

# Joint Perimeter and Signal Control of Urban Traffic via Network Utility Maximization

Negar Mehr<sup>1</sup>, Jennie Lioris<sup>2</sup>, Roberto Horowitz<sup>1</sup>, and Ramtin Pedarsani<sup>3</sup>

**Abstract**—With the ongoing rise of demand in traffic networks, congestion control has become of major importance for urban areas. In this paper, we introduce the notion of network utility maximization for boundary flow control of urban networks. We describe how maximizing the aggregate utility of the network leads to a fair allocation of network resources to different arrivals while maintaining system stability. We demonstrate how utility maximization problem can be solved using Alternating Direction Method of Multipliers (ADMM). We further show how our algorithm can be partially distributed such that each entry link finds its arrival for maximizing its own objective while maximizing the total utility of the network. We showcase the performance of our algorithm in an example illustrating fast convergence of our method and its capability in stabilizing the network.

## I. INTRODUCTION

The ongoing rise of vehicular traffic congestion in urban and metropolitan areas imposes significant costs such as fuel consumption and delay increases on transportation systems and cities. Due to these negative consequences, the task of controlling and improving the functionality of transportation networks is of great importance. The focus of this work is on optimizing the performance of an existing network of signalized intersections via scalable traffic control algorithms to increase the efficiency of the current available roads.

Since signal control is the main control input available for affecting urban traffic patterns, a wide range of signal control strategies have been proposed. The simplest form of signal control is fixed-time control where each light operates cyclically, and each phase receives a fixed amount of green splits during the cycle. Various tools such as SYNCHRO [1], VISGAOST [2], SCOOT [3], and OPAC [4] have been proposed for determining the timing plan of fixed-time controllers. SYNCHRO and VISGAOST use historical data for offline determination of timing plans. In SCOOT and OPAC, each intersection decides on its timing plan to optimize a performance measure of its upstream queues, neglecting the effect of timing plans on the downstream queues. A queueing-theoretic analysis of fixed-time control policies is conducted in [5].

In [6], Max Pressure (MP) control is presented, which is a distributed control scheme that provably maximizes the

network throughput and stabilizes the network in the presence of *feasible* arrivals. Using MP control, each intersection selects a stage of actuation that depends only on the length of adjacent queues. Due to nonlinearities and complexities of transportation networks, model predictive control laws are also shown to be successful in reducing total travel time and delays in both urban networks and freeways [7], [8], [9], [10]. Recently, synthesis from temporal logic specification has also been utilized for signal control [11], [12], [13], [14]. In such methods, the assumption is that the desired properties of the system can be encoded as formal specifications. Therefore, the control is found such that the temporal properties of interest are satisfied by the system trajectories.

Despite the effectiveness of the aforementioned control strategies, they are beneficial mostly in the regime of feasible arrivals. In fact, when the arrivals are not in the feasible region, regardless of the type of control deployed inside the network, the network is destabilized. In order to address this problem, TUC [15], which is a traffic responsive control strategy, was proposed for handling the saturated traffic conditions. In [15], the highly nonlinear dynamics of urban roads are simplified to a linear system, and the feedback gains obtained from solving an infinite horizon linear quadratic regulator are implemented to the system. In [16], traffic responsive control is developed for heterogeneous networks via perimeter control, where the amount of the boundary flow between different urban regions is determined using a PI controller. The authors in [16] model the traffic evolution in each region through Macroscopic Fundamental Diagrams (MFDs).

In this paper, we define a novel methodology for joint perimeter control and signal control of a single network for the case of infeasible arrivals. We consider a network with oversaturated arrivals and determine the timing plans and the amount of arrivals allowed to enter the network such that the network remains stable and free of congestion, the network utility is maximized, and different arrivals are treated *fairly*. Our approach is different from [16] as we consider perimeter control for a *single* network; thus, we do not require MFDs and partitioning the network. For a single network, we synthesize a joint congestion and signal control policy, and find the *optimal* boundary flows.

