Formal Synthesis of Distributed Optimal Traffic Control Policies

Sadra Sadraddini
Boston University
110 Cummington Mall
Boston, MA, USA
sadra@bu.edu

János Rudan
Boston University
110 Cummington Mall
Boston, MA, USA
rudanj@bu.edu

Calin Belta
Boston University
110 Cummington Mall
Boston, MA, USA
cbelta@bu.edu

ABSTRACT

We propose a formal methods approach to control traffic signals optimally from specifications described by metric temporal logic (MTL). Since real-time optimization is computationally infeasible beyond small-scale networks, we use a divide and conquer approach. We decompose the network into smaller subnetworks and synthesize assume-guarantee contracts for their interconnections. We show how to exploit mathematical properties of traffic dynamics to find time varying contracts by solving a constraint satisfaction problem. A model predictive control (MPC) approach is used to find local controls for each subnetwork to minimize induced delays, while assume-guarantee contracts and appropriately designed terminal constraints ensure the satisfaction of the specification all over the network. We present a case study on an urban traffic network.

CCS CONCEPTS

• **Networks** → **Formal specifications**; • **Theory of computation** → *Distributed algorithms*; Mixed discrete-continuous optimization; • **Applied computing** → *Transportation*;

KEYWORDS

Traffic Control; Formal Synthesis; Model Predictive Control

ACM Reference format:
Sadra Sadraddini, János Rudan, and Calin Belta. 2017. Formal Synthesis of Distributed Optimal Traffic Control Policies. In *Proceedings of The 8th ACM/IEEE International Conference on Cyber-Physical Systems, Pittsburgh, PA USA, April 2017 (ICCPS 2017),* 10 pages.
DOI: http://dx.doi.org/10.1145/3055004.3055011

1 INTRODUCTION

Traffic signals are the primary mechanism for traffic management. Typical signals are traffic lights in urban setting and ramp meters and (less commonly used) variable speed limits on freeways. There has been an extensive amount of research on control methodologies of traffic signals in recent years [4, 12, 14, 24]. Although traffic flow estimation techniques have much improved due to recent advances in sensing technologies, traffic control problems are still challenging. Traffic models are mathematically complex - they often include discrete and continuous variables (hybrid systems)- and various constraints are present to capture the safe operation of traffic signals (e.g. restrictions on the duration of traffic lights). The ultimate objective of traffic management is to minimize a cost such as the induced delay in the network, while taking into account the dynamics and constraints of the network.

Due to computational complexity, traditional optimal control techniques such as Hamilton-Jacobi-Bellman (HJB) equations are not suitable for traffic control. Traffic-responsive strategies employ approximate dynamic programming methods such as model predictive control (MPC) for real-time optimization. Notable implementations are SCOOT [16], OPAC [11] and UTOPIA [21]. However, real-time optimization is not possible beyond small-scale systems. Totally decentralized methods optimize the controls of each intersection individually but can cause gridlocks in the network [12]. Hierarchical distributed control architectures [22], while alleviating the real time computational complexity, are not able to formally guarantee global behaviors such as avoidance of traffic jams. In this paper, we propose a method in which, while control decisions are optimized locally, desired specifications are *guaranteed* globally.

Formal methods approaches have received increased attention from the control community in recent years. Using tools such as model checking [3] and automata theory, correct by design methods have been developed to control hybrid systems from high-level specifications described by temporal logics [31, 32]. In fact, traffic networks are amenable to such high-level specifications that can capture behaviors such as prevention of congestion, sequentiality of traffic lights and various reactiveness properties. The authors in [7] studied the control of traffic networks from linear temporal logic (LTL) specifications. However, there are drawbacks to existing methods. First, automata-based approaches require constructing finite-state abstractions. Even though mathematical properties of traffic networks facilitate efficient computation of finite-state transitions [5], state-space discretization is still necessary and computationally intractable beyond very few dimensions. Second, optimal control is not yet fully combined with finite-state approaches for the case when the specifications include safety properties. Recent works on MPC from signal temporal logic (STL) specifications [25, 27] rely on finite horizon predictions and do not guarantee safety properties and are vulnerable to disturbances.

We propose a method that mitigates the issues above. We use metric temporal logic (MTL) to describe system specifications. MTL is an extension of LTL with bounds on temporal operators, which is very useful to express time intervals for requirements in traffic networks. Furthermore, it is more general than STL as it can reason about both discrete-domain and continuous-domain signals, which is useful for expressing requirements for discrete variables such as traffic lights. We use a discrete-time piecewise affine model for traffic dynamics, where additive disturbances represent exogenous

ICCPS 2017, Pittsburgh, PA USA

DOI: http://dx.doi.org/10.1145/3055004.3055011

vehicular flow into the network. We follow a divide and conquer approach by partitioning the network into smaller subnetworks and synthesize controls locally for each subnetwork. For the dynamical interconnections between the subnetworks, we take an assume-guarantee approach [15]. Each subnetwork assumes the interconnection effects from its neighbors satisfy a set of contracts, while the controllers of neighboring subnetworks promise to maintain those contracts. Therefore, local controls can be planned optimally in a decentralized manner but with a global coordination induced by the contracts and the global network specification, which is ensured to be satisfied.

The main contributions of the paper are as follows. First, we provide a method to synthesize assume-guarantee contracts by feasibility checking of a mixed-integer linear programming (MILP) problem, which is computed offline and it can be applied to relatively large networks. Second, we find local controls optimally using a robust MPC approach, which has feasibility guarantees for both the local constraints and contracts hence the overall global specification is ensured. We present a case study on an urban traffic network and provide preliminary results on applying our methods to microscopic traffic models.

This work is related to the recent papers on compositional synthesis [2, 26]. The authors in [18] studied compositional synthesis for traffic networks from LTL specifications. However, this approach requires assumption mining [19] for all subnetworks, which is a computationally expensive procedure and it is conservative since it only considers fixed-time contracts, as opposed to our work where contracts are time varying. Compositional methods, ideally and under some assumptions, are applicable to arbitrarily large systems. Our contract synthesis process is centralized and is closely related to the methodology in [23], where the authors introduce *separable* control invariant sets for linear systems with fixed feedback gain. As opposed to [23], our method is able to deal with the hybrid nature of traffic networks, does not require fixed feedback structure, considers a much richer class of specifications, and finds controls optimally over a prediction horizon.

This paper is organized as follows. First, we provide necessary background on MTL and traffic network modeling in Sec. 2. We formulate the problem in Sec. 3. In Sec. 4, we explain how to partition a network considering constraints of the system and specification. The technical details on computation of assume-guarantee contracts and control synthesis are explained in Sec. 5 and Sec. 6, respectively. The case study is presented in Sec. 7.

2 PRELIMINARIES

2.1 Partially Ordered Sets

A partially ordered set [9] $\mathcal{S}$ is a set associated with a partial order relation $\preceq_{\mathcal{S}}$ that satisfies the following properties: 1) reflexivity: $s \preceq_{\mathcal{S}} s, \forall s \in \mathcal{S}$; 2) antisymmetry: $s_1 \preceq_{\mathcal{S}} s_2, s_2 \preceq_{\mathcal{S}} s_1 \Rightarrow s_1 = s_2$; 3) transitivity: $s_1 \preceq_{\mathcal{S}} s_2, s_2 \preceq_{\mathcal{S}} s_3 \Rightarrow s_1 \preceq_{\mathcal{S}} s_3$. Given an element $s \in \mathcal{S}$, we define the set $L(s) = \{s' \in \mathcal{S} | s' \preceq_{\mathcal{S}} s\}$. For the case $\mathcal{S} = \mathbb{R}^n$, we define the partial order relation $\preceq_+$ such that $s_1 \preceq s_2 \Leftrightarrow s_2 - s_1 \in \mathbb{R}^n_+$, where $\mathbb{R}^n_+$ is the positive orthant.

Definition 2.1. A set $\mathcal{S}' \subseteq \mathcal{S}$ is a *lower-set* if for all $s' \in \mathcal{S}'$, we have $L(s') \subseteq \mathcal{S}$.

2.2 Metric Temporal Logic

The detailed, formal definition of MTL is not presented here as it can be found in [20]. Here we just provide the necessary notation and some examples. Informally, MTL is constructed from a finite set of atomic propositions $AP = \{p_1, p_2, \cdots, p_{np}\}$, Boolean operators: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), and temporal operators: $\mathbf{G}_{[t_1,t_2)}$ (always between t_1 and t_2), $\mathbf{F}_{[t_1,t_2)}$ (eventually between t_1 and t_2), and $\mathbf{U}_{[t_1,t_2)}$ (until between t_1 and t_2). Punctualities are also allowed using $\mathbf{F}_{\{t\}}$ (exactly at time t). The set of atomic propositions that are true at time t is denoted by $o_t \in 2^{AP}$, which we refer to as the *observation* at time t. An infinite word over 2^{AP} is written as $\sigma := o_0 o_1 o_2 \cdots$. We use $\sigma_{[t_1,t_2)}, t_1 < t_2$, to refer to a specific portion of the word: $\sigma_{[t_1,t_2)} = o_{t_1} o_{t_1+1} \cdots o_{t_2-1}$, and σ_t for a suffix of a word starting at time t: $\sigma_t = o_t o_{t+1} \cdots$. The semantics of MTL formulas are defined over word suffixes. We write $\sigma_t \models \varphi$ to indicate suffix σ_t satisfies MTL formula φ.

