over those periods. Nevertheless, the considered optimization time, T_o , over which each subproblem is solved is actually larger to allow all the vehicles to exit the network, as shown in Fig. 1. In fact, when *online decision making* is sought, traffic signal plans

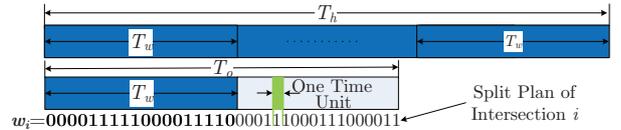


Fig. 1. Temporal decomposition of MITSC. The array of 1s and 0s represents the split plan of a particular intersection for one time window. The bold 1s and 0s indicate the decision part that will be realized for the next time-window.

must be updated every T_w time units, using only the solution corresponding to the the first T_w time units. For instance, instead of optimizing the problem over a 4-hour period, one could compute the solution every 10 minutes. In this case, a solution would have to be attained every 10 minutes, and applied to each intersection for the next 10-minute period. Apart from reducing the computational effort for the solution of the problem, temporal decomposition takes advantage of accurate prediction of traffic demand, which can only be achieved for short-term traffic predictions (10-15 minutes) [36].

Before arriving at the final MITSC formulation, the objective and constraints of the problem are introduced.

A. Objective

For the MITSC formulation, the objective is the minimization of the total travel-time (TTT) (the cumulative travel time of all vehicles). Assuming that the road network is empty both at the start and the end of the optimized time period T_o , the total travel delay can be expressed as:

$$\sum_{c \in \mathcal{D}} \sum_{t \in \mathcal{T}} t y_{c,t} - \sum_{c \in \mathcal{O}} \sum_{t \in \mathcal{T}} t D_{c,t}, \quad (7)$$

where $\mathcal{T} = \{1, \dots, T_o\}$, and $D_{c,t}$ denote the exogenous inflow traffic from an origin cell, $c, t \in \mathcal{T}$. As all vehicles enter and leave the road network, the entrance and exit times of each vehicle is included in the above expression, so that the total travel time is the sum of the travel times of all vehicles. The second term can be eliminated from the optimization problem since the starting time of each vehicle's journey is fixed a priori. Note that other objective functions can be described using CTM such as the minimization of the number of stops, or the maximization of throughput, or weighted combinations of these objectives [11]. Also it is important to consider a large enough T_o to ensure an empty network at the end of the considered period, as explained in Fig. 1; otherwise the optimal solution may involve the undesirable effect of holding-back vehicles from exiting the network because in this case their travel time will be excluded from the objective value.

B. CTM constraints

To model the traffic dynamics we consider four types of CTM cell sets: *ordinary* (\mathcal{E}), *origin* (\mathcal{O}), *destination* (\mathcal{D}) and *intersection* (\mathcal{I}). Ordinary cells have both inflow and outflow of vehicles as well as non-zero capacity at all times. Origin cells are similar to ordinary cells but instead of receiving inflow traffic from other cells they receive exogenous inflow traffic $D_{c,t}$, $c \in \mathcal{O}$, $t \in \mathcal{T}$. Destination cells sent their outflow traffic outside the network without restriction on capacity and

infinite space at their destination. Finally, intersection cells have variable capacity, $q_{c,t}$, imposed by the traffic signals phases. Despite the presence of the min operator in Eq. (1) which is nonlinear, Eqs. (1)-(3) can be transformed into the following linear programming (LP) formulation for $c \in \mathcal{A} = \mathcal{E} \cup \mathcal{O} \cup \mathcal{D} \cup \mathcal{I}$, $t \in \mathcal{T}$:

$$\max y_{c,t} \quad (8a)$$

$$\text{s.t. } y_{c,t} \leq n_{c,t}, \quad (8b)$$

$$0 \leq y_{c,t} \leq q_{c,t}, \quad (8c)$$

$$y_{c,t} \leq W_{c,t} \bar{n}_{c+1,t}, \quad (8d)$$

$$n_{c,t+1} = n_{c,t} + \bar{y}_{c,t} - y_{c,t}, \quad (8e)$$

where $\bar{n}_{c+1,t} = N_{c,t} - n_{c+1,t}$, if $c \in \mathcal{A} \setminus \mathcal{D}$ and ∞ otherwise, $\bar{y}_{c,t} = y_{c-1,t}$ if $c \in \mathcal{A} \setminus \mathcal{O}$ and $\bar{y}_{c,t} = D_{c,t}$ if $c \in \mathcal{O}$, while capacity $q_{c,t}$ is given by:

$$q_{c,t} = \begin{cases} Q_{c,t}, & c \in \mathcal{E} \cup \mathcal{O}, \\ w_{i,t} Q_{c,t}, & i \in \mathcal{R}, c \in \mathcal{I}_{1,i}, \\ (1 - w_{i,t}) Q_{c,t}, & i \in \mathcal{R}, c \in \mathcal{I}_{2,i}, \\ \infty, & c \in \mathcal{D}, \end{cases} \quad (9)$$

where $\mathcal{I}_{1,i}$ and $\mathcal{I}_{2,i}$ denote the set of cells that receive a green light during the first and second phase of the i th intersection split plans, $\mathcal{R} = \{1, \dots, N_I\}$ denotes the set of intersections, N_I the total number of intersections, and $w_{i,t}$, $i \in \mathcal{R}$, $t \in \mathcal{T}$ indicates whether phase 1 ($w_{i,t} = 1$) or phase 2 ($w_{i,t} = 0$) has the right-of-way. These variables are the *main decision variables* of the considered problem, implicitly optimizing the traffic signal split plans and cycle lengths of all intersections, and the offsets between them (also see Fig. 1). In the above LP program, constraints (8b) - (8d) force $y_{c,t}$ to be in the range $[0, \min\{n_{c,t}, W_{c,t} \bar{n}_{c+1,t}, q_{c,t}\}]$, while the maximization objective guarantees that $y_{c,t}$ will take the maximum value in this range. Without loss of generality, we will assume for the rest of the paper that the network is homogeneous.

To incorporate the different maximization expressions (8a) into our problem, a penalty minimization term $\alpha \sum_{c \notin \mathcal{D}} \sum_{t \in \mathcal{T}} t y_{c,t}$ is introduced into our primary objective function, where α is a penalty constant, eliminating to a certain extent the *unindented vehicle holding-back problem* [37].

C. Additional constraints

Apart from the constraints associated with the CTM model, we need to introduce more constraints regarding the traffic signal operation.

1) *Maintaining minimum green/red time*: For the operation of traffic lights, a minimum green/red time, G_{min} , is usually assumed for each phase. To check that the minimum green time constraint is preserved, it must be ensured that for $G_{min} + 1$ consecutive time units there is no more than one modification in the value of $w_{i,t}$. In mathematical terms this can be expressed for $i \in \mathcal{R}$ as:

$$w_{i,t} - w_{i,t-1} \leq u_{i,t}, \quad -w_{i,t} + w_{i,t-1} \leq u_{i,t}, \quad (10)$$

$$w_{i,t} + w_{i,t-1} \geq u_{i,t}, \quad 2 - w_{i,t} - w_{i,t-1} \geq u_{i,t}, \quad (11)$$

$$\sum_{\tau=t}^{t+G_{min}} u_{i,\tau} \leq 1, \quad t = 1, \dots, T_0 - G_{min} \quad (12)$$

Inequalities (10)-(11) are equivalent to $u_{i,t} = |w_{i,t} - w_{i,t-1}|$, when $w_{i,t} \in \{0, 1\}$, implying that $u_{i,t}$ indicates whether a signal change has occurred between $t - 1$ and t (green-to-red or red-to-green).

2) *Maintaining maximum green/red time*: Similar to the minimum green time, the maximum green time constraint is maintained when the same sign is not preserved for more than G_{max} time units, which can be expressed for $i \in \mathcal{R}$ as:

$$\sum_{\tau=t}^{t+G_{max}-1} w_{i,\tau} \leq G_{max}, \quad t = 1, \dots, T - G_{max} + 1 \quad (13)$$

$$\sum_{\tau=t}^{t+G_{max}-1} w_{i,\tau} \geq 1, \quad t = 1, \dots, T - G_{max} + 1 \quad (14)$$

Constraints (13) and (14) indicate that we cannot have $G_{max} + 1$ consecutive 1s and 0s, respectively.

3) *Ensuring flow conservation*: To ensure flow conservation in the network we must make sure that all entering and initially existing traffic must exit the network. Equivalently, it must be ensured that the network is empty at the end of the considered time horizon. This can be expressed as:

$$\sum_{c \in \mathcal{A}} n_{c,T_0+1} = 0. \quad (15)$$

D. Initial Condition Constraints

The *system state* at any time t is comprised of the *cell states*, denoted by $n_{c,t}$, $c \in \mathcal{A}$, $t \in \mathcal{T}$, as well as the state of the traffic lights $w_{i,t}$, $i \in \mathcal{R}$, $t \in \mathcal{T}$. Computing the next cell state requires only the current cell states in CTM, hence having the initial cell states

$$n_{c,1} = n_c^{init}, \quad c \in \mathcal{A} \quad (16)$$

is sufficient for the evolution of the traffic dynamics. On the contrary, for the correct evolution of the traffic signal decisions we need to consider at least G_{max} time units. Hence,

$$w_{i,t} = w_{i,t}^{init}, \quad i \in \mathcal{R}, \quad t = -G_{max} + 1, \dots, 0. \quad (17)$$

E. MILP formulation

In summary, the MITSC problem formulation is the following:

$$\begin{aligned} \text{MITSC} : \min & \quad \sum_{c \in \mathcal{D}} \sum_t t y_{c,t} + \alpha \sum_{c \notin \mathcal{D}} \sum_t t y_{c,t} \\ \text{s.t.} : & \quad - \text{CTM constraints for all cells (8b)-(8e)} \\ & \quad - \text{constraints (10)-(15)} \\ & \quad - \text{initial conditions (16) and (17)} \end{aligned}$$

In formulation *MITSC*, the main decision variables are the traffic light states $w_{i,t} \in \{0, 1\}$, which appear in the capacity expressions (9) of intersection cells, while there are also auxiliary variables $y_{c,t}$, $n_{c,t}$, $q_{c,t}$, $u_{i,t}$ that allow the evolution of CTM dynamics and the representation of the traffic signal constraints. Because the problem is composed of both binary ($w_{i,t}$) and continuous variables ($u_{i,t}$, $y_{c,t}$, $n_{c,t}$, $q_{c,t}$) it belongs to the class of Mixed Integer Linear Programming (MILP) and it is NP-hard to solve. By relaxing the binary variables to take continuous values ($w_{i,t} \in [0, 1]$), a lower bound on the optimal solution is obtained by solving an LP problem. The relaxed version of *MITSC* will be referred to as *RMITSC*.