We adopt PointQ [6] as our urban traffic model, and use the notion of utility maximization which is a well known congestion control scheme in communication networks [17], [18] for our control problem. We form an optimization problem that maximizes the aggregate utility of the network. Moreover, we demonstrate that by constructing the augmented

<sup>1</sup>N. Mehr and R. Horowitz are with the Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA email: negar.mehr@berkeley.edu, horowitz@berkeley.edu

<sup>2</sup>J. Lioris is with the ENPC ParisTech, Paris, France jennie.lioris@cermics.enpc.fr

<sup>3</sup>R. Pedarsani is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA email: ramtin@ece.ucsb.edu

Lagrangian and using Alternating Direction Method of Multipliers (ADMM) [19], the optimization problem can be solved iteratively such that the update step of the arrivals can be distributed while guaranteeing that the network queues will remain stable. Since our iterative control utilizes ADMM for distributing its computations, it converges much faster than that of the previous work in [20] where dual decomposition is used. This fast convergence is crucial since for physical systems such as transportation networks, we do require to stabilize the system in the minimum possible number of time steps.

A unique and important feature of our work is that it allows us to introduce a notion of *fairness* among arrivals. Fairness is of paramount practical importance as vehicles in all network links must finally get the right of accessing the network regardless of where in the network they arrive. However, to the best of our knowledge, fairness has not been considered in the literature of traffic control except for our previous work, where we utilized utility maximization for fair control of freeway arrivals [21].

The organization of this paper is as follows. In Section II, we illustrate the notation that is adopted throughout the paper. In Section III, we describe the modeling framework we have used. We present our control algorithm in Section IV, and demonstrate the effectiveness of our method in an example in Section V. Finally, we conclude the paper and describe the future directions in Section VI.

## II. NOTATION

We use the following notation in this paper.  $\mathbb{R}_+^n = \{x \in \mathbb{R}^n \mid x_i \geq 0, \forall 1 \leq i \leq n\}$  is the set of  $n$  dimensional vectors with non-negative elements. For a vector  $x \in \mathbb{R}^n$ ,  $x_i$  is the  $i$ th element of  $x$ . To distinguish matrices from vectors, matrices are depicted with upper case letters. We denote the identity matrix by  $I$ . For a matrix  $A$ ,  $A^T$  is the matrix transpose, and  $A_{ij}$  is the  $ij$ th element of  $A$ . Moreover, for a vector  $x$ , the inequalities are interpreted element-wise unless otherwise mentioned, i.e.,  $x \leq b$  implies that  $x_i \leq b_i, \forall 1 \leq i \leq n$ . We denote a sequence of variables indexed by integer times as  $x^k$  for  $k = 1, 2, \dots$ .

## III. POINTQ MODEL

The PointQ model was first presented in [6], where the evolution of the network is modeled as a controlled store-and-forward queueing network. A network graph is constructed from  $N$  nodes which represent the network intersections and  $L$  directed links. Let  $\mathcal{N}$  and  $\mathcal{L}$  be the set of network nodes and links respectively. Links are divided into three types: entry links  $\mathcal{L}_{\text{entry}}$ , internal links  $\mathcal{L}_{\text{internal}}$ , and exits links  $\mathcal{L}_{\text{exit}}$ . Entry links are the ones that have no starting node and carry the exogenous arrivals to the network, internal links are the ones that connect intersections, and exit links are the links with no end node. For each link  $l$ , we define  $f_l$  to be the flow of vehicles on link  $l$  (vehicles per time step). Moreover, for each entry link  $l$ , we let  $\lambda_l$  be the exogenous flow of vehicles (arrivals) to the link  $l$ .

In PointQ, for each turn movement at an intersection, there exists a separate queue. In other words, queues are defined by the allowed movements at intersections. As a result of this equivalence, queues and movements are used interchangeably in this paper. For each movement from link  $l$  to  $m$ ,  $f^{l,m}$  is the flow of vehicles for this movement, and  $r^{l,m}$  is the fraction of vehicles that leave link  $l$  to move to link  $m$ . Consequently, link and movement flows must satisfy:

$$\begin{aligned} f_l &= \lambda_l, & f^{l,m} &= r^{l,m} f_l, & l &\in \mathcal{L}_{\text{entry}} & (1) \\ f_m &= \sum_{l \in \mathcal{L}} f^{l,m}, & f^{m,o} &= r^{m,o} f_m, & m &\in \mathcal{L}_{\text{internal}} \cup \mathcal{L}_{\text{exit}}. & (2) \end{aligned}$$

In addition to the previously defined quantities, for each movement from link  $l$  to link  $m$ , we define  $\mu^{l,m}$  to be the saturation flow rate of this movement, which is defined as the maximum flow allowed for this movement.