Example 2.2. Let $AP = \{p_1, p_2\}$ and consider the infinite word $\sigma = \{p_1\}\emptyset\{p_2\}\{p_1, p_2\}\{p_2\}\overline{\emptyset}$, where overline stands for infinite repetition and $\emptyset$ is the empty set. We have $\sigma_2 \models \mathbf{G}_{[0,3)} p_2$ (always between 0 and 3, p_2 is true) since $\sigma_2 = \{p_2\}\{p_1, p_2\}\{p_2\}\overline{\emptyset}$ and p_2 appears in all of the first three observations in σ_2. However, $\sigma_4 \not\models \mathbf{F}_{[0,\infty)} p_1$ (eventually p_1 is true) since $\sigma_4 = \{p_2\}\overline{\emptyset}$ and p_1 never appears in this suffix.

The *horizon* of an MTL formula φ, denoted by $h(\varphi)$, is defined as the required length of a suffix to decide the satisfaction of φ. For the formulas in Example 2.2, we have $h(\mathbf{G}_{[0,3)} p_2) = 3$ and $h(\mathbf{F}_{[0,\infty)} p_1) = \infty$. A recursive definition for computing the horizon of an MTL formula is given in [10]. The satisfaction of φ by σ_t is decided only by $\sigma_{[t,\, t+h^{\varphi})}$ and the rest of the observations are irrelevant.

Definition 2.3. An MTL formula φ is *bounded* if $h(\varphi) < \infty$.

2.3 Traffic Network Model

A link l is defined as a one-way traffic road segment with the following attributes: $c_l \in \mathbb{R}_+$ is the capacity of l; $x_{l,t} \in [0, c_l]$ is the volume of vehicles on l at time t, $t \in \mathbb{N}$; $q_l \in \mathbb{R}_+$ is the maximum outflow (maximum volume of vehicles that can flow out of l in one time step); $u_{l,t} \in \mathcal{U}_l$ is the control input of l at time t, where $\mathcal{U}_l$ depends on the type of l:

- if l is controlled with traffic lights: $\mathcal{U}_l = \{0, 1\}$, where 1 (0) corresponds to green (red) light;
- if l is controlled with ramp meter/speed limit: $\mathcal{U}_l = [0, 1]$, where u_l is the ratio of the maximum outflow q_l that is allowed to flow out of the link in one time step;
- if l is uncontrolled: $\mathcal{U}_l = \{1\}$.

Additional constraints on control inputs (e.g., sequentiality constraints) are expressed using MTL formulas and are considered as a part of the problem formulated in Sec. 3.

Definition 2.4. A traffic network is defined as a tuple

$$\mathcal{N} = (\mathcal{L}, \delta, \alpha, \beta, \mathcal{A}, w),$$

where:

- $\mathcal{L}$ is the set of links in the network;

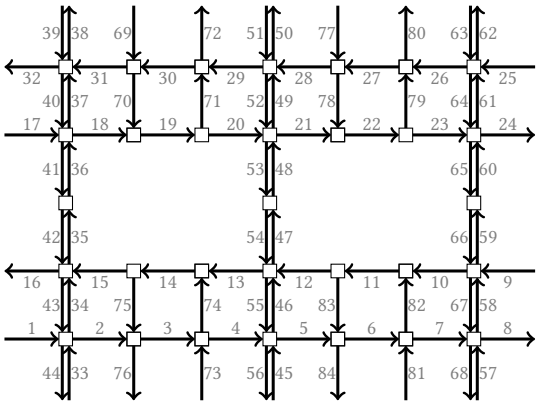

Figure 1: Traffic Network Topology

- $\delta : \mathcal{L} \to 2^{\mathcal{L}}$, where $\delta(l)$ is the set of downstream links of l (downstream function; defines network topology);
- $\alpha : \mathcal{L} \times \mathcal{L} \to [0, 1]$, where $\alpha(l, l'), l' \in \delta(l)$, is the proportion of vacancies available in l' dedicated to l, which is assumed to be constant (capacity ratios);
- $\beta : \mathcal{L} \times \mathcal{L} \to [0, 1]$, where $\beta(l, l'), l' \in \delta(l)$, is the ratio of volume flowing from l to l', which is assumed to be constant (turning ratios);
- $\mathcal{A} \subset \mathcal{L} \times \mathcal{L}$ is the set of *antagonistic pairs*. Two links form an antagonistic pair if their traffic lights are not allowed to be green simultaneously [1] (defines traffic phases);
- $w : \mathcal{L} \times \mathbb{N} \to \mathbb{R}_+$, where $w(l, t)$ is the *exogenous demand* (volume of vehicles entering from outside of the network) towards l at time t.

Although not required in the technical approach of this paper, it is helpful to visualize a traffic network as a directed graph (see Fig. 1).

Example 2.5. Consider the network in Fig. 1 with 84 links. This network represents two urban areas on north and south sides connected by three bridges in between (the network topology is inspired by the Boston-Cambridge area). Each link is shown as a directed edge between nodes (intersections) shown as squares. For all links we have $c_l = 40, q_l = 15$. The long bridges in the middle are divided into two separate links, and links 35, 41, 47, 53, 59, 65 are uncontrolled (i.e., there are no traffic lights in the middle of the bridges). We have $l' \in \delta(l)$ if the head of edge representing l is followed by the tail of edge representing l'. For example, we have $\delta(1) = \{2, 34, 44\}$, $\delta(14) = \{15, 75\}$, $\delta(53) = \{54\}$, $\delta(76) = \emptyset$, etc. Antagonistic pairs are determined by trivial traffic conventions. For instance, $\{12, 54\} \in \mathcal{A}$, as the pair head into a common intersection in perpendicular directions. The values for $w(l, t)$ vary between 0 and 8 vehicles per time step, depending on the location of the link. The detailed valuations for w, and for other network components from Definition 2.4 including α and β, are not provided here but are available in [1].

We define the set of outgoing links of a network by $\mathcal{L}^{\text{out}} := \{l \in \mathcal{L} | \delta(l) = \emptyset\}$ and the set of internal links as $\mathcal{L}^{\text{int}} := \mathcal{L} \setminus \mathcal{L}^{\text{out}}$.

[1] We assume that all antagonistic pairs are controlled by traffic lights.

When describing the dynamics of a traffic network $\mathcal{N}$, we are only interested in the evolution of the internal links. The outflow of an internal link l at time t is defined as:

$$f_{l,t} := \min \left\{ x_{l,t}, u_{l,t} q_l, \min_{l' \in \delta(l)} \frac{\alpha(l, l')}{\beta(l, l')} (c_{l'} - x_{l',t}) \right\}. \quad (1)$$

Note that the outflow is zero if $u_{l,t} = 0$ (red light). The last argument of the minimizer is determined by the minimum available vacancy in the downstream links of l. Physically, the outflow model above is governed by the first-in-first-out (FIFO) rule [6]. As a consequence of this rule, lack of enough vacancy in a link blocks the flow of its upstream links to all other surrounding links. For example, in Fig. 1, if link 4 does not have enough vacancy for accommodating flow from link 3, the flow from 3 to 74 is also blocked.