IV. SPATIAL DECOMPOSITION

Solving the system-wide problem *MITSC* in a distributed setting, requires proper geographical partitioning of the network in a way that the optimal solution is obtained through the iterative solution of easy subproblems and the local exchange of information between neighboring areas. A good policy towards this direction, is to divide the considered network in areas of one intersection so that sensing, decision making, local communication (with adjacent intersections) and control is handled locally by the intersection controller (IC) of the area, similar to [24], [25]. Sensing is necessary to monitor and measure the traffic state of its controlled area. Computation is needed for predicting its incoming flows (from the outside of the network) and also for deciding about traffic signal timing plans. Targeting globally optimal solutions, ICs need to exchange messages with immediately adjacent intersections to collaboratively derive their traffic signal timing plans. Finally, control capabilities are essential for the online realization of the computed plans.

This spatial decomposition is motivated by the nature of transportation networks, as explained through the example topology in Fig. 2, 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 , $i = 1, \dots, 4$ so that each area is associated to exactly one intersection. In this distributed setting, area A_i is controlled by intersection controller IC_i , while interaction between adjacent intersections can take place only through boundary cells.

This can be understood by observing the linear eqs. (8) which describe the state evolution of an ordinary cell c , from which it is clear that the computation of its variables ($y_{c,t}$, $n_{c,t+1}$) at one particular time-step t , information is only needed from cells $c-1$ and $c+1$. Specifically, eq. (8d) requires variable $n_{c+1,t}$ from its successor cell, while eq. (8e) requires $y_{c-1,t}$ from its predecessor cell. 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 not belonging to the particular area. Particularly, in Fig. 2 cell 105 of area A_1 requires the number of vehicles in its successor cell 106 of A_2 , while cell 705 of A_1 requires the number of vehicles leaving out of its predecessor cell 704 of A_4 .

It is clear from this example that ordinary cells give rise to two different types of boundary cells: (a) input and (b) output boundary cells. An *input boundary cell* $c \in \mathcal{B}_i^I$ 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* $c \in \mathcal{B}_i^O$ 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 , \mathcal{B}_i^{IE} and \mathcal{B}_i^{OE} 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 , $\mathcal{B}_i = \mathcal{B}_i^I \cup \mathcal{B}_i^O$ and the corresponding extended set $\mathcal{B}_i^E = \mathcal{B}_i^{IE} \cup \mathcal{B}_i^{OE}$. For instance, for area A_1 we have that $\mathcal{B}_1^I = \{705, 506\}$,

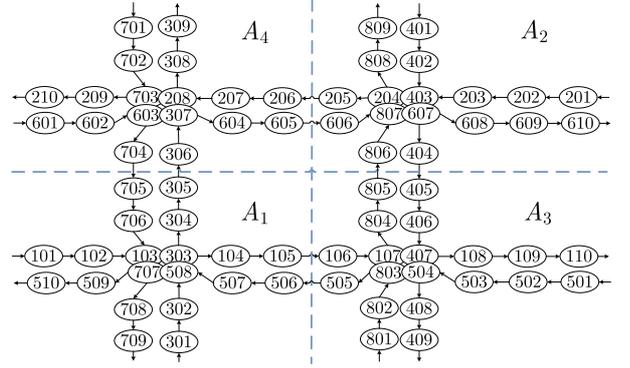


Fig. 2. CTM for a 2-by-2 intersection topology. The dashed lines indicate the *spatial decomposition* of the network into four areas, each one controlled by the corresponding intersection controller.

$\mathcal{B}_1^O = \{105, 305\}$, $\mathcal{B}_1 = \{705, 506, 105, 305\}$, $\mathcal{B}_1^{IE} = \{705, , 306, 506, 106\}$, $\mathcal{B}_1^{OE} = \{105, 305, , 505, 704\}$ and $\mathcal{B}_1^E = \{705, , 306, 506, 106, 105, 305, , 505, 704\}$.

The defined boundary cells are essential in spatially decomposing a given transportation network for the CTM model, and can be applied for the decomposition of several transportation problems such as the freeway ramp metering and the dynamic traffic assignment problems. Here, this spatial decomposition is utilised for the distributed solution of *MITSC* via ADMM. It should be emphasized that an arbitrary number of intersections could be considered for each area, given that a controller can manage the entire area, without affecting the final derived solution. That is because each area does not decide independently, but a global decision is reached through collaboration, contrary to existing systems, such as [17] and [9], where subsystems are designed to be jointly optimized in a central manner and share common characteristics (e.g. common cycle length and interrelated phase splits). Nevertheless, spatial decomposition controls the amount of communication exchanged and the computation required from each area controller. The choice of decomposing the system in single-intersection areas is to allow a fully distributed architecture. Also, it should be emphasized that such a problem need not necessarily involve all intersections of a city but rather parts of the city where traffic is interrelated [38].

V. DISTRIBUTED FORMULATION AND SOLUTION OF MITSC

The centralized solution of the MITSC problem may provide better performance if it can be derived, but it has several shortcomings as a solution strategy. Firstly, solving this problem is usually intractable and hence not suitable for online decision making because the problem is complex (NP-hard) and of large-scale (especially when the problem involves a large time horizon and several intersections). Secondly, centralized solutions require global information about the status of the network and hence may be prone to communication related failures. Thirdly, solving the problem centrally is not robust, as failure of the central unit will result in complete failure of the system. On the other hand, distributed strategies can be more robust to failures. For example, communication failure

between certain intersections may result in partial coordination loss, but intersections can continue working either individually or in subgroups. Additionally, by temporally and spatially decomposing the problem into subproblems of smaller size, although global optimality is not maintained, computationally efficient and good quality solutions can be obtained.

A distributed solution to the *MITSC* problem is obtained by performing two consecutive phases. In *phase 1*, the relaxed problem, *RMITSC*, is solved distributedly via ADMM so as to avoid the exponential complexity of *MITSC*. This is achieved by transforming *RMITSC* into an appropriate ADMM form and providing close-form solutions to the optimization problems arising in the three iterative ADMM steps, as explained in section V-A; a distributed algorithm summarizing the ADMM steps and explaining the information neighboring subsystems is also provided (Algorithm 1). In *phase 2*, a distributed rounding scheme is used to obtain a close to optimal solution to *MITSC* by rounding-off fractional values of decision variables obtained from the solution of *RMITSC* in phase 1. Section V-B describes two novel distributed rounding schemes based on minimizing the cumulative departure rate error from the relaxed solution (section V-B1) and the roundoff error of the fractional decision variables (section V-B2).

A. Phase 1: Distributed Solution of *RMITSC*

To derive a distributed formulation for *RMITSC*, we need to convert the problem into a form that is decomposable for each intersection. This implies that each IC should only solve subproblems associated with variables belonging to its area. Towards this direction, we rewrite the centralised *RMITSC* problem into the following form that distinguishes decoupled constraints for each intersection (Eqs. (18b), (18c) and (18f)), from coupled constraints that involve variables from more than one intersections (Eqs. (18d) and (18e)):

$$\min \sum_{i=1}^{N_I} \mathbf{f}_i^T \mathbf{x}_i \quad (18a)$$

$$\text{s.t. } \mathbf{A}_i \mathbf{x}_i = \mathbf{a}_i, \quad i \in \mathcal{R}, \quad (18b)$$

$$\mathbf{B}_i \mathbf{x}_i \leq \mathbf{b}_i, \quad i \in \mathcal{R}, \quad (18c)$$

$$y_{c,t} + s_{c,t} = WN - Wn_{c+1,t}, \quad c \in \mathcal{B}_i^O, t \in \mathcal{T}, i \in \mathcal{R}, \quad (18d)$$

$$n_{c,t+1} = n_{c,t} + y_{c-1,t} - y_{c,t}, \quad c \in \mathcal{B}_i^I, t \in \mathcal{T}, i \in \mathcal{R}, \quad (18e)$$

$$\mathbf{x}_i^l \leq \mathbf{x}_i \leq \mathbf{x}_i^u, \quad i \in \mathcal{R}, \quad (18f)$$

where $\mathbf{x}_i = [\mathbf{n}_i^T, \mathbf{y}_i^T, \mathbf{w}_i^T, \mathbf{u}_i^T, \mathbf{s}_i^T]^T$ denotes the vector of variables of area A_i , $\mathbf{n}_i \in \mathbb{R}^{|\mathcal{A}_i|(T_o+1) \times 1}$, $\mathbf{y}_i \in \mathbb{R}^{|\mathcal{A}_i|T_o \times 1}$, $\mathbf{w}_i \in \mathbb{R}^{T_o \times 1}$, $\mathbf{u}_i \in \mathbb{R}^{(T_o-1) \times 1}$ are vectorized versions of variables $n_{c,t}$, $y_{c,t}$, $w_{i,t}$ and $u_{i,t}$, $c \in \mathcal{A}_i$, $i \in \mathcal{R}$, $t \in \mathcal{T}$, \mathbf{x}_i^l and \mathbf{x}_i^u are the lower and upper bounds of the corresponding variables, while dummy variables $\mathbf{s}_i \in \mathbb{R}^{|\mathcal{B}_i^O|T_o \times 1}$ are introduced to convert inequalities (8d) for output boundary cells of area A_i into equalities². Expression (18a) captures the linear objective function, where vector \mathbf{f}_i denotes the cost associated with variables \mathbf{x}_i , while Eqs. (18b) and (18c) capture all decoupled equality and inequality constraints whose variables belong to the same area i.e. Eqs. (8b), (8c), (10)-(17), (8d) excluding

output boundary cells and (8e) excluding input boundary cells. The excluded constraints are represented by the coupled equalities (18d) and (18e) which involve variables from two different areas and hence cannot be handled directly by the same IC. Note that the flow constraint $\sum_{c \in \mathcal{A}} n_{c,T_o+1} = 0$, as well as any initialization constraints have been included in the above formulation as bound constraints; for example the flow constraint can be written as: $n_{c,T_o+1} = 0$ or $0 \leq n_{c,T_o+1} \leq 0$, $c \in \mathcal{A}$.