### A. Stages and Simultaneous Movements of Fixed-time Control

Assume that the network intersections are controlled by fixed-time controllers with a common cycle time  $T$  for all intersections. Then, for each node  $n$ , a cycle is divided into  $S_n$  stages. Each stage is a set of movements that are actuated simultaneously during a cycle. In each stage  $j, 1 \leq j \leq S_n$ , an arbitrary movement from link  $l$  to  $m$  receives  $g_j^{l,m}$  fraction of green time, meaning that this movement receives green signal for  $g_j^{l,m}T$  seconds during stage  $j$  at its corresponding intersection.

**Remark 1.** *In each stage  $j$ , it is normally the case that multiple movements are actuated. Let the movements from links  $l$  and  $u$  to links  $m$  and  $v$  be such movements. In such cases, it holds that  $g_j^{l,m} = g_j^{u,v}$ .*

In addition to the requirements imposed by simultaneous actuation of certain movements during signal stages stated in Remark (1), the sum of green ratios of all stages adds up to 1 for each intersection  $n$ . Mathematically, for node  $n$ , if  $d_j$  is the number of the queues that are simultaneously actuated in stage  $j$ , we have

$$\sum_{j=1}^{S_n} \frac{g_j^{l,m}}{d_j} = 1. \quad (3)$$

It is worth mentioning that if one wants to consider the clearance time between stages, Equation (3) must be modified such that sum of the green ratios adds up to  $1 - \epsilon_n$ , where  $\epsilon_n$  is the proportion of the cycle time when the signal is ‘‘all red’’ at intersection  $n$ .

**Remark 2.** *Since there might exist multiple actuations of a queue during several stages of a node  $n$ , we use  $p^{l,m} = \sum_j g_j^{l,m}$  as the aggregate green ratio of this movement during one cycle time.*

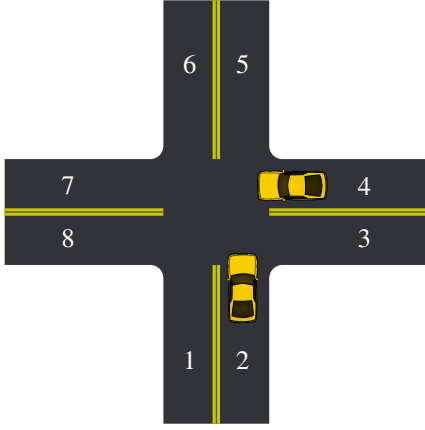


Fig. 1: Schematic of an intersection.

In order for our queueing network to be stable, one must ensure that the flow of each link is smaller than or equal to the total service received by that link (per cycle) i.e.,

$$f^{l,m} \leq \mu^{l,m} p^{l,m} \quad \forall l, m \in \mathcal{L}. \quad (4)$$

**Example:** Consider the schematic intersection demonstrated in Figure 1. We have 8 links, where 2, 4, 6, and 8 are entry links, and 1, 3, 5 and 7 are exit links. There is no internal link in this example. Assume that there exist only through and right movements. The origin-destination pairs for all network queues are: (2,5), (4,7), (8,3), (6,1), (2,3), (4,5), (6,7) and (8,1). Thus, we have a total of 8 queues. Assume that the intersection has 2 stages, each of which lasts half of the cycle time. During each stage, the following movements are actuated:

**First Stage:** (2,5), (4,5), (2,3), (6,1), (8,1), and (6,7).

**Second Stage:** (4,7), (6,7), (4,5), (8,3), (8,1), and (2,3).

Therefore, for the first stage, we have:

$$g_1^{2,5} = g_1^{4,5} = g_1^{2,3} = g_1^{6,1} = g_1^{8,1} = g_1^{6,7} = 0.5.$$

and for the second stage, we have:

$$g_2^{4,7} = g_2^{6,7} = g_2^{4,5} = g_2^{8,3} = g_2^{8,1} = g_2^{2,3} = 0.5.$$

As we expect, having  $d_1 = d_2 = 6$ , the green ratios of all stages add up to 1:

$$\begin{aligned} & \frac{1}{6}(g_1^{2,5} + g_1^{4,5} + g_1^{2,3} + g_1^{6,1} + g_1^{8,1} + g_1^{6,7}) + \\ & \frac{1}{6}(g_1^{4,7} + g_1^{6,7} + g_1^{4,5} + g_1^{8,3} + g_1^{8,1} + g_1^{2,3}) = 1. \end{aligned}$$

The aggregate green ratio of movements is defined accordingly:

$$\begin{aligned} p^{2,5} &= p^{6,1} = p^{4,7} = p^{8,3} = 0.5, \\ p^{4,5} &= p^{2,3} = p^{8,1} = p^{6,7} = 1. \end{aligned}$$

### B. Compact Notation of the PointQ Model

To make the notation compact, let  $\lambda$  and  $f \in \mathbb{R}_+^L$  be the vector of link arrivals and flows, respectively, with  $\lambda_l$  being the arrival rate of link  $l$  if  $l \in \mathcal{L}_{\text{entry}}$  and 0 otherwise. We can also

collect turn ratios  $r^{l,m}$  into a matrix  $R$  such that  $R_{lm} = r^{l,m}$ . Using Equations (1) and (2), it is easy to verify that link flows can be obtained as follows:

$$f = (I - R^T)^{-1} \lambda. \quad (5)$$

See [6] for the proof of (5).

Furthermore, assuming that we have a total of  $Q$  possible movements or queues in the network, we use  $\varphi$  and  $p \in \mathbb{R}_+^Q$  as the vectors of movement flows and aggregate green ratios. In other words,  $\varphi$  and  $p$  are the vectors constructed by the collection of  $f^{l,m}$  and  $p^{l,m}$  for all  $l$  and  $m$ , for which there exists a possible movement. Alternatively,  $\varphi$  and  $p$  denote the vector of nominal rates of the queues and their allocated fraction of service at each cycle. Bringing movement service rates together in a diagonal matrix  $M$  such that its  $k_{\text{th}}$  diagonal entry is equal to the service rate of the  $k_{\text{th}}$  movement, stability condition of Equation (4) can be written as:

$$\varphi \leq Mp. \quad (6)$$

Note that using Equations (1) and (2), one can observe that the elements of  $\varphi$  and  $f$  are mapped through the turn ratios. Hence, one has

$$\varphi = \Gamma f, \quad (7)$$

where  $\Gamma_{Q \times L}$  is a constant matrix such that  $\varphi_k = \gamma_k f = r^{l,m} f_l$  with  $l$  and  $m$  being the origin and destination links of the  $k_{\text{th}}$  queue, and  $\gamma_k$  being the  $k_{\text{th}}$  row of matrix  $\Gamma$ . Deploying Equations (5) and (7), the stability condition in Equation (4) can be rewritten as:

$$\Gamma(I - R^T)^{-1} \lambda \leq Mp. \quad (8)$$

We can also stack stage green ratios  $g_j^{l,m}$  of all network queues at all stages in a vector  $g \in \mathbb{R}_+^K$  where  $K$  is the total number of stage green ratios for all queues in the network. Since  $p^{l,m} = \sum_j g_j^{l,m}$ , the mapping between  $g$  and  $p$  can be written as:

$$A_{g \rightarrow p} g = p, \quad (9)$$

with  $A_{g \rightarrow p}$  being a matrix of appropriate dimensions for capturing  $p^{l,m} = \sum_j g_j^{l,m}$ . Moreover, using our vectorized notation and Remark 1, we can encode equality of stage actuation times for simultaneous actuation of queues in a single stage by the following equality:

$$A_{\text{eq}} g = 0, \quad (10)$$

where each row of  $A_{\text{eq}}$  captures equality of two stage actuation times. Also, we rewrite equation Equation (3) as:

$$A_{\text{sum}} g = 1_{N \times 1}. \quad (11)$$

For simplicity, from now on, we omit the superscript describing the origin-destination links of a queue. Additionally, we summarize Equations (9), (10) and (11), dictating the constraints imposed by the control requirements, via the following linear equality constraints:

$$A_c g = b_c. \quad (12)$$

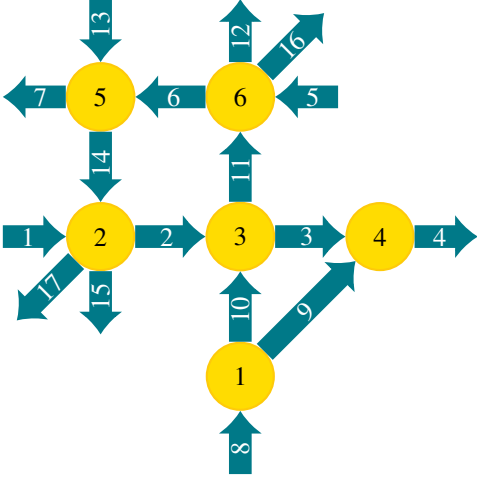


Fig. 2: The topology of the exemplar network.

The matrix  $A_c$  in Equation (12) is simply attained by augmenting  $A_{g \rightarrow p}$ ,  $A_{eq}$  and  $A_{sum}$  whilst  $b_c$  is an augmented vector of appropriate dimensions corresponding to the right hand side of Equations (9), (10) and (11).

#### IV. CONTROL ALGORITHM

##### A. Optimization Formulation

As mentioned previously, at a high level, we aim to maximize the amount of flow allowed to enter the network, while network stability is preserved, and arrivals are treated fairly. To this end, we propose to maximize the total utility of network arrivals subject to the stability condition (4). In particular, we wish to solve the following optimization problem:

$$\begin{aligned} & \underset{\lambda, g}{\text{maximize}} && \sum_{l \in \mathcal{L}_{\text{entry}}} U(\lambda_l) \\ & \text{subject to} && \Gamma(I - R^T)^{-1} \lambda \leq M A_{g \rightarrow p} g. \end{aligned} \quad (13)$$

The utility function,  $U(\cdot)$  in Equation (13) is a strictly concave increasing function of arrival rate  $\lambda_l$ . Examples of such utility functions include  $\log(x)$  and  $x^a$  for  $a < 1$ . Such functions have been extensively used for incorporating the notion of fairness among arrivals in communication networks [22]. The constraints in optimization problem (13) guarantee that the system stability conditions are satisfied. Rather than directly imposing the set of constraints  $A_c g = b_c$  onto the optimization problem (13) and solve it centrally, we propose to solve (13) iteratively to make parts of the computation distributed.

Before we proceed on how we distribute the optimization problem (13), note that we can summarize the linear inequality constraints in Equation (13) by  $A_\lambda \lambda + A_g g \leq 0$ . Additionally, in order to convert inequality constraints to equality constraints, we utilize slack variables  $0 \leq \delta \in \mathbb{R}_+^Q$ , to rewrite (13) as:

$$\begin{aligned} & \underset{\lambda, g}{\text{maximize}} && \sum_{l \in \mathcal{L}_{\text{entry}}} U(\lambda_l) \\ & \text{subject to} && A_\lambda \lambda + A_g g + \delta = 0. \end{aligned} \quad (14)$$

##### B. Iterative Solution of Utility Maximization Problem

The special structure of optimization problem (14) enables us to use ideas from Augmented Lagrangian [23] and ADMM [19] techniques to solve (14) iteratively such that the update step of  $g$  is separated from  $\lambda$ . This further leads to distributing the update step of  $\lambda$  such that each entry link solves its own optimization problem to decide on the amount of flow it can let in. To achieve this goal, construct the augmented Lagrangian of (14) as follows:

$$\begin{aligned} L_\rho = \sum_{l \in \mathcal{L}_{\text{entry}}} U(\lambda_l) + \alpha^T (A_\lambda \lambda + A_g g + \delta) + \\ \frac{1}{2} \rho \|A_\lambda \lambda + A_g g + \delta\|^2. \end{aligned} \quad (15)$$