For all antagonistic pairs $(l, l') \in \mathcal{A}$, we have $u_{l,t} + u_{l',t} \leq 1, \forall t \in \mathbb{N}$. The evolution of an internal link l is given by:

$$x_{l,t+1} = \min \left\{ x_{l,t} - f_{l,t} + \sum_{l', l \in \delta(l')} \beta(l', l) f_{l',t} + w(l, t),\ c_l \right\}. \quad (2)$$

The volume of vehicles that leave the network through an outgoing link at time t is:

$$y_{l,t} := \sum_{l', l \in \delta(l')} \beta(l', l) f_{l',t},\ l \in \mathcal{L}^{\text{out}}. \quad (3)$$

The network dynamics is represented in a compact form as the following discrete-time system:

$$x_{t+1} = F(x_t, u_t, w_t), \quad (4)$$

$$y_t = G(x_t, u_t), \quad (5)$$

where $x_t = \{x_{l,t}\}_{l \in \mathcal{L}^{\text{int}}}$, is the *state* at time t. We have $x_t \in \mathcal{X}, \forall t \in \mathbb{N}$, where $\mathcal{X} = \prod_{l \in \mathcal{L}^{\text{int}}} [0, c_l]$. $u_t = \{u_{l,t}\}_{l \in \mathcal{L}^{\text{int}}}$ is the *control input* at time t. We have $u_t \in \mathcal{U}, \forall t \in \mathbb{N}$, where $\mathcal{U} \subseteq \prod_{l \in \mathcal{L}^{\text{int}}} \mathcal{U}_l$. $w_t = \{w(l, t)\}_{l \in \mathcal{L}^{\text{int}}}$ is the *additive disturbance* at time t. We have $w_t \in \mathcal{W}, \forall t \in \mathbb{N}$, where $\mathcal{W} \subset \mathbb{R}_+^{|\mathcal{L}^{\text{int}}|}$ is the set of admissible disturbances. $y_t = \{y_{l,t}\}_{l \in \mathcal{L}^{\text{out}}}$ is the *output* at time t. We have $y_t \subset \mathbb{R}_+^{|\mathcal{L}^{\text{out}}|}, \forall t \in \mathbb{N}$. Note that both $\mathcal{X}$ and $\mathcal{W}$ are real-valued, so the model treats the flow of vehicles as a fluid. Also, both $F : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \to \mathcal{X}$ and $G : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_+^{|\mathcal{L}^{\text{out}}|}$ are piecewise affine and positive.

3 PROBLEM STATEMENT

Consider a traffic network $\mathcal{N}$.

Assumption 1. *The set of admissible additive disturbances is in the form* $\mathcal{W} = L(w^{\max})$, $w^{\max} \in \mathbb{R}_+^{|\mathcal{L}^{int}|}$.

The assumption above is reasonable when the sources of the exogenous demands are independent of each other. Thus we have $w(l, t) \in [0, w_{l,t}^{\max}], \forall l \in \mathcal{L}^{\text{int}}, \forall t \in \mathbb{N}$.

Definition 3.1. A *control policy* $\mu = \{\mu_t \big| t \in \mathbb{N}\}$ is a set of relations that map the (partial) history of state and controls into an admissible control action:

$$u_t = \mu_t(x_0, \cdots, x_t, u_0, \cdots, u_{t-1}),$$

where $\mu_t : \mathcal{X}^{t+1} \times \mathcal{U}^t \to \mathcal{U}$.

We assume full and exact state knowledge. We later explain how to relax this assumption in Sec. 6. Given an initial condition x_0, a control policy μ and a sequence of additive disturbances $\mathbf{w} = w_0, w_1, \cdots$, the *system run* is defined as:

$$\zeta(x_0, \mu, \mathbf{w}) := (x_0, u_0), (x_1, u_1), \cdots . \tag{6}$$

Similarly, the *output run* is defined as $\xi(x_0, \mu, \mathbf{w}) := y_0, y_1, \cdots$.

Definition 3.2. The congestion-free set of a network is defined as the set $\Pi \subset \mathcal{X} \times \mathcal{U}$, where:

$$\Pi := \quad \Big\{(x,u)\Big| \min\{u_l q_l, x_l\} \leq \min_{l' \in \delta(l)} \Big\{\frac{\alpha(l,l')}{\beta(l,l')}(c_{l'} - x_{l'})\Big\} \wedge \\ x_l - f_l + \sum_{l', l \in \delta(l')} \beta(l', l) f_{l'} + w_l^{\max} \leq c_l, \forall l \in \mathcal{L}^{\text{int}}\Big\}.$$

If the system is in the congestion-free set, then the two following properties hold. First, the last argument from the minimization in (1) is never the minimizer. Thus, the flow of a link is never obstructed due to the lack of enough vacancy in its downstream links. Second, c_l in (2) is also never the minimizer. Therefore, the total exogenous demand from outside of the network is accommodated in the network hence there is no flow obstruction from outside of the network as well. Our primary interest is finding a control policy such that the evolution of the network is always restricted to the congestion-free set.

Furthermore, we are also interested in various other objectives described using MTL. We allow for two types of atomic propositions. First are linear predicates over state, which are of the following form:

$$p_x = (a_1 x_{l_1} + a_2 x_{l_2} + \cdots + a_{n_p} x_{l_{n_p}} \leq b), \tag{7}$$

where $l_1, l_2, \cdots, l_{n_p}$ are the links whose vehicular volumes appear in p_x and $b, a_i \in \mathbb{R}_+, i = 1, \cdots, n_p$. Since all values are positive, the half space induced by the linear predicate above is a lower-set in $\mathcal{X}$ with partial order relation $\preceq_+$. Therefore, state predicates are not falsified when vehicular volumes are decreased. It is also assumed that no negation operator applies to state predicates. The second are predicates over controls, which are in the following form:

$$p_u = (u_l \sim b_u), \tag{8}$$

where $\sim \in \{\leq, \geq, =\}, b_u \in [0, 1]$. Using MTL temporal operators and Boolean connectives, we can describe a wide variety of temporal properties for traffic networks.

Example 3.3. Consider the network in Fig. 1 and the following MTL specifications:

- $\varphi_1 = \mathbf{F}_{[0,6)}\left((u_{12} = 0) \wedge (u_{46} = 0) \wedge (u_{54} = 0)\right)$; which reads: "within 6 time units, all the traffic lights of links heading toward intersection at middle southern area turn red (hence pedestrians can cross the intersection in diagonal directions)".
- $\varphi_2 = \neg\Big((u_{28} = 0) \ \wedge \ \mathbf{F}_{\{1\}}(u_{28} = 1) \wedge \mathbf{F}_{\{2\}}(u_{28} = 0)\Big)$; which states: "the traffic light of link 28 can not be green for just one time step."
- $\varphi_3 = (x_{59} + x_{60} + x_{65} + x_{66} \leq 100)$; which states: "the total volume of the vehicles on the eastern bridge is less than 100."
- $\varphi_4 = (x_{73} \leq 5) \vee \mathbf{F}_{[0,4)}(u_{73} = 1)$; which translates to: "if the volume of vehicles on link 73 exceeds 5, its traffic light eventually turns green within 4 time units."

Given a bounded MTL formula φ, we consider specifications of the following form:

$$\Phi^{global} := \mathbf{G}_{[0,\infty)}\Phi, \tag{9}$$

where $\Phi = ((x, u) \in \Pi) \wedge \varphi$, and Π is the congestion-free set from Definition 3.2. The operator $\mathbf{G}_{[0,\infty)}$ (unbounded always) ensures that the congestion-free property and the requirements in φ hold for all times. Note that $h(\Phi) = h(\varphi)$. It can be shown that the proposition $((x, u) \in \Pi)$ can be transformed into a Boolean formula over state and control predicates by translating the minimizers into mixed-logical equations. We omit the explanation here as a similar procedure can be found in [13]. Given Φ and its atomic propositions in the form of (7), (8), the infinite word generated by run (6) is denoted by $\sigma(\zeta(\mu, x_0, \mathbf{w}))$.

Control policies guaranteeing satisfaction of Φ^{global} are often not unique. Thus, we are interested in choosing one optimally with respect to a cost function. The cost criterion that we consider in this paper is the total amount of delay induced in the network, which is defined as:

$$J(x_0, \mu, \mathbf{w}) = \sum_{t=0}^{\infty} \sum_{l \in \mathcal{L}^{\text{int}}} \gamma^t (x_{l,t} - f_{l,t}), \tag{10}$$

where $\gamma \in (0, 1)$ is the discount factor that is introduced to make the infinite horizon cost properly defined. Observe that $x_{l,t} - f_{l,t}$ is the volume of vehicles on l at time t that are unable to travel in the network for one time step. It is straightforward to show that $J \leq \frac{1}{1-\gamma} \sum_{l \in \mathcal{L}^{\text{int}}} c_l$.