To be able to solve the problem in a distributed manner via the ADMM algorithm, we rewrite it into the equivalent form:

$$\min \sum_{i=1}^{N_I} \mathbf{f}_i^T \mathbf{x}_i \quad (19a)$$

$$\text{s.t. } \mathbf{A}_i \mathbf{x}_i = \mathbf{a}_i, \quad i \in \mathcal{R}, \quad (19b)$$

$$\mathbf{B}_i \mathbf{x}_i + \boldsymbol{\zeta}_i = \mathbf{b}_i, \quad i \in \mathcal{R}, \quad (19c)$$

$$y_{c,t} + s_{c,t} = WN - W\xi_{n_{c+1,t}}, \quad c \in \mathcal{B}_i^O, t \in \mathcal{T}, i \in \mathcal{R}, \quad (19d)$$

$$n_{c,t+1} = n_{c,t} + \xi_{y_{c-1,t}} - y_{c,t}, \quad c \in \mathcal{B}_i^I, t \in \mathcal{T}, i \in \mathcal{R}, \quad (19e)$$

$$\boldsymbol{\xi}_i = \mathbf{x}_i, \quad i \in \mathcal{R}, \quad (19f)$$

$$\mathbf{x}_i^l \leq \boldsymbol{\xi}_i \leq \mathbf{x}_i^u, \quad \boldsymbol{\zeta}_i \geq 0, \quad i \in \mathcal{R}, \quad (19g)$$

where the equivalence is apparent from the fact that $\boldsymbol{\xi}_i = \mathbf{x}_i$, $i \in \mathcal{R}$, while variables $\boldsymbol{\zeta}_i$ are introduced to transform inequalities (18c) into equalities (19c). For better clarity, we abuse the notation and imply that variables $\xi_{n_{c,t}}$ indicate correspondence to variables $n_{c,t}$; the same applies to the other variable sets.

The above formulation adheres to ADMM formulation (4) with $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_{N_I}^T]^T$ and $\mathbf{z} = [\boldsymbol{\zeta}_1^T, \dots, \boldsymbol{\zeta}_{N_I}^T, \boldsymbol{\xi}_1^T, \dots, \boldsymbol{\xi}_{N_I}^T]^T$, while \mathcal{C}_x and \mathcal{C}_z are given by:

$$\mathcal{C}_x = \{\mathbf{x} | \mathbf{A}_i \mathbf{x}_i = \mathbf{a}_i, \quad i \in \mathcal{R}\},$$

$$\mathcal{C}_z = \{\mathbf{z} | \mathbf{z} = \{\boldsymbol{\xi}, \boldsymbol{\zeta}\}, \mathbf{x}^l \leq \boldsymbol{\xi} \leq \mathbf{x}^u, \boldsymbol{\zeta} \geq 0\}.$$

Constraints (19c)-(19f) are coupling constraints between \mathbf{x} and \mathbf{z} , representing eq. (4b) of ADMM formulation.

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

$$\begin{aligned} \min \mathcal{L}(\mathbf{x}, \boldsymbol{\zeta}, \boldsymbol{\xi}, \boldsymbol{\kappa}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}) = & \\ & \sum_{i=1}^{N_I} \left(\mathbf{f}_i^T \mathbf{x}_i + h_{\mathcal{C}_{x_i}}(\mathbf{x}_i) + h_{\mathcal{C}_{\xi_i}}(\boldsymbol{\xi}_i) + h_{\mathcal{C}_{\zeta_i}}(\boldsymbol{\zeta}_i) \right) \\ & + \frac{\rho}{2} \sum_{i=1}^{N_I} \left(\|\mathbf{B}_i \mathbf{x}_i + \boldsymbol{\zeta}_i - \mathbf{b}_i + \boldsymbol{\kappa}_i\|_2^2 + \|\mathbf{x}_i - \boldsymbol{\xi}_i + \boldsymbol{\nu}_i\|_2^2 \right. \\ & + \sum_{c \in \mathcal{B}_i^O} \sum_{t=1}^{T_o} (y_{c,t} + s_{c,t} + W\xi_{n_{c+1,t}} - WN + \lambda_{c,t})^2 \\ & \left. + \sum_{c \in \mathcal{B}_i^I} \sum_{t=1}^{T_o} (n_{c,t+1} - n_{c,t} + y_{c-1,t} - \xi_{y_{c-1,t}} + \mu_{c,t})^2 \right), \end{aligned}$$

where $\boldsymbol{\kappa}$, $\boldsymbol{\nu}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ are the scaled Lagrange multipliers of constraints (19c), (19f), (19d), and (19e), respectively. Hence, ADMM consists of iteratively solving $\mathcal{L}(\mathbf{x}, \boldsymbol{\zeta}, \boldsymbol{\xi}, \boldsymbol{\kappa}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu})$ over \mathbf{x} (step 1), then $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ in parallel (step 2), and then updating $\boldsymbol{\kappa}$, $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ (step 3). After the k th iteration of the ADMM algorithm, vectors \mathbf{x}^k , $\boldsymbol{\zeta}^k$, $\boldsymbol{\xi}^k$, $\boldsymbol{\kappa}^k$, $\boldsymbol{\lambda}^k$, $\boldsymbol{\mu}^k$, $\boldsymbol{\nu}^k$ have

²Area-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 A_i respectively.

been attained; next, we show how the problems that appear in iteration $k + 1$ can be solved.

Step 1 of ADMM consists of solving the problem over \mathbf{x} . The problem to be solved can be defined as:

$$\begin{aligned} \mathbf{x}^{k+1} = \arg \min \mathcal{L}(\mathbf{x}, \boldsymbol{\zeta}^k, \boldsymbol{\xi}^k, \boldsymbol{\kappa}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \boldsymbol{\nu}^k) = & \\ \sum_{i=1}^{N_I} \left\{ \mathbf{f}_i^T \mathbf{x}_i + hc_{x_i}(\mathbf{x}_i) \right. & \\ + \frac{\rho}{2} \left(\|\mathbf{B}_i \mathbf{x}_i + \boldsymbol{\zeta}_i^k - \mathbf{b}_i + \boldsymbol{\kappa}_i^k\|_2^2 + \|\mathbf{x}_i - \boldsymbol{\xi}_i^k + \boldsymbol{\nu}_i^k\|_2^2 \right) & \\ + \sum_{c \in \mathcal{B}_i^O} \sum_{t=1}^{T_o} \left(y_{c,t} + s_{c,t} + W \xi_{n_{c+1,t}}^k - WN + \lambda_{c,t}^k \right)^2 & \\ \left. + \sum_{c \in \mathcal{B}_i^I} \sum_{t=1}^{T_o} \left(n_{c,t+1} - n_{c,t} + y_{c,t} - \xi_{y_{c-1,t}}^k + \mu_{c,t}^k \right)^2 \right\} & \end{aligned}$$

The above formulation is decomposable for each \mathbf{x}_i , meaning that we can solve N_I smaller subproblems, one for each intersection. The only limitation is that the values of $\xi_{n_{c+1,t}}^k$ and $\xi_{y_{c-1,t}}^k$ that appear in the last two terms are not known to IC_i . However, these quantities can easily be communicated to IC_i as they belong to immediate neighboring intersections. Hence, assuming that these quantities are known to the appropriate intersections, \mathbf{x}^{k+1} can be computed by solving N_I problems in parallel, one at each intersection. To simplify the notation, let us define the following:

$$\mathbf{e}_{i1}^k = -\boldsymbol{\zeta}_i^k + \mathbf{b}_i - \boldsymbol{\kappa}_i^k, \quad (20)$$

$$\mathbf{e}_{i4}^k = \boldsymbol{\xi}_i^k - \boldsymbol{\nu}_i^k, \quad (21)$$

$$\sum_{c \in \mathcal{B}_i^O} \sum_{t=1}^{T_o} (y_{c,t} + s_{c,t} + W \xi_{n_{c+1,t}}^k - WN + \lambda_{c,t}^k)^2 = \|\mathbf{C}_i \mathbf{x}_i - \mathbf{e}_{i2}^k\|_2^2, \quad (22)$$

$$\sum_{c \in \mathcal{B}_i^I} \sum_{t=1}^{T_o} (n_{c,t+1} - n_{c,t} + y_{c,t} - \xi_{y_{c-1,t}}^k + \mu_{c,t}^k)^2 = \|\mathbf{H}_i \mathbf{x}_i - \mathbf{e}_{i3}^k\|_2^2. \quad (23)$$