In (15),  $\alpha \in \mathbb{R}_+^Q$  is the vector of dual variables or prices, and  $\rho$  is a finite positive number or increasing sequence penalizing for deviations from equality constraints. We can then solve (15) iteratively via the following algorithm:

- 1) At  $k = 0$ , initialize  $\alpha^0$ ,  $\lambda^0$ ,  $\delta^0$ , and  $\rho > 0$  arbitrarily.
- 2) Update  $g^k$  and  $\delta^k$  as follows:

$$\begin{aligned} [g^{k+1}, \delta^{k+1}] = \underset{g, \delta}{\text{argmax}} & \alpha^{kT} (A_\lambda \lambda^k + A_g g + \delta) \\ & + \frac{1}{2} \rho \|A_\lambda \lambda^k + A_g g + \delta\|^2 \\ \text{subject to} & A_c g = b_c. \end{aligned} \quad (16)$$

- 3) Update  $\lambda^k$  as follows:

$$\begin{aligned} \lambda^{k+1} = \underset{\lambda}{\text{argmax}} & \sum_{l \in \mathcal{L}_{\text{entry}}} U(\lambda_l) + \alpha^{kT} (A_\lambda \lambda + A_g g^{k+1} + \\ & \delta^{k+1}) + \frac{1}{2} \rho \|A_\lambda \lambda + A_g g^{k+1} + \delta^{k+1}\|^2. \end{aligned} \quad (18)$$

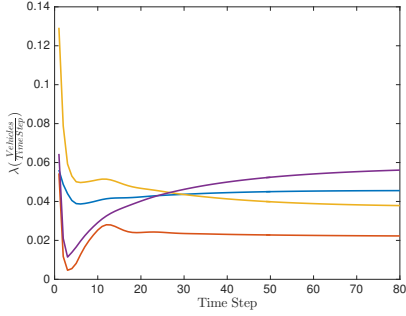
- 4) Update  $\alpha$  by:

$$\alpha^{k+1} = \alpha^k + \rho (A_\lambda \lambda^{k+1} + A_g g^{k+1} + \delta^{k+1}). \quad (19)$$

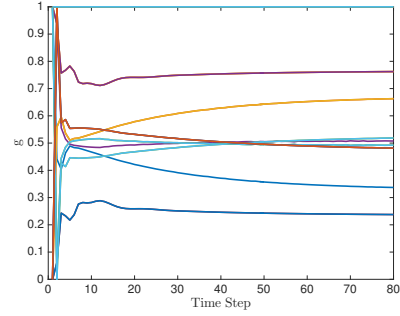
- 5) Apply  $\lambda^{k+1}$  and  $g^{k+1}$  to the system and go to step 2 in the next cycle time  $T$ .

The implicit assumption in the above implementation is that the time step of the algorithm is the cycle time of the fixed-time control. In other words, at the beginning of every cycle time, we update  $g$  and  $\lambda$ , apply them and repeat the same procedure in the next cycle time. It is important to mention that the additional control requirements are satisfied by constraining  $g$  in step 2. This further assures that the obtained green ratios satisfy the hard constraints that are essential to synthesize a valid signal plan that is implementable at each cycle.

Note that the optimization problem (18) is an unconstrained optimization problem. Moreover, for each arrival  $i$ , its objective function consists of  $U(\lambda_i)$ , quadratic terms and a linear



(a) Arrivals versus time steps (iterations).



(b) Green ratios of queues versus time steps (iterations).

Fig. 3: Arrivals and green ratios of the queues obtained from the utility maximization algorithm.

term. In fact, the optimization problem (18) has the following format:

$$\lambda^{k+1} = \underset{\lambda}{\operatorname{argmax}} \sum_{l \in \mathcal{L}_{\text{entry}}} U(\lambda_l) + \sum_{l, m \in \mathcal{L}_{\text{entry}}} f_{lm}(\lambda_l, \lambda_m), \quad (20)$$

with  $U(\lambda_l)$  being strictly concave and  $f_{lm}(\lambda_l, \lambda_m) = \beta_{lm}(\lambda_l \lambda_m) + \gamma \lambda_l$  being concave quadratic functions. This special structure of the objective function allows us to solve (20) in a distributed fashion using Min Sum Message Passing Algorithm [24] as follows:

- 1) At  $i = 0$ , initialize  $\tilde{\lambda}(0) > 0$  arbitrarily.
- 2) Communicate  $\tilde{\lambda}(i)$ s to entry links.
- 3) Let each entry link  $l$  update its arrival rate by:

$$\lambda_l(i+1) = \underset{\lambda}{\operatorname{argmax}} U(\lambda) + \sum_{m \in \mathcal{L}_{\text{entry}-l}} \beta_{lm}(\lambda \tilde{\lambda}_m) + \gamma \lambda + \beta_{ll} \lambda^2. \quad (21)$$

- 4) Go to step 2 and repeat the procedure.

In the above algorithm,  $\lambda_l(i)$  is the value of  $\lambda_l$  at the  $i$ th communication of the message passage algorithm. This is different from  $\lambda_l^k$  which is  $\lambda_l$  at time step  $k$ .  $\lambda_l^k$  is implemented at time step  $k$ , whereas,  $\lambda_l(i)$  is only utilized when it converges to the optimal solution. Once the algorithm converges, we use the obtained updated arrival rates to update the dual variables via (19). An interesting property of (21) is that it has an analytical solution which eliminates the need for further computations. Let  $f_{lm}(\lambda, \tilde{\lambda}_m) = \beta_{lm}(\lambda \tilde{\lambda}_m)$ . It is easy to verify that the solution to (21) satisfies:

$$U'(\lambda) + \sum_{m \in \mathcal{L}_{\text{entry}-l}} \beta_{lm} \tilde{\lambda}_m + \gamma + 2\beta_{ll} \lambda = 0, \quad (22)$$

where  $\beta_{lm} \tilde{\lambda}_m + \gamma$  is simply a constant known value and  $U'(\lambda)$  is the derivative of the utility function. Note that distributing the computation is compulsory when dealing with Cyber Physical Systems such as transportation networks where there is normally limited computational capacity on the field; thus, we do require to formulate the problem such that it can be solved in a distributed fashion.

The solution to this problem would be fully distributed provided that (16) can also be distributed. Due to the hard

constraints on  $g$ , it is generally hard to achieve this goal. Nonetheless, since we have introduced a quadratic program in (16), we can use active set methods [25] to distribute (16) with few number of communications and iterations as illustrated in [26]. This implies that formulating the problem such that we end up with a quadratic program in (16) paves the way for distributing (16) as well.

## V. EXAMPLE

In order to evaluate the performance of our algorithm, we utilize it for boundary flow control and signal control of the network shown in Figure 2. The network is subject to arrivals in links 1, 8, 5, and 13. We wish to regulate the flow that is allowed to enter through entry links by our algorithm while guaranteeing that the network is free of congestion. The network contains 17 links and 20 queues. Turn ratios at intersections are known and assumed to be constant. We used the  $\log(\cdot)$  function as our utility function. The cycle time for all intersections is 90 seconds.

Figure 3 illustrates the arrivals and green ratios found during 80 time steps. As it can be observed, the algorithm converges to the optimal solution of maximizing the aggregate network utility in a small number of iterations. We further ran the simple dual-decomposition-based method (without the extra quadratic term in the Lagrangian) on the same network. However, it took more than 4000 time steps for the solution to converge, which makes it essentially impractical for transportation networks where the time step of the system is at the order of cycle times.

In order to verify that our control algorithm can successfully stabilize the network, we examine the queue lengths for all movements in the network to assure that they remain bounded. Figure 4 demonstrates the evolution of the sum of all queues in the network, which clearly remains bounded throughout the simulation demonstrating that the control successfully preserves network stability.