Problem 1. *Given a traffic network $\mathcal{N}$ with dynamics* (4)*, a time bounded MTL specification φ with atomic propositions in the form of* (7), (8)*, and the cost function J as in* (10)*, find a control policy μ and a set of initial conditions $\mathcal{X}_0 \subset \mathcal{X}$ such that*

$$\sigma_0(\zeta(x_0, \mu, \mathbf{w})) \models \Phi^{global}, \forall x_0 \in \mathcal{X}, \forall \mathbf{w}.$$

Furthermore, given $x_0 \in \mathcal{X}_0$, choose the optimal control policy such that the worst-case cost is minimized:

$$\begin{aligned} &\textit{minimize} \quad \max_{\mathbf{w}} J((x_0, \mu, \mathbf{w})) \\ &\textit{subject to} \quad \sigma_0(\zeta(x_0, \mu, \mathbf{w})) \models \Phi^{global}, \forall \mathbf{w}. \end{aligned} \tag{11}$$

Our approach to Problem 1 involves some approximations. First, we only find a subset of all admissible initial conditions. We discuss the completeness of our method in Sec. 5. Second, we approximate the infinite horizon constrained optimal control in Problem 1 as a receding horizon optimal control problem, i.e. we use a model predictive control (MPC) scheme. As mentioned earlier, there are two primary challenges to this approach. First, we need to guarantee that control synthesis in a receding horizon manner ensures global specification (9), which has infinite time semantics. This issue is covered in Sec. 6. Second, constrained optimization is computationally expensive beyond small networks hence controls can not be found in a centralized manner in real time. To overcome this issue, we partition the network into smaller subnetworks. The dynamics of subnetworks become interconnected since vehicles that leave a

subnetwork arrive in other subnetworks. As mentioned earlier, we follow an assume-guarantee approach to design contracts for the interconnections of the subnetworks. The details are explained in Sec. 5. Each subnetwork's MPC incorporates the relevant contracts, as explained in Sec. 6. Due to space limit, the proofs are omitted here.

4 NETWORK PARTITIONING

In this section, we explain how to partition a network into smaller subnetworks. We write φ in conjunction normal form (CNF) [2]:

$$\varphi = \bigwedge_{k=1}^{n_\varphi} \varphi_k^{conj}, \tag{12}$$

where each φ_k^{conj} can not be written as a conjunction of multiple MTL formulas. We define

$$\text{Links}(\varphi) = \{l \in \mathcal{L} | x_l \text{ or } u_l \text{ appears in } \varphi\}.$$

Assumption 2. *MTL formula φ is* sparse *in the sense that for all $k, k' \in \{1, \cdots, n_\varphi\}, k \neq k'$, we have $Links(\varphi_k^{conj}) \subseteq Links(\varphi_{k'}^{conj})$ or $Links(\varphi_k^{conj}) \cap Links(\varphi_{k'}^{conj}) = \emptyset$.*

The assumption above is reasonable in traffic networks since we are usually interested in requirements at specific locations. While our method can handle non-sparse specifications by introducing some conservativeness (when designing contracts in Sec. 5), we do not discuss it in this paper.

Given a network $\mathcal{N}$ and an integer N, we partition $\mathcal{N}$ into $\mathcal{N}^1, \mathcal{N}^2, \cdots, \mathcal{N}^N$. We take the following considerations into account when a network is partitioned. First, the set of subnetwork internal links has to be disjoint: $\bigcup_{i=1}^N \mathcal{L}^{\text{int},i} = \mathcal{L}^{\text{int}}, \mathcal{L}^{\text{int},i} \cap \mathcal{L}^{\text{int},j} = \emptyset, \forall i \neq j$, where $\mathcal{L}^{\text{int},i}$ is the set of internal links of subnetwork $\mathcal{N}^i$. Second, we desire that the size of a subnetwork (defined by the number of its internal links) does not exceed a predetermined bound K, hence the size of its MPC optimization problem (which is directly related to the number of internal links) is confined. Therefore, we have $|\mathcal{L}^{\text{int},i}| \leq K, i = 1, \cdots, N$. Third, MTL formula φ has to be translated into a conjunction of "local" MTL formulas for each subnetwork. Using the CNF of φ, we require that $\exists i \in \{1, \cdots, N\}$ such that $\text{Links}(\varphi_k^{conj}) \subseteq \mathcal{L}^{\text{int},i}$, $k = 1, \cdots, n_\varphi$. Fourth, for all $l, l' \in \mathcal{A}$, we have $\exists i \in \{1, \cdots, N\}$ such that $l, l' \in \mathcal{L}^{\text{int},i}$. Therefore, the links of each antagonistic pair are assigned to a single subnetwork. This is important since subnetworks are controlled in a decentralized way and the controls for antagonistic pairs are directly coupled. Fifth, for all $l \in \mathcal{L}^i$, we have $\delta^i(l) = \delta(l) \cap \mathcal{L}^i$, where δ^i defines the downstream function of $\mathcal{N}^i$. Finally, the partitioning should lead to sparsity in the sense that the interconnections between subnetworks are minimal. As it will be explained later, while not affecting the satisfaction of the global specification (9), interconnections impose constraints that may introduce conservativeness into the planning of controls. Within all possible partitionings, want to choose the one for which the total number of interconnections is minimized:

$$\text{minimize } \frac{1}{2} \sum_i^N \sum_{j, i \neq j}^N |\mathcal{L}^i \cap \mathcal{L}^j|.$$

[2]Every MTL formula can be written in CNF.

The presence of constraints related to the specification makes our network partitioning problem different from the traditional graph partitioning problems in the literature [30]. For each internal link l of $\mathcal{N}$, we define N binary variables $b_l^i \in \{0, 1\}, i = 1, \cdots, N$, where $b_l^i = 1$ indicates l is assigned to the internal links of network $\mathcal{N}^i$. We note that links in $\mathcal{L}^{\text{out}}$ are excluded from the assignment process. We formulate the requirements that were explained above as the following integer constraints:

$$\begin{cases} \sum_{i=1}^N b_l^i = 1, \forall l \in \mathcal{L}^{\text{int}}, \\ \sum_{l \in \mathcal{L}} b_l^i \leq K, \\ b_l^i = b_{l'}^i, \forall (l, l') \in \mathcal{A}, \\ b_l^i = b_{l'}^i, \forall l, l' \in \text{Links}(\varphi_k^{conj}), k = 1, \cdots, n_\varphi, \end{cases} \tag{13}$$

where $i = 1, \cdots, N$. The first states that the sets of internal links are disjoint, the second ensures that the size of each subnetwork is bounded by K, the third reads that antagonistic links are assigned to the same subnetwork, and fourth declares that all the links in non-conjunctive sub-specifications are assigned to the same subnetwork.

It is easy to show that the total number of interconnections is equal to the summation of the differences in binary assignments of links that are related by δ:

$$\sum_i^N \sum_{j, i \neq j}^N |\mathcal{L}^i \cap \mathcal{L}^j| = \sum_{l \in \mathcal{L}, l' \in \delta(l)} \sum_i^N \sum_{j, i \neq j}^N |b_l^i - b_{l'}^j|. \tag{14}$$

The decisions for $\{b_l^i\}_{i=1,\cdots,N, l \in \mathcal{L}^{\text{int}}}$ are found by solving the following integer programming problem:

$$\begin{aligned} &\text{minimize} && \sum_{l \in \mathcal{L}, l' \in \delta(l)} \sum_i^N \sum_{j, i \neq j}^N |b_l^i - b_{l'}^j| \\ &\text{subject to} && (13). \end{aligned} \tag{15}$$

The complexity of integer programming problems grow exponentially with the number of variables. The choices for N, K are determined by the user. Usually, K is related to the computation power available for solving the MPC optimization problems in real time. We choose $N \geq 1$ as the minimum integer such that a feasible solution to (15) exists.

Example 4.1. The partitioning for the network in Fig. 1 with $N = 4, K = 20$, is illustrated in Fig. 2. Solving (15) with 524 binary variables took 0.65 seconds on a dual core 3GHz Macbook Pro using Gurobi [3]. The internal links of each subnetwork are shown with the same color.

Throughout this paper, we use "local" and "global" to refer to the attributes of a subnetwork and the original network, respectively. Once a network is partitioned, $\alpha^i, \beta^i, \mathcal{A}^i, i = 1, \cdots, N$, are constructed in the obvious way such that the values in α, β and the pairs in $\mathcal{A}$ are inherited. Note that an interconnection is essentially a local outgoing link for one subnetwork and a local internal link

[3]www.gurobi.com

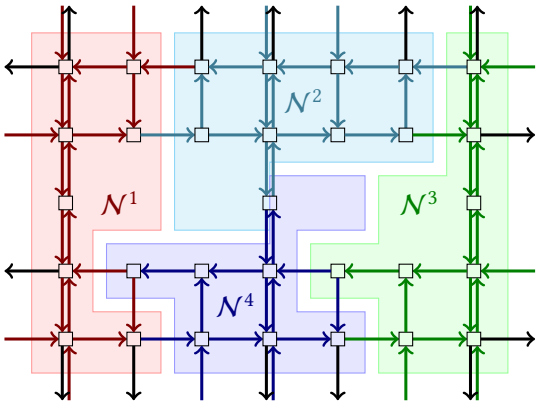

Figure 2: Network Partitioned into 4 Subnetworks

for the other. We define the local version of Φ for subnetwork $\mathcal{N}^i$ as:

$$\Phi^i := ((x_i, u_i) \in \pi^i) \wedge \bigwedge_{\text{Links}(\varphi_k^{conj}) \subseteq \mathcal{L}^{\text{int},i}} \varphi_k^{conj}, \tag{16}$$

where π^i is defined for subnetwork $\mathcal{N}^i$ in the same way as Definition 3.2. Note that the control of each link is assigned to a single subsystem. The control policy, the set of initial conditions and the run of subsystem i are denoted by μ^i, $\mathcal{X}_0^i$, and ζ^i, respectively.