Based on the above definitions, IC_i needs to solve the problem:

$$\min_{\mathbf{x}_i} \frac{1}{2} \mathbf{x}_i^T \mathbf{F}_i \mathbf{x}_i - \mathbf{x}_i^T \mathbf{p}_i^k \quad (24)$$

$$\mathbf{A}_i \mathbf{x}_i = \mathbf{a}_i,$$

where $\mathbf{F}_i = \mathbf{B}_i^T \mathbf{B}_i + \mathbf{C}_i^T \mathbf{C}_i + \mathbf{H}_i^T \mathbf{H}_i + \mathbf{I}_i$, with \mathbf{I}_i being the identity matrix of appropriate dimensions, and $\mathbf{p}_i^k = \mathbf{B}_i^T \mathbf{e}_{i1}^k + \mathbf{C}_i^T \mathbf{e}_{i2}^k + \mathbf{D}_i^T \mathbf{e}_{i3}^k + \mathbf{e}_{i4}^k - (1/\rho) \mathbf{f}_i$. Problem (24) is a convex quadratic programming problem with linear equality constraints whose solution can be obtained by solving the linear system of equations resulting from the KKT conditions (see ch. 16 of [39]):

$$\begin{bmatrix} \mathbf{F}_i & \mathbf{A}_i^T \\ \mathbf{A}_i & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i^{k+1} \\ \boldsymbol{\sigma} \end{bmatrix} = \mathbf{K}_i \begin{bmatrix} \mathbf{x}_i^{k+1} \\ \boldsymbol{\sigma} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_i^k \\ \mathbf{a}_i \end{bmatrix}, \quad (25)$$

where $\boldsymbol{\sigma}$ are dual variables associated with the equality constraints, and $\mathbf{K}_i \in \mathbb{R}^{(M_{x,i} + M_{A_{eq,i}}) \times (M_{x,i} + M_{A_{eq,i}})}$, with $M_{x,i} = (2|\mathcal{A}_i| + |\mathcal{B}_i^O| + 2)T_o + |\mathcal{A}_i| - 1$ and $M_{A_{eq,i}} = (|\mathcal{A}_i| - |\mathcal{B}_i^O| + |\mathcal{D}_i|)T_o$, being the number of rows in \mathbf{x}_i and \mathbf{A}_i , respectively.

Eq. (25) needs to be solved several times at IC_i , one per iteration, so it is important to reduce the computational

complexity of solving this system. Because matrix \mathbf{K}_i is constant for different iterations, we can decompose and store it once during initialization; in this way the solution of (25) can be computed an order of magnitude cheaper. An efficient and robust decomposition method suitable for sparse matrices is LDL^T [40]. Using this method, we can derive matrices $\mathbf{L}_i, \mathbf{D}_i, \mathbf{P}_i$ such that:

$$\mathbf{K}_i = \mathbf{P}_i \mathbf{L}_i \mathbf{D}_i \mathbf{L}_i^T \mathbf{P}_i^T,$$

where \mathbf{L}_i is lower triangular, \mathbf{D}_i is block diagonal with blocks of dimension 1 or 2, and \mathbf{P}_i is an orthogonal permutation matrix chosen according to the sparsity pattern of \mathbf{K}_i . The solution of (25) is then:

$$\begin{bmatrix} \mathbf{x}_i^{k+1} \\ \boldsymbol{\sigma} \end{bmatrix} = \mathbf{P}_i \left((\mathbf{L}_i^T)^{-1} \left(\mathbf{D}_i^{-1} \left(\mathbf{L}_i^{-1} \left(\mathbf{P}_i^T \begin{bmatrix} \mathbf{p}_i^k \\ \mathbf{a}_i \end{bmatrix} \right) \right) \right) \right) \quad (26)$$

To compute eq. (26) multiple times, we need to store $\mathbf{L}_i, \mathbf{D}_i^{-1}$ and \mathbf{P}_i . The computation of \mathbf{D}_i^{-1} can be performed during initialization and only requires the inversion of matrices of dimension 1 or 2, while the computation of terms $\mathbf{L}_i^{-1} \boldsymbol{\chi}_2$ and $(\mathbf{L}_i^T)^{-1} \boldsymbol{\chi}_1$ can be performed in each iteration using forward and backward substitution, as \mathbf{L}_i and \mathbf{L}_i^T are lower and upper triangular respectively.

Step 2 of ADMM consists of solving the problem over $\boldsymbol{\zeta}$ and $\boldsymbol{\xi}$. The problem to be solved can be defined as:

$$\begin{aligned} [\boldsymbol{\zeta}^{k+1}; \boldsymbol{\xi}^{k+1}] = \arg \min \mathcal{L}(\mathbf{x}^{k+1}, \boldsymbol{\zeta}, \boldsymbol{\xi}, \boldsymbol{\kappa}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \boldsymbol{\nu}^k) = & \\ \sum_{i=1}^{N_I} \left(hc_{\boldsymbol{\zeta}_i}(\boldsymbol{\zeta}_i) + hc_{\boldsymbol{\xi}_i}(\boldsymbol{\xi}_i) + \right. & \\ \frac{\rho}{2} \left(\|\mathbf{B}_i \mathbf{x}_i^{k+1} + \boldsymbol{\zeta}_i - \mathbf{b}_i + \boldsymbol{\kappa}_i^k\|_2^2 + \|\mathbf{x}_i^{k+1} - \boldsymbol{\xi}_i + \boldsymbol{\nu}_i^k\|_2^2 + \right. & \\ \sum_{c \in \mathcal{B}_i^I} \sum_{t=1}^{T_o} \left(y_{c-1,t}^{k+1} + s_{c-1,t}^{k+1} + W \xi_{n_{c,t}}^{k+1} - WN + \lambda_{c-1,t}^k \right)^2 + & \\ \left. \left. \sum_{c \in \mathcal{B}_i^O} \sum_{t=1}^{T_o} \left(n_{c+1,t+1}^{k+1} - n_{c+1,t}^{k+1} + y_{c+1,t}^{k+1} - \xi_{y_{c,t}}^k + \mu_{c+1,t}^k \right)^2 \right) \right) & \end{aligned}$$

The above formulation is separable for variables $\boldsymbol{\zeta}_i$ and $\boldsymbol{\xi}_i$. Similar to step 1, the only limitation is that $y_{c-1,t}^{k+1}, s_{c-1,t}^{k+1}, \lambda_{c-1,t}^k, c \in \mathcal{B}_i^I$ and $y_{c+1,t}^{k+1}, n_{c+1,t}^{k+1}, \mu_{c+1,t}^k, c \in \mathcal{B}_i^O, t = 1, \dots, T_o$, are not known to IC_i and need to be communicated to IC_i from adjacent intersections. Hence, assuming that these quantities are known to appropriate intersections, $\boldsymbol{\zeta}^{k+1}$ and $\boldsymbol{\xi}^{k+1}$ can be computed analytically by solving N_I problems in parallel, one at each intersection. The following analytical solutions can easily be obtained:

$$\boldsymbol{\zeta}_i^{k+1} = \min \{ \mathbf{0}, -(\mathbf{B}_i \mathbf{x}_i^{k+1} - \mathbf{b}_i + \boldsymbol{\kappa}_i^k) \} \quad (27)$$

$$\boldsymbol{\xi}_i^{k+1} = \min \{ \mathbf{x}_i^u, \max \{ \mathbf{x}_i^l, \mathbf{x}_i^{k+1} + \boldsymbol{\nu}_i^k \} \}, \quad (28)$$

Special attention should be given to the fact that eq. (28) applies to all $\boldsymbol{\xi}_i$ variables, except from $\xi_{n_{c,t}}, c \in \mathcal{B}_i^I, t \in \mathcal{T}$ and $\xi_{y_{c,t}}, c \in \mathcal{B}_i^O, t \in \mathcal{T}$. For the former variables we need to solve the problem:

$$\begin{aligned} \min \left(y_{c-1,t}^{k+1} + s_{c-1,t}^{k+1} + W \xi_{n_{c,t}}^{k+1} - WN + \lambda_{c-1,t}^k \right)^2 + & \\ \left(n_{c,t}^{k+1} - \xi_{n_{c,t}}^{k+1} + \nu_{n_{c,t}}^k \right)^2 & \\ \text{s.t. } x_{n_{c,t}}^l \leq \xi_{n_{c,t}} \leq x_{n_{c,t}}^u & \end{aligned} \quad (29)$$

Problem (29) can be solved analytically by finding the value of $\xi_{n_{c,t}}$ that zeroes the derivative of the objective function and projecting onto the feasible set, which yields:

$$\xi_{n_{c,t}}^{k+1} = \min \left\{ x_{n_{c,t}}^u, \max \left\{ x_{n_{c,t}}^l, (1+W^2)^{-1} \left(n_{c,t}^{k+1} + \nu_{n_{c,t}}^k - W(y_{c-1,t}^{k+1} + s_{c-1,t}^{k+1} - WN + \lambda_{c-1,t}^k) \right) \right\} \right\}. \quad (30)$$

For variables $\xi_{y_{c,t}}$, $c \in \mathcal{B}_i^O$, $t \in \mathcal{T}$ we need to solve:

$$\begin{aligned} \min \quad & (n_{c+1,t+1}^{k+1} - n_{c+1,t}^{k+1} + y_{c+1,t}^{k+1} - \xi_{y_{c,t}} + \mu_{c+1,t}^k)^2 + \\ & (y_{c,t}^{k+1} - \xi_{y_{c,t}} + \nu_{y_{c,t}}^k)^2 \\ \text{s.t.} \quad & x_{y_{c,t}}^l \leq \xi_{y_{c,t}} \leq x_{y_{c,t}}^u \end{aligned} \quad (31)$$

whose solution is given by:

$$\xi_{y_{c,t}}^{k+1} = \min \left\{ x_{n_{c,t}}^u, \max \left\{ x_{n_{c,t}}^l, \frac{1}{2} (n_{c+1,t+1}^{k+1} - n_{c+1,t}^{k+1} + y_{c+1,t}^{k+1} + \mu_{c+1,t}^k + y_{c,t}^{k+1} + \nu_{y_{c,t}}^k) \right\} \right\}. \quad (32)$$