For each $\mathcal{N}^i$, $i = 1, \cdots, N$, we define the sets of downstream and upstream subnetworks as $Down(\mathcal{N}^i) := \left\{\mathcal{N}^j | \exists l \in \mathcal{L}^{\text{int},i}, \delta(l) \in \mathcal{L}^{\text{int},j}\right\}$, $Up(\mathcal{N}^i) := \left\{\mathcal{N}^j | \exists l \in \mathcal{L}^{\text{int},j}, \delta(l) \in \mathcal{L}^{\text{int},i}\right\}$, respectively. Since most interconnections are two-way, often two subnetworks are both upstream and downstream of each other. Vehicles leaving a subnetwork can enter to one of its downstream subnetworks or leave the entire network. In other words, some components of the output of a subnetwork become additive disturbances for its downstream subnetwork. For all interconnections we have:

$$w^i(l,t) = w(l,t) + \sum_{\mathcal{N}^j \in Up(\mathcal{N}^i), l \in \mathcal{L}^{j,\text{out}}} y_{l,t}^j. \tag{17}$$

We write the system equations for each subnetwork as $x_{t+1}^i = F^i(x_t^i, u_t^i, w_t^i)$, $y_t^i = G^i(x_t^i, u_t^i)$.

5 CONTRACT SYNTHESIS

In this section, we explain how the assume-guarantee contracts between subnetworks in a partition are found and used for decentralized control synthesis, while satisfying the global specification (9). We use the fact that traffic dynamics are *monotone* in the congestion-free set and the satisfaction of (9) can be converted into a set-invariance problem in the trajectory space of length $h(\Phi)$.

5.1 Assume-Guarantee Contracts

For notation convenience, we denote the components of output of $\mathcal{N}^i$ that affect $\mathcal{N}^j$ as $y_t^{i\to j} := \{y_{l,t}^i\}_{l \in \mathcal{L}^{\text{out},i}, \mathcal{L}^{\text{int},j}}$. We also define $\xi^{i\to j} := y_0^{i\to j}, y_1^{i\to j}, \cdots$.

Definition 5.1. An *assume-guarantee contract* (AGC) $\psi^{i\to j}$ is an MTL formula, with atomic propositions as predicates over $y_t^{i\to j}$, such that:

- subnetwork $\mathcal{N}^j$ *assumes* that $\xi^{i\to j}$ (which acts on as disturbance) satisfies $\mathbf{G}_{[0,\infty)}\psi^{i\to j}$;
- subnetwork $\mathcal{N}^i$ *guarantees* that $\xi^{i\to j}$ (which is its output) satisfies $\mathbf{G}_{[0,\infty)}\psi^{i\to j}$.

The key idea in contract-based control design is that once AGCs are found such that local control polices exist to ensure them, then the local controllers can operate in a decentralized manner. We also require the following assumption for synchronization of time between subnetworks:

Assumption 3. *All subnetworks have access to a global clock.*

This assumption will be further discussed in Sec. 5.5. Given subnetworks $\mathcal{N}^1, \cdots, \mathcal{N}^N$, we design all AGCs such that

$$\bigwedge_{i=1}^{N} (\mathbf{G}_{[0,\infty)}(\Phi^i \wedge \bigwedge_{\mathcal{N}^j \in Down(\mathcal{N}^i)} \psi^{i\to j})) \Rightarrow \Phi^{global}. \tag{18}$$

Moreover, there should exist local control policies μ^i and a non-empty set of initial conditions $\mathcal{X}_0^i$, $i = 1, \cdots, N$, such that both assumptions and guarantees are met. That is to say for all allowable $\mathbf{w}^i$ that satisfy $\wedge_{\mathcal{N}^j \in Up(\mathcal{N}^i)} \psi^{j\to i}$ (assumptions), we have

$$\sigma_0^i(\zeta^i(x_0^i, \mu^i, \mathbf{w}^i)) \models \mathbf{G}_{[0,\infty)}(\Phi^i \wedge \bigwedge_{\mathcal{N}^j \in Down(\mathcal{N}^i)} \psi^{i\to j}), \tag{19}$$

for all $x_0^i \in \mathcal{X}_0^i$ (guarantees). Note that word σ^i is constructed from the local propositions including those of the local contracts.

As one can observe, the design of AGCs falls into circular reasoning for subnetworks that have two-way interconnections. Moreover, AGCs are often not unique. In this paper, we formulate the problem of contract synthesis as a single constraint satisfaction problem, which leads to feasibility check for a MILP problem. Although the complexity grows exponentially (in the worst case) with respect to network size, we show that the computation time is small for fairly large networks (e.g., the network in Fig. 1). Note that contracts are computed offline.

5.2 Monotonicity

Proposition 5.2. *System* (4) *is monotone with respect to the additive disturbances in the sense that* $\forall x \in \mathcal{X}, \forall u \in \mathcal{U}$, *we have* $F(x,u,w) \preceq_+ F(x,u,w^{\max}), \forall w \in \mathcal{W}$.

Proposition 5.3. *System* (4),(5) *is* monotone with respect to the state *in the congestion-free set, i.e. for all* $(x,u) \in \Pi$, $(x',u) \in \Pi$ *such that* $x \preceq_+ x'$, *we have* $F(x,u,w) \preceq_+ F(x',u,w), \forall w \in \mathcal{W}$, *and* $G(x,u) \preceq_+ G(x',u)$.

It is worth to note that monotonicity property is not valid under FIFO rule for congested flow in diverging intersections [6]. Monotonicity enables us to evaluate the worst-case trajectories by setting the disturbances to $w^{\max}$, a technique that we use frequently in this paper.

5.3 Language Realization Set

We define the extended state $s_t, t \in \mathbb{N}$ such that

$$s_t := \left((x_t, u_t), (x_{t+1}, u_{t+1}), \cdots, (x_{t+h(\Phi)-1}, u_{t+h(\Phi)-1})\right),$$

where $s_t \in \mathcal{S}, \mathcal{S} = \prod_{\tau=0}^{h(\Phi)}(\mathcal{X} \times \mathcal{U})$. For notation convenience, we define $X_{\mathcal{S}}$ and $U_{\mathcal{S}}$ such that $x_t = X_{\mathcal{S}}(s_t)$ and $u_t = U_{\mathcal{S}}(s_t)$. Using s_t and atomic propositions in Φ, one can obtain the observations in $\sigma_{[t,t+h(\Phi))}$. Therefore, we can check whether $\sigma_{[t,t+h(\Phi))}(s_t) \models \Phi$.

Definition 5.4. The *language realization set* (LRS) of an MTL formula Φ is defined as:

$$\text{LRS}(\Phi) := \left\{s_0 \in \mathcal{S} | \sigma_{[0:h(\Phi))}(s_0) \models \Phi\right\}. \tag{20}$$

In other words, $\text{LRS}(\Phi) \subseteq \mathcal{S}$ is the set of all $h(\Phi)$-length runs that generate suffixes satisfying Φ.

Proposition 5.5. *The word generated by $\zeta(x_0, \mu, w)$ satisfies $\mathbf{G}_{[0,\infty)}\Phi$ if and only if $s_t \in LRS(\Phi), \forall t \in \mathbb{N}$.*

The relation $\preceq_{\mathcal{S}}$ is defined such that $s \preceq_{\mathcal{S}} s'$ indicates 1) $x_k \preceq_+ x'_k$, and 2) $u_k = u'_k$, $k = 0, 1, \cdots, h(\Phi) - 1$.

Proposition 5.6. *The relation $\preceq_{\mathcal{S}}$ is a partial order.*

For the remainder of this section, we use partial order $\preceq_{\mathcal{S}}$. Restricting the state predicates to 7, the following can be stated.