Step 3 of ADMM consists of updating the dual variables associated with the coupled equality constraints. IC_i needs to perform the following updates, for $t \in \mathcal{T}$:

$$\kappa_i^{k+1} = \mathbf{B}_i \mathbf{x}_i^{k+1} + \zeta_i^{k+1} - \mathbf{b}_i + \kappa_i^k \quad (33)$$

$$\lambda_{c,t}^{k+1} = y_{c,t}^{k+1} + s_{c,t}^{k+1} + W \xi_{n_{c+1,t}}^{k+1} - WN + \lambda_{c,t}^k, \quad c \in \mathcal{B}_i^O, \quad (34)$$

$$\mu_{c,t}^{k+1} = n_{c,t+1}^{k+1} - n_{c,t}^{k+1} + y_{c,t}^{k+1} - \xi_{y_{c-1,t}}^{k+1} + \mu_{c,t}^k, \quad c \in \mathcal{B}_i^I, \quad (35)$$

$$\nu_i^{k+1} = \mathbf{x}_i^{k+1} - \xi_i^{k+1} + \nu_i^k. \quad (36)$$

The computation of updates (33) and (36) are based only on local information, while updates (34) and (35) require knowledge of terms $\xi_{n_{c+1,t}}^{k+1}$, $c \in \mathcal{B}_i^O$ and $\xi_{y_{c-1,t}}^{k+1}$, $c \in \mathcal{B}_i^I$, $t \in \mathcal{T}$ which can be obtained from neighboring intersections.

From the above analysis, it is clear that in each iteration of ADMM we need to perform three computation steps followed by three local communication steps. Communication step 1 can be eliminated as values $\xi_{n_{c+1,t}}^{k+1}$, $c \in \mathcal{B}_i^O$ and $\xi_{y_{c-1,t}}^{k+1}$, $c \in \mathcal{B}_i^I$, $t \in \mathcal{T}$ 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 ξ -values it needs for computation step 3. In our case, computing the necessary ξ -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 ADMM for the solution of \mathcal{RMITS} . Although, Algorithm 1 is iterative, it is important to highlight the fact that all three steps of each iteration are *analytically* computed according to Eqs. (26), (28)-(32), and (33)-(36) respectively. Obtaining distributed analytical solutions to all steps of the ADMM algorithm is not a standard result, but emanates from the structure of the problem and the targeted introduction of variables ξ and ζ into formulation (19).

Algorithm 1 : Distributed ADMM algorithm for the solution of \mathcal{RMITS} problem

for (each intersection controller IC_i) **do**

Init.: Set $\zeta_i^0 = \xi_i^0 = \kappa_i^0 = \lambda_i^0 = \mu_i^0 = \nu_i^0 = \mathbf{0}$.

repeat

1. Compute \mathbf{x}_i^{k+1} using eq. (26)(Step 1).

2. Communicate to neighboring intersection $j \in \mathcal{N}_i$ the values of $y_{c,t}^{k+1}$, $s_{c,t}^{k+1}$, $n_{c,t}^{k+1}$, $c \in \{\mathcal{B}_i \cap \mathcal{B}_j^E\}$, $t \in \mathcal{T}$.

3. Compute ζ_i^{k+1} using eq. (27) and ξ_i^{k+1} using Eqs. (28), (30) and (32). Additionally, compute values for $\xi_{n_{c,t}}$ and $\xi_{y_{c,t}}$, $c \in \{\mathcal{B}_i^E \setminus \mathcal{B}_i\}$, $t \in \mathcal{T}$ using Eqs. (30) and (32) (Step 2).

4. Update the values of dual variables κ_i^{k+1} , λ_i^{k+1} , μ_i^{k+1} , ν_i^{k+1} using eqs. (33)-(36). Additionally update $\lambda_{c,t}$ and $\mu_{c,t}$, $c \in \{\mathcal{B}_i^E \setminus \mathcal{B}_i\}$, $t \in \mathcal{T}$ using Eqs. (34) and (35) (Step 3).

5. Compute primal and dual residuals (eqs. (5), (6)) and check stopping conditions.

until (Convergence achieved)

end for

B. Phase 2: Distributed rounding

Having obtained a distributed solution to \mathcal{RMITS} via ADMM, the next step is to round the main decision variables ($w_{i,t}$, $i \in \mathcal{R}$, $t = 1, \dots, T_w$) so as to obtain a close to optimal solution to \mathcal{MITS} . It is important to note that rounding should be performed in a distributed manner in accordance to the variables that are known to each IC, and also that the decision variables need to be rounded-off only for the time-window T_w and not for the whole optimization period T_o . In the next sections we describe two different distributed rounding schemes.

1) *Distributed Cumulative Departure Rate Rounding (DC-DRR) Scheme:* In the cumulative departure rate (CDR) method, originally proposed as a heuristic for rounding solutions in a centralized setting [14], the main idea is to attempt to match the cumulative departure rate (CDR), resulting from \mathcal{RMITS} with the CDR obtained from binary decisions for each movement of an intersection. Considering two phase intersections of one way arterials (e.g. Fig. 2), the CDR of phase p of intersection i at time τ is defined as:

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

where $\mathcal{D}_{p,i}$ denotes the set of cells immediately downstream all movements of phase p of intersection i .

DCDRR is outlined in Algorithm 2. Note that DCDRR 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 $n_{c,t}$ respectively.

2) *Distributed Decision Variable Rounding (DDVR) Scheme:* The solution of the relaxed problem provides fractional decision variables $w_{i,t}$ which indicate the optimal portion of green or red light for each time unit and phase. One way to roundoff the decision variables is to minimize

Algorithm 2 : DCDRR Scheme

for (each intersection controller $IC_i, i \in \mathcal{R}$) **do**
Init.: Compute $\bar{C}_{p,t,i}, \forall t$, of \mathcal{RMITS} and set CDR of binary decision to: $C_{p,0,i} = 0, p = \{1, 2\}$.
for ($\tau = 1, \dots, T_w$) **do**
 Communicate $n_{c,\tau}, c \in \mathcal{B}_i^I$ to IC_j where $c-1 \in \mathcal{B}_j^O$.
 Wait for reception of $n_{c+1,\tau}, c \in \mathcal{B}_i^O$ from other ICs.
if (Constraints (10) - (14) are not violated) **then**
 For $w_{i,\tau} = k, k \in \{0, 1\}$ compute $y_{c_1,t}^k$ and $y_{c_2,t}^k, c_1 \in \mathcal{I}_{1,i}$ and $c_2 \in \mathcal{I}_{2,i}$, based on Eq. (1).
 Set $C_{p,\tau,i}^k = C_{p,\tau-1,i} + \sum_{c \in \mathcal{D}_{p,i}} y_{c,t}^k, p = \{1, 2\}$.
 Set $w_{i,\tau} = \arg \min_{k \in \{0,1\}} \left\{ \sum_{p=1}^2 |C_{p,\tau,i}^k - \bar{C}_{p,\tau,i}| \right\}$.
else
 Set $w_{i,\tau}$ to a value that violates no constraint.
end if
 Update $y_{c,\tau}, c \in \mathcal{A}_i$ and $C_{p,\tau,i}$ based on $w_{i,\tau}$.
 Communicate $y_{c,\tau}, c \in \mathcal{B}_i^O$ to IC_j where $c+1 \in \mathcal{B}_j^I$.
 Wait for reception of $y_{c-1,\tau}, c \in \mathcal{B}_i^I$ from other ICs.
 Update $n_{c,\tau+1}, c \in \mathcal{A}_i$ using Eq. (3).
end for
end for

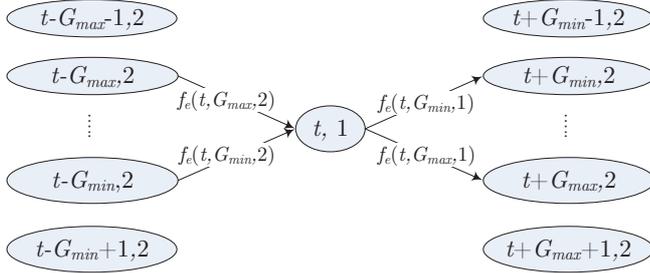


Fig. 3. Transition into and out of vertex $(t, 1)$ of the constructed graph for the decision variable rounding scheme for a two phase intersection.

the roundoff error subject to the satisfaction of the minimum and maximum green time constraints. In mathematical programming terms this problem can be expressed as:

$$\begin{aligned} \min \quad & \sum_{i \in \mathcal{R}} \sum_{t=1}^{T_o} w_{i,t}^B f_{1,i,t} + (1 - w_{i,t}^B) f_{2,i,t} \quad (37) \\ \text{s.t.} \quad & - \text{traffic light constraints (10)-(14),} \\ & - \text{initial traffic light conditions (17),} \\ & w_{i,t}^B \in \{0, 1\}, i \in \mathcal{R}, t \in \mathcal{T}, \end{aligned}$$

where $f_{1,i,t} = |1 - w_{i,t}|$ and $f_{2,i,t} = |0 - w_{i,t}|$, are the costs of having a green light at intersection i and time t for phases 1 and 2 respectively. Note that the problem is separable for different intersections so that the problem is equivalent to N_I independent problems, each involving the decision variables of one intersection. Despite the problem being MILP and combinatorial in nature, we have developed an optimal solution procedure of problem (37), which is based on building an appropriate graph and solving a shortest path problem using Dijkstra's algorithm for each intersection.

For a two phase intersection, the graph is comprised of vertices $(t, p), t \in \mathcal{T}, p \in \{1, 2\}$ 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 costs indicate the cumulative cost of making the particular transition. Fig. 3 shows the transitions into and out of state $(t, 1)$ for an arbitrary area A_i . As can be seen, this state can be reached if the green light of phase 2 started at $t' \in \{t - G_{max}, \dots, t - G_{min}\}$ and the duration of the green is of length $t - t'$. In this case the transition cost is $f_e(t, t - t', 2) = \sum_{\tau=t'}^{t-1} f_{2,i,\tau}$ which indicates the cumulative cost resulting from the deviation between the continuous and binary values of the decision variables. Similarly, edges out of state $(t, 1)$ and into state $(t', 2)$ can be defined with costs $f_e(t, t' - t, 1) = \sum_{\tau=t}^{t'-1} f_{1,i,\tau}, t' \in \{t + G_{min}, \dots, t + G_{max}\}$. 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, for the considered intersection, c . Therefore, the rounding cost for time units 10-14 is equal to $f_e(10, 5, 1) = \sum_{t=10}^{14} f_{1,i,t} = \sum_{t=10}^{14} |1 - w_{i,t}|$. By properly building the graph of traffic light states and phase transitions, the minimal rounding cost of problem (37) is obtained by finding the path of 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 results in the same decision with problem (37), at a fraction of the execution time (2-3 orders of magnitude faster), as solving a shortest path problem is of low polynomial complexity. Another attractive characteristic of DDVR is that it can be obtained independently for different intersections, resulting in a completely decentralized rounding scheme with no communication requirements.

C. Computational and communication complexity

The main computational step of both DCDRR and DDVR is the first step of each iteration of Algorithm 1 that requires the solution of the linear equations system (25), which is of computational complexity $\mathcal{O}((M_{x,i} + M_{A_{eq},i})^3)$. Steps 2 and 3 of Algorithm 1 and the distributed rounding schemes have negligible cost compared to step 1. As already mentioned, because matrix \mathbf{K}_i appearing in (25) is the same for all iterations, by computing offline and caching \mathbf{K}_i 's LDL decomposition, which is of computational complexity $\mathcal{O}((M_{x,i} + M_{A_{eq},i})^3)$, the computational complexity of solving (25) is reduced to $\mathcal{O}((M_{x,i} + M_{A_{eq},i})^2)$ for dense matrices. Note that this complexity implies that the solution of (25) would scale quadratically to the number of cells $|A_i|$ and the optimization horizon T_o , if \mathbf{K}_i was dense. Nonetheless, LDL decomposition exploits the sparsity of \mathbf{K}_i to reduce the computational complexity even further; in Section VI-C, it is shown that each iteration of Algorithm 1, approximately scales to the power of 1.25 for both T_o and $|A_i|$, which implies that the practical computational complexity of DCDRR and DDVR

is $\mathcal{O}(M_I(|\mathcal{A}_i|T_o)^{1.25})$, where M_I is the number of ADMM iterations.

Regarding the communication complexity of the proposed approaches, in the k th iteration of ADMM algorithm, IC_i , $i \in \mathcal{R}$ has to communicate the newly computed values of $y_{c,t}, s_{c,t}, n_{c,t}$, $t \in \mathcal{T}$ of the cells on the boundary between two adjacent areas. Hence, the iteration communication cost of IC_i is $\mathcal{O}(3|\mathcal{B}_i|T_o)$ and the total communication cost of ADMM is $\mathcal{O}(3M_I T_o \sum_{i \in \mathcal{R}} |\mathcal{B}_i|)$. DCDRR involves two communication steps for IC_i with cost $\mathcal{O}(|\mathcal{B}_i|)$ for each of the T_w iterations, while DDVR requires no communication. Hence the total communication cost of DCDRR and DDVR is $\mathcal{O}((3M_I T_o + T_w) \sum_{i \in \mathcal{R}} |\mathcal{B}_i|)$ and $\mathcal{O}(3M_I T_o \sum_{i \in \mathcal{R}} |\mathcal{B}_i|)$, respectively.

VI. SIMULATION RESULTS

The effectiveness of the proposed distributed strategies was evaluated for a four-intersection base topology of two-way, two-lane alternating direction arterials as shown in Fig. 2. 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 time are assumed to be equal to 15s and 45s respectively, i.e. $G_{min} = 3$ and $G_{max} = 8$. The time horizon considered is $T_h = 1440$ time units (2h), with traffic being generated for 360 time units (0.5h). The time horizon is temporally decomposed into 10min intervals ($T_w = 120$). In the evaluation, we consider three traffic scenarios corresponding to *low* (10% congestion level³), *moderate* (50% congestion level), and *high* congestion (90% congestion level) randomly generated incoming traffic. Next we examine the effectiveness of the proposed distributed strategies in terms of convergence, optimality and scalability. In the ADMM algorithm, we set $\rho = N_I$, while for faster convergence we consider an over-relaxation parameter $a = 1.6$ (see [27]). The centralized optimal solution to all LP and MILP problems considered was obtained using Gurobi [41], while close to optimal problem solutions via genetic algorithms (GA) were obtained using the global optimization toolbox of Matlab using both random (GARI) and deterministic initialization (GADI) from the obtained DCDRR and DDVR solutions. All 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.

A. Convergence Properties

Fig. 4 illustrates the convergence properties of the distributed ADMM algorithm for the base topology when the moderate traffic scenario is employed when considering one time-window for optimization. The left y-axis shows the relative deviation of the objective value of $\mathcal{R}MITSC$ obtained using the distributed ADMM algorithm compared to the

³*Congestion level* is defined as the ratio of the average to the maximum arrival rate over the considered traffic generation period at all traffic sources.

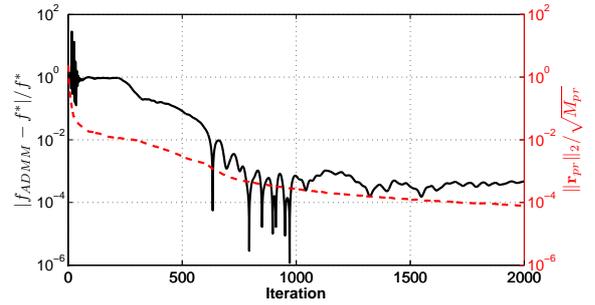


Fig. 4. Relative suboptimality and primal feasibility of the distributed ADMM algorithm.

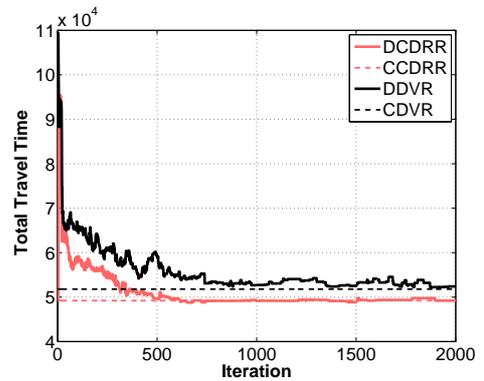


Fig. 5. Convergence of the distributed rounding schemes with respect to the corresponding centralised rounding schemes.

optimal objective value obtained from a centralised LP solver, while the right y-axis depicts the primal residual (defined in Appendix II-B, Eq. (5)) normalised over the total number of primal variables (M_{pr}). As can be seen, there is very good convergence with respect to both metrics as a reduction of 4-5 orders of magnitude is achieved, attaining values less than 10^{-3} for both primal feasibility and relative suboptimality after 1000 iterations. Such convergence is sufficient for our application because the solution of $\mathcal{R}MITSC$ is just an intermediate step towards the solution to $MITSC$ via distributed rounding. Hence, we have adopted a stopping criterion of 2000 iterations for all considered problem instances.

To examine the convergence of the distributed rounding schemes, Fig. 5 compares the solution to the original problem obtained by rounding the centralised relaxed solution and the one from the distributed ADMM algorithm for different iterations. It can be seen that the solutions of both distributed rounding schemes converges to the corresponding centralised solutions in about 800 iterations; for the particular scenario, it appears that CDR rounding is better than DVR both in terms of performance and solution fluctuation after convergence (1% and 5% fluctuation respectively).

B. Distributed Scheme Comparison

Fig. 6 depicts the relative deviation of the two distributed rounding schemes from the optimal total travel time obtained via a MILP solver for the base topology. As can be seen, the DCDRR scheme is superior to the DDVR scheme achieving results within 8% of the optimal solution for all traffic

scenarios considered. This deviation from optimality results from the sub-optimality of both the DCDRR scheme due to the NP-hardness of the *MITSC* problem and the temporal decomposition of the original problem. DDVR scheme is relatively good for the low and high traffic scenarios (10%, 30%, 70%, 90% congestion level) achieving performance within 15% of optimality, but for the moderate traffic scenario its performance is very low (76.5% of optimality). Fig. 6 also depicts the performance of the two centralised computational intelligence schemes GARI and GADI. As can be seen, the GARI scheme performs worse than DCDRR but better than DDVR while the GADI is the best performing suboptimal scheme attaining performance within 1.2% of optimality. That is because GADI uses the solutions of the distributed heuristics for initialization giving always equal or better performance than DCDRR and DDVR. Solution comparison to the optimal has not been possible for larger topologies due to the prohibitive computational cost of optimally solving the resulting MILP problems.