Proposition 5.7. *$LRS(\Phi) \subset \mathcal{S}$ is a lower-set.*

5.4 Controlled Invariance

The evolution of the extended state s_t is written as:

$$s_{t+1} = F_{\mathcal{S}}(s_t, v_t, d_t), \tag{21}$$

where $v_t = u_{t+h(\Phi)}$ is the extended control, $d_t = w_{t+h(\Phi)}$ is the extended disturbance, and $F_{\mathcal{S}}(s_t, v_t, d_t)$ is:

$$\left((x_1, u_1), \cdots, (x_{(h(\Phi)}, u_{(h(\Phi)}), (F(x_{(h(\Phi)}, v_t, d_t)), v_t)\right).$$

Proposition 5.8. *System (21) is monotone with respect to the extended state in LRS(Φ).*

In order to satisfy $\mathbf{G}_{[0,\infty)}\Phi$, we need to compute a robust control invariant set (RCIS) [17] that lies entirely in $\text{LRS}(\Phi)$. Computation of an RCIS inside a non-convex set for a hybrid system is a computationally difficult problem. We exploit monotonicity to propose an alternative computational approach. The following theorem is an extension of the one in our previous work [28],[29]:

Theorem 5.9. *Consider a sequence of extended states $s_0^*, s_1^*, \cdots, s_{T-1}^*$, and a sequence of extended controls $v_0^*, v_1^*, \cdots, v_{T-1}^*$, such that 1) $s_{k+1}^* = F_{\mathcal{S}}(s_k, v_k, d_k^{\max})$, 2) $s_k^* \in LRS(\Phi)$, $k = 0, 1, \cdots, T-1$, 3) $s_T \preceq s_0^*$. Then $\Omega := \bigcup_{k=0}^{T-1} L(s_k^*)$ is an RCIS inside LRS(Φ).*

We refer to the obtained extended control sequence $v_0^*, v_1^*, \cdots, v_{T-1}^*$ as s-sequence, where T is its length. The main feature of the theorem above is that we can compute an RCIS without using the traditional fixed-point algorithm [17], which is computationally intractable for hybrid systems and does not guarantee termination in finite steps.

Piecewise affine systems can be transformed into a set of mixed-integer linear equations [13]. Temporal logic constraints characterizing $\text{LRS}(\Phi)$ can also be translated into mixed-integer constraints [25]. Finally, the terminal condition $s_T^* \preceq s_0^*$ is a linear constraint. The details of the procedures are not explained in this paper as they are well documented in the mentioned works. The conditions in Theorem 5.9 becomes equivalent to finding a feasible solution to a MILP problem. Note that even though the computational complexity of solving a MILP is NP-hard in general, finding a feasible solution is significantly faster than finding an optimal solution.

In order to compute an RCIS, we start from $T = 1$ and implement $T \leftarrow T + 1$ until the MILP for Theorem 5.9 becomes feasible. In [29], we showed that computing RCISs using a simplified version of Theorem 5.9 is almost complete. That is to say, as T becomes larger, a feasible solution should exist if there exists any non-empty RCIS. However, the same result does not hold here as we are also considering predicates over controls.

Example 5.10. We formulated the conditions in Theorem 5.9 for the network in Fig. 1. We have $\varphi = \bigwedge_{i=1}^{4} \varphi_i$, where φ_i's are given in Example 3.3. Note that $h(\varphi) = 6$. The smallest T for which a feasible solution exists is $T = 6$. The corresponding MILP has 5981 variables (1236 binary) and 2902 constraints. It takes 8.6 seconds on a dual core 3.0 GHz MacBook Pro to find a feasible solution and hence an RCIS in $\text{LRS}(\Phi)$, which lies in $\mathbb{R}_+^{420} \times \{0, 1\}^{384}$.

Proposition 5.11. *A control policy satisfying Φ^{global} starting from x_0 exists if $x_0 \in \mathcal{X}_0^{\Omega}$, where*

$$\mathcal{X}_0^{\Omega} := \left\{x_0 \in \mathcal{X} \middle| \exists i \in \{0, \cdots, T-1\}, x_0 \leq X_{\mathcal{S}}(s_i^*)\right\}. \tag{22}$$

The set of admissible initial conditions are characterized by the computed RCIS. In order to find the set of all admissible initial conditions, one has to compute the maximal RCIS, which is unfortunately not possible beyond few dimensions.

Proposition 5.12. *[28] If $v_0^*, v_1^*, \cdots, v_{T-1}^*$ is a s-sequence corresponding to $s_0^*, s_1^*, \cdots, s_{T-1}^*$, then for all initial conditions in $L(X_{\mathcal{S}}(s_0^*))$, applying the following open-loop control sequence:*

$$u^{open\text{-}loop} := \overline{v_0^*, v_1^*, \cdots, v_{T-1}^*} \tag{23}$$

guarantees satisfaction of Φ^{global} for all allowable $\mathbf{w}$.

Therefore, an open-loop solution to Problem 1 is obtained but without any optimality considerations.

5.5 Contracts

Now we explain how to extract contracts from a feasible solution for the constraints in Theorem 5.9. We denote s, s^* and $\mathcal{S}$ corresponding to subnetwork $\mathcal{N}^i$ by s^i, $s^{*,i}$ and $\mathcal{S}^i$, respectively.

Definition 5.13. The *global clock* is defined as $\mathbb{G} : \mathbb{N} \to \{0, 1, \cdots, T-1\}$, where $\mathbb{G}_t$ is its value at time t.

As mentioned earlier, we assume that all the local controllers have the knowledge of the global clock. The "natural" evolution of the global clock is such that if $\mathbb{G}_t = \tau$, then $\mathbb{G}_{t+1} = (\tau + 1) \bmod T$. However, in case local controllers share information, we allow the value of the global clock to be determined by the local controllers. The details are explained in Sec. 6.2.

Proposition 5.14. *Suppose the global clock value at time t is τ and the extended state is s_t, where $s_t \preceq_{\mathcal{S}} s_\tau^*$. Then we have $y_t^i \preceq_+ y_\tau^{*,i}$, where $y_\tau^{*,i} = G^i(x_\tau^{*,i}, u_\tau^{*,i})$, $x_\tau^{*,i} = X_{\mathcal{S}}(s_\tau^*)$, $u_\tau^{*,i} = U_{\mathcal{S}}(s_\tau^*)$.*

Therefore, we have

$$w^{i,\max}(l,\tau) = w^{\max}(l,\tau) + \sum_{\mathcal{N}^j \in Up(\mathcal{N}^i), l \in \mathcal{L}^{j,\text{out}}} y^{*,j}_{l,\tau}. \quad (24)$$

Proposition 5.15. *We have $s_t \preceq_{\mathcal{S}} s^*_\tau$ if and only if $s^i_t \preceq_{\mathcal{S}^i} s^{*,i}_\tau$, $i = 1, \cdots, N$.*

Considering these, the contracts are given as:

$$\psi^{i\to j} = \bigwedge_{\tau=0}^{T-1} \left((\tau = \mathbb{G}_t) \Rightarrow (y^{i\to j}_t \leq y^{*,i\to j}_\tau) \right). \quad (25)$$

This contract synthesis process ensures that there exists at least one local control policy for each subnetwork such that the contracts are maintained alongside with the local specifications. One such local policy is the open-loop control sequence in (23), which can be implemented in a decentralized way. Often other local control policies also exist from which an optimal one can be selected considering a cost function, as discussed in the next section.

6 CONTROL SYNTHESIS

In this section, we explain how to find controls optimally. The cost function in (10) can be written as:

$$J := \sum_{i=1}^{N} J^i,\ J^i(x^i_0, \mu^i, \mathbf{w}^i) = \sum_{t=0}^{\infty} \sum_{l \in \mathcal{L}^{i,\text{out}}} \gamma^t (x^i_{l,t} - f^i_{l,t}). \quad (26)$$

It is worth to note that when the system is in the congestion-free set, $f_{l,t}$ only depends on the state and control of link l. Thus the decomposition of the cost as (26) is valid. We wish to find an optimal control policy μ^i for each subnetwork $\mathcal{N}^i$ such that its local specification and contracts to downstream neighbors are satisfied:

$$\begin{array}{ll} \text{minimize} & \max_{\mathbf{w}^i} J^i((x^i_0, \mu^i, \mathbf{w}^i)) \\ \text{subject to} & \sigma^i_0(\zeta^i(x^i_0, \mu^i, \mathbf{w}^i)) \models \left(\mathbf{G}_{[0,\infty)} \Phi^i \wedge \right. \\ & \left. \bigwedge_{j \in Down(\mathcal{N}^i)} \psi^{i\to j} \right),\ \forall \mathbf{w}^i. \end{array}$$

We find controls optimally using a decentralized MPC approach. In this setting, local controllers do not exchange any information and the pre-designed contracts are the only global provision. Next, we explain how to extend the decentralizing framework into a simple *cooperative MPC* algorithm where the local controllers determine the value of the global clock by exchanging some information.

6.1 Decentralized Model Predictive Control

Given an MPC prediction horizon H, we denote $u^{i,H}_t := u^i_{0|t}, \cdots, u^i_{H|t}$. The length H is determined by the user but as explained later, we recommend $H > h(\Phi)$. Given $w^{i,H}_t = w^i_{0|t}, \cdots, w^i_{H-1|t}$, the H-length prediction of the system state and output are denoted by $x^{i,H}_t = x^i_{0|t}, \cdots, x^i_{H|t}$ and $y^{i,H}_t = y^i_{0|t}, \cdots, y^i_{H|t}$, respectively. At each time, we optimize $u^{i,H}_t$, implement $u^i_{0|t}$ and solve the optimization problem at next time. The global clock at time t is supposed to be a known value. The MPC optimization problem at time t is given as follows:

$$\begin{array}{lll} u^{i,H}_t = & \text{argmin} & \sum_{k=0}^{H} \sum_{l \in \mathcal{L}^{i,\text{out}}} \gamma^k (x^i_{l,k|t} - f^i_{l,k|t}) \\ & \text{subject to} & s^i_{k-h(\Phi)|t} \in \text{LRS}(\Phi^i) \\ & & \sigma_t(\zeta^{i\to j}_{k|t}) \models \mathbf{G}_{[t,t+H]} \psi^{i\to j}, j \in Down(\mathcal{N}^i), \\ & & s^i_{H-h(\Phi)|t} \in L(s^*_{\tau+H-h(\Phi)}), \\ & & x^i_{k+1|t} = F(x^i_{k|t}, u^i_{k|t}, w^{i,\max}_{k|t}), \\ & & \mathbb{G}_{t+k} = (\tau + k) \bmod T, k = 0, \cdots, H. \end{array} \quad (27)$$

There are five lines of constraints that are explained as follows. The first is indicating that the all the H-length predictions of the extended states are in the language realization set, hence the specification is not violated in finite time. Due to the MTL temporal operators, the time window of constraints is shifted by $h(\Phi)$ [27]. It is worth to note that we require the knowledge of the recent $h(\Phi)$-length history of the controls and states. For time $t = 0$, we assume that all the previous propositions from this history are true [27]. The second line stands for the constraints of the contracts for downstream subnetworks. The third line states that the last predicted extended state has to lie inside the projection of RCIS Ω on $\mathcal{S}^i$, which provides a sufficient condition for establishing recursive feasibility (see Theorem 6.1 below). Since the partial ordering $\preceq_{\mathcal{S}^i}$ has equality constraints on controls, the length of the MPC horizon with free decision variables is $H - h(\Phi)$. Therefore, we need $H > h(\Phi)$. Otherwise, there exists only one feasible solution to (27). The fourth line is stating that the predictions are computed using the largest values of disturbances, which due to monotonicity, corresponds to the worst-case scenario. Thus, the MPC algorithm is *robust* in the sense that all constraints are satisfied for all allowable disturbances. The fifth line stands for the natural evolution of the global clock and the range of indices of predictions.

Theorem 6.1. *The MPC optimization problem* (27) *is recursively feasible in the sense that if it is feasible at time t and a valid control decision is implemented, then it is guaranteed to be feasible at time $t + 1$.*

Proposition 6.2. *The global specification* (9) *is satisfied if* (27) *is recursively feasible for all subnetworks.*

It should be noted that the solutions obtained from MPC are optimal only over the finite prediction horizon. Therefore, solutions can be suboptimal compared to the global optimum in (11). We also note that contracts can introduce conservativeness since subnetworks assume maximum allowable disturbances from the upstream neighbors. To mitigate this conservativeness, we desire that the number of subnetworks - and interconnections - be as small as possible. A very long prediction horizon can also cause conservativeness since the worst-case values at far future times are considered in the optimization problem. This issue can be alleviated by using an appropriate discount factor in the cost function. In the case when the state knowledge is noisy, the values for states in (27) have to be replaced by their upper-bound estimates. Due to monotonicity, this ensures that the correctness of the specification and the contracts are maintained for all possible state values. However, one may get infeasibility for (27) because of unrealistic state

measurements, even though feasibility is guaranteed for the true state values. In this case, one has to relax the constraints. A less conservative treatment of noisy data may require a probabilistic framework, which is out of the scope of this paper.

6.2 Cooperative Model Predictive Control

Here we assume that the local controllers are able to communicate with each other. In order to improve optimality, we introduce a simple modification for the decentralized MPC algorithm as follows. At each time, the controllers implement (27) for all allowable values of the global clock. Therefore, we define $J^{opt,i} = \{J_\tau^{opt,i}\}_{\tau\in\{0,\cdots,T-1\}}$, $J^{opt,i} \in \mathbb{R}_+^T$, where $J_\tau^{opt,i}$ is the optimal cost of MPC for subnetwork $\mathcal{N}^i$, $i = 1,\cdots,N$, obtained from setting the clock variable $\mathbb{G}_t$ to τ. If an MPC optimization problem is infeasible, we set its cost to ∞. Next, we choose the *best* global clock value as follows:

$$\tau^* = \min_{\tau\in\{0,\cdots,T\}} \sum_{i=1}^{N} J_\tau^{opt,i}. \tag{28}$$

Note that recursive feasibility guarantees that for at least one clock variable the sum of costs is finite. Distributed computation of (28) can be accomplished using a distributed average consensus algorithm [8], assuming that the controllers of the subnetworks communicate on a connected graph. Then, each controller implements the controls which correspond to the minimum cost in the average $\frac{1}{N}\sum_{i=1}^{N} J^{opt,i}$. We note that our cooperative MPC technique is still preliminary and there are many open directions to improve this approach.

7 SIMULATION RESULTS

In this section, we present numerical results of applying our methods to the network shown in Fig. 1. Motivated by time scales of real traffic networks, we consider a time step of 20 seconds in all simulations.

7.1 Macroscopic Simulation

We used the model in (4), which is "macroscopic" in the sense that it describes the aggregated vehicular dynamics instead of modeling each vehicle. The specification is given as in Example 5.10. For all MPC algorithms, we use $H = 11$ and $\gamma = 0.5$. We simulate the system for 45 time steps. The total delay accumulated over the network during this time frame is $\sum_{t=0}^{45}\sum_{l\in\mathcal{L}^{\text{int}}}(x_{l,t} - f_{l,t})$. All the computation times are given for implementations on a dual-core 3.0GHz Macbook Pro. The software for implementations are available in [1] for download. In the following implementations, the values for the exogenous demand for the network are drawn from a uniform distribution over $L(w^{\max})$, with the exception of links from subnetwork $\mathcal{N}^1$, where we set their exogenous demands to their largest admissible values.

Open-loop (OL). We implemented the control sequence from (23). This control policy is not traffic-responsive but ensures the satisfaction of the global specification (9). The total accumulated delay was 4786 [vehicles×time-step]. The implementation does not require any online computation effort or measurement of the state.

Centralized MPC (CeMPC). We implemented the MPC algorithm (27) for the complete network $\mathcal{N}$. We do not need the contract constraints as the network is undivided in this setup. Therefore, there is no conservativeness induced by the contracts. The accumulated delay was 2878, which indicates about 40% decrease compared to the OL policy. However, the computation time for each time step is very large (see Table 1), which indicates that CeMPC is not suitable for real time traffic management.

Table 1: Computation Time per Time Step and Accumulated Delay for Different Control Policies

	OL	CeMPC	DeMPC	CoMPC
Max. Comp. Time (s)	-	1762	0.93	7.09
Avg. Comp. Time (s)	-	86.7	0.17	1.21
Accumulated Delay	4786	2878	3216	3112

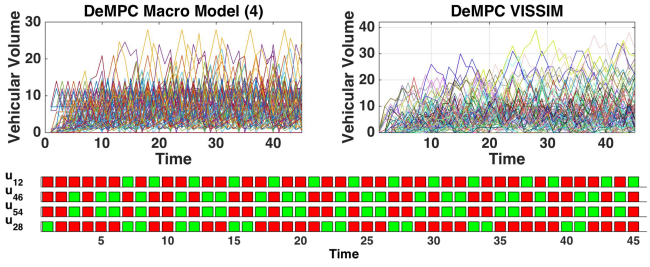

Figure 3: Simulation Results

Decentralized MPC (DeMPC). Here we implemented the MPC algorithm (27) for each subnetwork individually. The accumulated delay was 3216, which is a bit larger than the one for the CeMPC but still significantly smaller than the one for the OL policy. The computation times were less than a second (see Table 1). Therefore, DeMPC is appropriate for real time traffic management.

Cooperative MPC (CoMPC). Here we implemented the method from Sec. 6.2. During the simulation, the natural evolution of the global clock was overridden for 7 times, mainly in order to prioritize the heavy traffic in subnetwork $\mathcal{N}^1$. The accumulated delay was 3112, which is slightly smaller than the one for DeMPC. The computation times are longer due to solving multiple MILPs ($T = 6$ in this case), but the computations can be performed in parallel.

It is observed that the specification is satisfied by each implementation (which is also implied by the the fact that all the MPC problems were feasible). The trajectories always remain in the congestion-free set and all the sub-specifications in Example 3.3 are always met. For instance, the traffic lights corresponding to the sub-specifications φ_1, φ_2, and the vehicular volumes over time are shown in Fig. 3 (using DeMPC with all exogenous demands set to their maximum). It is also observed that the number of vehicles on the eastern bridge never exceeds 100 (sub-specification φ_3). For the only case when the volume on link 73 exceeded 5, its traffic light turned green immediately (sub-specification φ_4).