The performance of the distributed schemes has also been compared for larger topologies with up to 72 intersections against the performance of GARI and GADI. Fig. 7 illustrates the total travel time of the different heuristic schemes relative to GADI. As expected DCDRR is significantly superior to DDVR by up to 30% achieving performance within 6% from the best value obtained via GADI, while in most cases the performance of the two is the same. It is also interesting to observe that the relative performance of the distributed schemes is significantly affected by the level of congestion with DCDRR having the best performance over DDVR for the moderate congestion level case. However, it should be noted that the solution of DDVR is completely decentralised and requires no communication while DCDRR requires two rounds of communication for each time unit considered. Thus, there is a clear tradeoff between performance and communication overhead for the proposed distributed rounding schemes. Finally, the GARI is the worst performing scheme in almost all scenarios considered. Figs. 6 and 7 illustrate that the performance of the different solution schemes depends on the congestion level, with the 50% congestion level case yielding the worst results. The reason is that at 50% congestion level the system is on the verge of building congestion so that a “good” strategy will prevent it, while a “bad” one will not, resulting in large performance differences.

C. Scalability

The ability of the distributed ADMM algorithm to scale well for increasing problem sizes is also examined for the base topology under the moderate traffic scenario. Fig. 8(a) depicts the execution time with increasing number of cells per intersection. Because iterations must be performed synchronously for all intersections, each data point in the figure represents the maximum execution time of any intersection for a specific iteration. It can be observed that for a fixed topology the variation of the execution time for different iterations is quite small. The dotted line shows the nonlinear least-squares fit of the polynomial model $a_1x^{a_2} + a_3$, which indicates that the execution time scales almost linearly to the number of

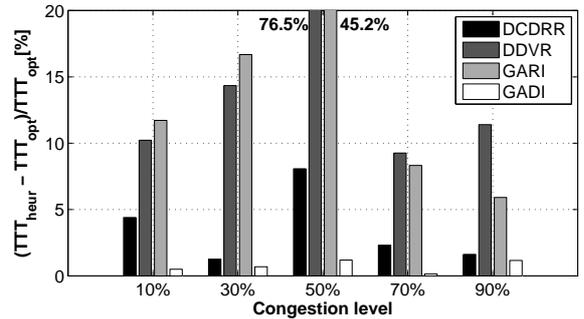


Fig. 6. Comparison between the optimal solution of MITSC and the solutions from different heuristic schemes for a 2x2 intersection topology.

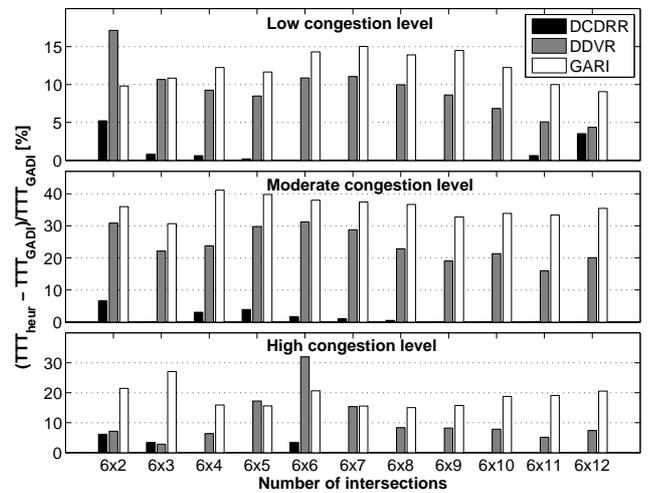


Fig. 7. Comparison of heuristic schemes with scheme GADI for large intersection topologies; absence of a bar for DCDRR implies attainment of same performance with GADI.

cells. Similar behavior can be observed about the effect of the optimization horizon time in Fig. 8(b). In fact, the exponent of the fitted model ($a_2 = 1.27$) is nearly identical to the one obtained when varying number of cells ($a_2 = 1.24$). This result implies that the computational complexity of solving the linear system of equations (25) has been reduced from $O(N^3)$ to $O(N^{1.25})$ via the selected matrix decomposition technique and the caching of the decomposition matrices. The effect of the number of intersections on the execution time is also examined in Fig. 8(c). Ideally, the execution time per iteration and intersection, should not be affected by the number of intersections. Nevertheless, different overheads can increase the maximum iteration time. The figure, clearly indicates that this overhead is relatively small, as increasing the number of intersections from 4 (2x2) to 144 (12x12) results in a 20% increase of execution time.

The demonstrated scalability results (Figs. 8(a) - 8(c)) clearly indicate that the proposed algorithm scales well to realistic problem sizes and also maintains the online nature of the proposed technique. Even if each iteration takes 100ms to complete, running 2000 iterations would require approximately 3mins of computation time (plus the communication overhead), whereas the algorithm has a deadline of 10mins to

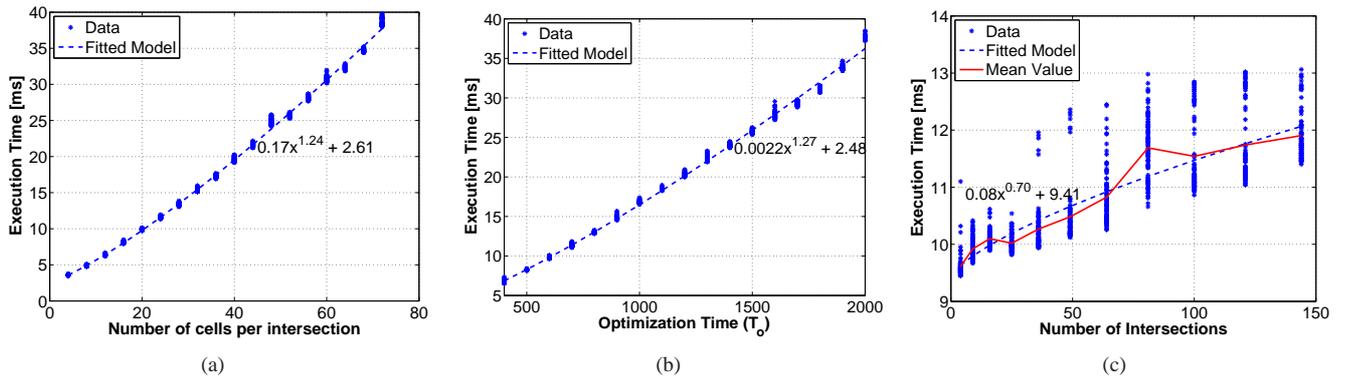


Fig. 8. Maximum execution time per iteration for all intersections with varying: (a) number of cells per intersection, (b) optimization time, (c) total number of intersections.

provide the result.

VII. CONCLUSIONS & FUTURE WORK

In this paper, we have developed distributed online solution strategies for the multiple intersection traffic signal control problem achieved via spatial and temporal decomposition. The considered problem is addressed in two phases. In the first phase, the relaxed problem is solved in a distributed fashion by an ADMM algorithm after appropriate reformulation and decomposition of the problem. In the second phase, the relaxed solution is exploited to attain solutions to the original problem through the development of two distributed rounding schemes. DCDRR scheme attempts to match the cumulative departure rates at each intersection, while DDVR minimizes the roundoff error of decision variables while satisfying the traffic light constraints. Performance evaluation has shown that ADMM has good convergence and scaling properties which allows the online distributed solution of large-scale problems. Regarding optimality, DCDRR produced results within 8% from the optimal for the considered scenario and significantly outperformed DDVR and genetic algorithms with random initialization.

There are a number of issues that can be further investigated. One direction is to examine the effect on convergence and optimality when directly solving the original problem via ADMM. Another direction is to examine how other transportation problems such as the optimal coordinated ramp metering and the network traffic state estimation can be solved in an online and distributed fashion. Furthermore, higher order macroscopic models can be considered that take into consideration more complex phenomena arising in urban road networks such as the acceleration/deceleration of vehicles and the platoon dispersion effect. Finally, the proposed approach should be tested in the context of a microsimulator to compare its performance against other distributed methods, e.g. based on multiagent reinforcement learning, and also test its robustness under traffic perturbations due to modeling uncertainties, noise and disturbances, to investigate the accommodation extend of such perturbations.

REFERENCES

- [1] S. Timotheou, C. G. Panayiotou, and M. M. Polycarpou, "Towards Distributed Online Cooperative Traffic Signal Control using the Cell Transmission Model," in *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, Netherlands*, pp. 1737–1742, IEEE, 2013.
- [2] "National Transportation Statistics," tech. rep., U.S. Department of Transportation, 2012.
- [3] P. Koonce, L. Rodegerdts, K. Lee, S. Quayle, S. Beard, C. Braud, J. Bonneson, P. Tarnoff, and T. Urbanik, "Traffic signal timing manual," tech. rep., 2008.
- [4] J. Haddad, B. De Schutter, D. Mahalel, I. Ioslovich, and P. O. Gutman, "Optimal Steady-State Control for Isolated Traffic Intersections," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2612–2617, 2010.
- [5] Y. Geng and C. G. Cassandras, "Traffic light control using Infinitesimal Perturbation Analysis," in *IEEE 51st Annual Conference on Decision and Control (CDC), 2012*, pp. 7001–7006, IEEE, 2012.
- [6] Y. S. Murat and E. Gedizlioglu, "A fuzzy logic multi-phased signal control model for isolated junctions," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 1, pp. 19–36, 2005.
- [7] N. V. Findler and J. Stapp, "Distributed approach to optimized control of street traffic signals," *Journal of Transportation Engineering*, vol. 118, no. 1, pp. 99–110, 1992.
- [8] N. H. Gartner, S. F. Assman, F. Lasaga, and D. L. Hou, "A multi-band approach to arterial traffic signal optimization," *Transportation Research Part B: Methodological*, vol. 25, no. 1, pp. 55 – 74, 1991.
- [9] L. Adacher, "A global optimization approach to solve the traffic signal synchronization problem," *Procedia - Social and Behavioral Sciences*, vol. 54, pp. 1270 – 1277, 2012. Proceedings of {EWGT2012} - 15th Meeting of the {EURO} Working Group on Transportation, September 2012, Paris.
- [10] J. Binning, G. Burtenshaw, and M. Crabtree, "Transyt 13 user guide," tech. rep., Transportation Research Laboratory, Crowthorne, Berkshire, UK, 2009.
- [11] W.-H. Lin and C. Wang, "An enhanced 0-1 mixed-integer LP formulation for traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 238–245, 2004.
- [12] C. Beard and A. Ziliaskopoulos, "System optimal signal optimization formulation," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1978, no. 1, pp. 102–112, 2006.
- [13] H. Ceylan and M. G. Bell, "Traffic signal timing optimisation based on genetic algorithm approach, including drivers routing," *Transportation Research Part B: Methodological*, vol. 38, no. 4, pp. 329–342, 2004.
- [14] Q. He, W.-H. Lin, H. Liu, and K. Head, "Heuristic algorithms to solve 0–1 mixed integer LP formulations for traffic signal control problems," in *IEEE International Conference on Service Operations and Logistics and Informatics (SOLI), 2010*, pp. 118–124, IEEE, 2010.
- [15] D. Robertson and R. Bretherton, "Optimizing networks of traffic signals in real time-the SCOOT method," *IEEE Transactions on Vehicular Technology*, vol. 40, pp. 11–15, Feb 1991.
- [16] C. Bielefeldt and F. Busch, "Motion-a new on-line traffic signal network control system," in *Processing of the Seventh International Conference on Road Traffic Monitoring and Control*, pp. 55–59, Apr 1994.
- [17] A. G. Sims and K. Dobinson, "The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits," *IEEE Transactions on Vehicular Technology*, vol. 29, no. 2, pp. 130–137, 1980.
- [18] P. Mirchandani and L. Head, "A real-time traffic signal control system:

architecture, algorithms, and analysis,” *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 6, pp. 415–432, 2001.

- [19] K. Aboudolas, M. Papageorgiou, A. Kouvelas, and E. Kosmatopoulos, “A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks,” *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 680–694, 2010.
- [20] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, “Fast model predictive control for urban road networks via MILP,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 846–856, 2011.
- [21] T. L. Thorpe and C. W. Anderson, “Traffic light control using sarsa with three state representations,” tech. rep., IBM Corporation, 1996.
- [22] M. Wiering, J. v. Veenen, J. Vreeken, and A. Koopman, “Intelligent traffic light control,” tech. rep., Utrecht University: Information and Computing Sciences, 2004.
- [23] M. Vasirani and S. Ossowski, “A computational market for distributed control of urban road traffic systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 313–321, 2011.
- [24] A. Bazzan, “A distributed approach for coordination of traffic signal agents,” *Autonomous Agents and Multi-Agent Systems*, vol. 10, no. 1, pp. 131–164, 2005.
- [25] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, “Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): Methodology and large-scale application on downtown Toronto,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, pp. 1140–1150, Sept 2013.
- [26] T. Wongpiromsarn, T. Uthacharoenpong, Y. Wang, E. Frazzoli, and D. Wang, “Distributed traffic signal control for maximum network throughput,” in *15th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2012, pp. 588–595, IEEE, 2012.
- [27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [28] C. F. Daganzo, “The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory,” *Transportation Research Part B: Methodological*, vol. 28, no. 4, pp. 269–287, 1994.
- [29] M. J. Lighthill and G. B. Whitham, “On kinematic waves. II. A theory of traffic flow on long crowded roads,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.
- [30] P. I. Richards, “Shock waves on the highway,” *Operations research*, vol. 4, no. 1, pp. 42–51, 1956.
- [31] H. Rakha and B. Crowther, “Comparison of Greenshields, Pipes, and Van Aerde car-following and traffic stream models,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1802, no. 1, pp. 248–262, 2002.
- [32] C. M. Tampère, R. Corthout, D. Cattrysse, and L. H. Immers, “A generic class of first order node models for dynamic macroscopic simulation of traffic flows,” *Transportation Research Part B: Methodological*, vol. 45, no. 1, pp. 289–309, 2011.
- [33] M. Maher, “A comparison of the use of the cell transmission and platoon dispersion models in transyt 13,” *Transportation Planning and Technology*, vol. 34, no. 1, pp. 71–85, 2011.
- [34] R. Glowinski and A. Marrocco, “Sur l’approximation, par elements finis d’ordre un, et la resolution, par penalisation-dualité, d’une classe de problèmes de Dirichlet non lineaires,” *Revue Française d’Automatique, Informatique, et Recherche Opérationnelle*, vol. 9, pp. 41–76, 1975.
- [35] T. Goldstein, B. O’Donoghue, and S. Setzer, “Fast alternating direction optimization methods,” *CAM report*, pp. 12–35, 2012.
- [36] Y. Wang, J. H. van Schuppen, and J. L. M. Vrancken, “On-line Distributed Prediction of Traffic Flow in a Large-Scale Traffic Network,” in *Proceedings of the 19th ITS World Congress, Vienna, Austria*, pp. 1–15, October 2012.
- [37] K. Doan and S. V. Ukkusuri, “On the holding-back problem in the cell transmission based dynamic traffic assignment models,” *Transportation Research Part B: Methodological*, vol. 46, no. 9, pp. 1218–1238, 2012.
- [38] J. Vrancken, J. H. van Schuppen, M. dos Santos Soares, and F. Ottenhof, “A hierarchical network model for road traffic control,” in *International Conference on Networking, Sensing and Control, 2009. ICNSC’09.*, pp. 340–344, IEEE, 2009.
- [39] N. Jorge and S. J. Wright, *Numerical optimization*, vol. 2. New York: Springer, 1999.
- [40] I. S. Duff, “MA57—a code for the solution of sparse symmetric definite and indefinite systems,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 30, no. 2, pp. 118–144, 2004.
- [41] I. Gurobi Optimization, “Gurobi optimizer reference manual,” 2013.



Stelios Timotheou is a Research Associate at the KIOS Research Center for Intelligent Systems and Networks of the University of Cyprus (UCY). He received a B.Sc. from the Electrical and Computer Engineering (ECE) School of the National Technical University of Athens, and an M.Sc. and Ph.D. from the Electrical and Electronic Engineering Department of Imperial College London. In previous appointments, he was a Visiting Lecturer at the ECE Department of UCY, a Research Associate at the Computer Laboratory of the University of Cambridge and a Visiting Scholar at the Intelligent Transportation Systems Center & Testbed, University of Toronto. His research focuses on the modelling and system-wide solution of problems in complex and uncertain environments that require real-time and close to optimal decisions by developing optimisation, machine learning and computational intelligence techniques. Application areas of his work include intelligent transportation systems, disaster management, neural networks, and communication systems. He is a member of the IEEE and the ACM.

bridge and a Visiting Scholar at the Intelligent Transportation Systems Center & Testbed, University of Toronto. His research focuses on the modelling and system-wide solution of problems in complex and uncertain environments that require real-time and close to optimal decisions by developing optimisation, machine learning and computational intelligence techniques. Application areas of his work include intelligent transportation systems, disaster management, neural networks, and communication systems. He is a member of the IEEE and the ACM.



Christos Panayiotou has received a B.Sc. and a Ph.D. degree in Electrical and Computer Engineering from the University of Massachusetts at Amherst, in 1994 and 1999 respectively. He also received an MBA from the Isenberg School of Management, at the aforementioned university in 1999. From 1999 to 2002 he was a Research Associate at the Center for Information and System Engineering (CISE) and the Manufacturing Engineering Department at Boston University. Since 2002 he has been with the Electrical and Computer Engineering Department at University of Cyprus (UCY) where he is currently an Associate Professor. His research interests include optimization and control of discrete-event and hybrid systems, transportation networks, sensor networks, computer communication networks, distributed control systems, fault diagnosis and fault tolerant systems, resource allocation, simulation. Christos is an Associate Editor for the Conference Editorial Board of the IEEE Control Systems Society, the Journal of Discrete-Event Dynamical Systems, and the European Journal of Control. He is also a reviewer for various conferences and journals, and he has served in the organizing and program committees of various international conferences.

Department at University of Cyprus (UCY) where he is currently an Associate Professor. His research interests include optimization and control of discrete-event and hybrid systems, transportation networks, sensor networks, computer communication networks, distributed control systems, fault diagnosis and fault tolerant systems, resource allocation, simulation. Christos is an Associate Editor for the Conference Editorial Board of the IEEE Control Systems Society, the Journal of Discrete-Event Dynamical Systems, and the European Journal of Control. He is also a reviewer for various conferences and journals, and he has served in the organizing and program committees of various international conferences.



Marios M. Polycarpou is a Professor of Electrical and Computer Engineering and the Director of the KIOS Research Center for Intelligent Systems and Networks at the University of Cyprus. He received the B.A. degree in Computer Science and the B.Sc. degree in Electrical Engineering both from Rice University, Houston, TX, USA in 1987, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Southern California, Los Angeles, CA, in 1989 and 1992 respectively. Prior to joining the University of Cyprus as founding Department

Chair in 2001, he was Professor of Electrical and Computer Engineering and Computer Science at the University of Cincinnati, Ohio, USA. His teaching and research interests are in intelligent systems and control, fault diagnosis, computational intelligence, adaptive and cooperative control, and large-scale systems. Dr. Polycarpou has published over 230 papers in these areas and is the holder of 6 patents. Prof. Polycarpou is a Fellow of the IEEE and currently serves as the President of the IEEE Computational Intelligence Society. He has served as the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems from 2004 until 2010. He participated in more than 60 research projects/grants, funded by several agencies and industry in Europe and the United States. In 2011, Dr. Polycarpou was awarded the prestigious European Research Council (ERC) Advanced Grant.