7.2 Microscopic Simulation

Here we show some preliminary results on implementing our methods on microscopic traffic simulators. We used the PTV VISSIM[4]

[4] http://vision-traffic.ptvgroup.com/products/ptv-vissim

microscopic traffic simulator, which is a widely used, highly realistic simulator that incorporates many aspects of traffic such as driver models, vehicle classes, priorities, conflicts, etc. A screenshot of the VISSIM implementation of the traffic network in Fig. 1 is shown in Fig. 4. The simulator was used in a closed-loop setup: the controller has access to the traffic volumes and the traffic light settings via a MATLAB interface. The length of each link is set to 300 meters. The maximal speed of the vehicles was set to 15 m/s with an acceleration profile of a normal passenger car. The sample time is 20 seconds. We simulate the system for 15 minutes (equivalent to 45 time steps). We observe that a naive (centralized) MPC controller with no feasibility guarantees leads to congestion in the network after about 25 time steps. However, using the methods in this paper we were able to to avoid congestion and satisfy all the specifications. The results are shown in Fig. 3. The video of our VISSIM implementation is included in [1].

The macroscopic model (4) used for control synthesis does not necessarily capture all the behaviors in the VISSIM model. Therefore, we are unable to formally guarantee that the specification is always satisfied by the VISSIM model. However, in our simulations we observed that the VISSIM simulations always satisfy the specification using the controls we found for (4). We leave further investigation of the relation between macroscopic and microscopic models from a formal methods perspective to our future work.

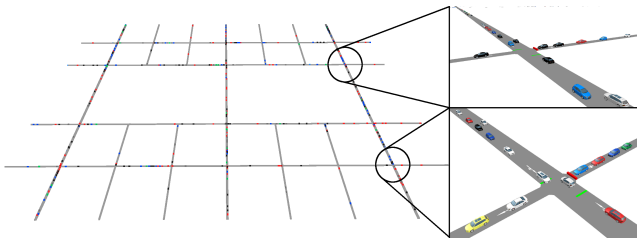

Figure 4: VISSIM Implementation

ACKNOWLEDGEMENTS

This work was partially supported by the NSF under grants CPS-1446151 and CMMI-1400167. János Rudan gratefully acknowledges the support of the Rosztoczy Foundation.

REFERENCES

[1] http://sites.bu.edu/hyness/format-distributed. (03/02/2017).
[2] Rajeev Alur, Salar Moarref, and Ufuk Topcu. 2016. Compositional synthesis with parametric reactive controllers. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. ACM, 215–224.
[3] Christel Baier, Joost-Pieter Katoen, and Kim Guldstrand Larsen. 2008. *Principles of model checking*. MIT press.
[4] Giacomo Como, Enrico Lovisari, and Ketan Savla. 2016. Convexity and Robustness of Dynamic Network Traffic Assignment for Control of Freeway Networks. *IFAC-PapersOnLine* 49, 3 (2016), 335–340.
[5] S Coogan and M Arcak. 2015. Efficient finite abstraction of mixed monotone systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control, 58-67, 2015*. ACM, 58–67.
[6] Samuel Coogan, Murat Arcak, and Alexander A Kurzhanskiy. 2016. Mixed monotonicity of partial first-in-first-out traffic flow models. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 7611–7616.
[7] S. Coogan, E. A. Gol, M. Arcak, and C. Belta. 2016. Traffic Network Control From Temporal Logic Specifications. *IEEE Transactions on Control of Network Systems* 3, 2 (June 2016), 162–172. DOI: http://dx.doi.org/10.1109/TCNS.2015.2428471
[8] Jorge Cortés. 2006. Finite-time convergent gradient flows with applications to network consensus. *Automatica* 42, 11 (2006), 1993–2000.
[9] Brian A Davey and Hilary A Priestley. 2002. *Introduction to lattices and order*. Cambridge university press.
[10] Adel Dokhanchi, Bardh Hoxha, and Georgios Fainekos. 2014. On-Line Monitoring for Temporal Logic Robustness. In *International Conference on Runtime Verification*. Springer, 231–246.
[11] Nathan H Gartner. 1983. OPAC: A Demand-Responsive Strategy for Traffic Signal Control. *Transportation Research Record* 906 (1983), 75–81.
[12] Jean Gregoire, Xiangjun Qian, Emilio Frazzoli, Arnaud De La Fortelle, and Tichakorn Wongpiromsarn. 2015. Capacity-aware backpressure traffic signal control. *IEEE Transactions on Control of Network Systems* 2, 2 (2015), 164–173.
[13] Wpmh Heemels, B De Schutter, and Alberto Bemporad. 2001. Equivalence of hybrid dynamical models. *Automatica* 37, 7 (2001), 1085–1091. DOI: http://dx.doi.org/10.1016/S0005-1098(01)00059-0
[14] Andreas Hegyi, Bart De Schutter, and Hans Hellendoorn. 2005. Model predictive control for optimal coordination of ramp metering and variable speed limits. *Transportation Research Part C: Emerging Technologies* 13, 3 (2005), 185–209.
[15] Thomas A Henzinger, Shaz Qadeer, and Sriram K Rajamani. 1998. You assume, we guarantee: Methodology and case studies. In *International Conference on Computer Aided Verification*. Springer, 440–451.
[16] PB Hunt, DI Robertson, RD Bretherton, and RI Winton. 1981. *SCOOT-a traffic responsive method of coordinating signals*. Technical Report.
[17] Eric C Kerrigan. 2000. *Robust Constraint Satisfaction: Invariant Sets and Predictive Control*. Ph.D. Dissertation. University of Cambridge.
[18] Eric S Kim, Murat Arcak, and Sanjit A Seshia. 2015. Compositional controller synthesis for vehicular traffic networks. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. 6165–6171. DOI: http://dx.doi.org/10.1109/CDC.2015.7403189
[19] Eric S Kim, Murat Arcak, and Sanjit A Seshia. 2016. Directed Specifications and Assumption Mining for Monotone Dynamical Systems. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. ACM, 21–30.
[20] Ron Koymans. 1990. Specifying real-time properties with metric temporal logic. *Real-Time Systems* 2, 4 (1990), 255–299. DOI: http://dx.doi.org/10.1007/BF01995674
[21] Vito Mauro and C Di Taranto. 1990. Utopia. *Control, computers, communications in transportation* (1990).
[22] Pitu Mirchandani and Larry Head. 2001. A real-time traffic signal control system: Architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies* 9, 6 (2001), 415–432. DOI: http://dx.doi.org/10.1016/S0968-090X(00)00047-4
[23] P. Nilsson and N. Ozay. 2016. Synthesis of separable controlled invariant sets for modular local control design. In *2016 American Control Conference (ACC)*. 5656–5663. DOI: http://dx.doi.org/10.1109/ACC.2016.7526557
[24] Markos Papageorgiou, C Diakaki, V Dinopoulou, A Kotsialos, and Yibing Wang. 2003. Review of road traffic control strategies. *Proc. IEEE* 91, 12 (2003), 2043–2067. DOI: http://dx.doi.org/10.1109/JPROC.2003.819610
[25] Vasumathi Raman, Alexandre Donzé, Mehdi Maasoumy, Richard M Murray, Alberto Sangiovanni-Vincentelli, and Sanjit A Seshia. 2014. Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control*. IEEE, 81–87.
[26] Matthias Rungger and Majid Zamani. 2015. Compositional construction of approximate abstractions. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*. ACM, 68–77.
[27] S. Sadraddini and C. Belta. 2015. Robust temporal logic model predictive control. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 772–779. DOI: http://dx.doi.org/10.1109/ALLERTON.2015.7447084
[28] S. Sadraddini and C. Belta. 2016. Safety control of monotone systems with bounded uncertainties. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. 4874–4879. DOI: http://dx.doi.org/10.1109/CDC.2016.7799014
[29] Sadra Sadraddini and Calin Belta. 2017. Formal Synthesis of Control Strategies for Positive Monotone Systems. (2017). arXiv:arXiv:1702.08501
[30] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence* 22, 8 (2000), 888–905.
[31] Paulo Tabuada and George J. Pappas. 2006. Linear time logic control of discrete-time linear systems. *IEEE Trans. Automat. Control* 51, 12 (2006), 1862–1877. DOI: http://dx.doi.org/10.1109/TAC.2006.886494
[32] Boyan Yordanov, Jana Tumova, Ivana Cerna, Jiří Barnat, and Calin Belta. 2012. Temporal Logic Control of Discrete-Time Piecewise Affine Systems. *IEEE Trans. Automat. Control* 57, 6 (2012), 1491–1504. DOI: http://dx.doi.org/10.1109/TAC.2011.2178328