## VI. CONCLUSION AND FUTURE WORK

In summary, we have introduced the notion of network utility maximization for fair allocation of available network resources to different arrivals that want to enter the network,



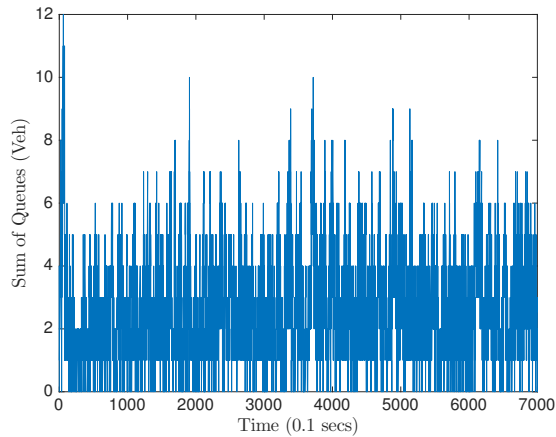


Fig. 4: Sum of all queues vs. time.

while stabilizing the network. To the best of our knowledge, no other control law has been proposed in the literature capable of encoding this property and synthesizing joint network congestion control and signal control. We demonstrated how our algorithm can be partially distributed to reduce the computational burdens when dealing with large-scale networks. We further showed that using ADMM, our algorithm can achieve much faster convergence rate compared to the existing dual decomposition methods for utility maximization. Finally, we illustrated that our algorithm successfully stabilizes an exemplar network.

We note that for our solution to be fully distributed, we need to be able to distribute the computation required for updating the timings. Utilizing distributed active set or  $\epsilon$ -exact penalty function [27] methods for achieving this goal can be of importance and interest. Additionally, since our iterative control algorithm can potentially adapt to the changes of system parameters, it can be employed for developing control policies that are adaptive and robust.

#### ACKNOWLEDGMENT

This work is supported by the National Science Foundation under Grant No. CPS 1446145 and the startup grant for Ramtin Pedarsani.

#### REFERENCES

- [1] D. Husch and J. Albeck, "Synchro 6: Traffic signal software, user guide," *Albany, Calif*, vol. 364, pp. 368–0, 2003.
- [2] J. Stevanovic, A. Stevanovic, P. T. Martin, and T. Bauer, "Stochastic optimization of traffic control and transit priority settings in vissim," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 3, pp. 332–349, 2008.
- [3] D. I. Robertson and R. D. Bretherton, "Optimizing networks of traffic signals in real time—the scoot method," *IEEE Transactions on vehicular technology*, vol. 40, no. 1, pp. 11–15, 1991.
- [4] N. H. Gartner, F. J. Pooran, and C. M. Andrews, "Implementation of the opac adaptive control strategy in a traffic signal network," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*. IEEE, 2001, pp. 195–200.
- [5] A. Muralidharan, R. Pedarsani, and P. Varaiya, "Analysis of fixed-time control," *Transportation Research Part B: Methodological*, vol. 73, pp. 81–90, 2015.

- [6] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.
- [7] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn, "Efficient network-wide model-based predictive control for urban traffic networks," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 122–140, 2012.
- [8] S. Koehler, N. Mehr, R. Horowitz, and F. Borrelli, "Stable hybrid model predictive control for ramp metering," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 1083–1088.
- [9] N. Mehr, D. Sadigh, and R. Horowitz, "Probabilistic controller synthesis for freeway traffic networks," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 880–880.
- [10] 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.
- [11] S. Coogan, E. A. Gol, M. Arcaç, and C. Belta, "Controlling a network of signalized intersections from temporal logical specifications," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 3919–3924.
- [12] S. Sadraddini and C. Belta, "A provably correct mpc approach to safety control of urban traffic networks," *arXiv preprint arXiv:1602.01028*, 2016.
- [13] N. Mehr and R. Horowitz, "Probabilistic freeway ramp metering," *rN*, vol. 1, no. f1, p. 1, 2016.
- [14] N. Mehr, D. Sadigh, R. Horowitz, S. S. Sastry, and S. A. Seshia, "Stochastic predictive freeway ramp metering from signal temporal logic specifications," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 4884–4889.
- [15] C. Diakaki, M. Papageorgiou, and K. Aboudolas, "A multivariable regulator approach to traffic-responsive network-wide signal control," *Control Engineering Practice*, vol. 10, no. 2, pp. 183–195, 2002.
- [16] A. Kouvelas, M. Saeedmanesh, and N. Geroliminis, "Feedback perimeter control for heterogeneous urban networks using adaptive optimization," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 882–887.
- [17] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.
- [18] R. Pedarsani, "Robust scheduling for queueing networks," 2015.
- [19] 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, April 2011.
- [20] R. Pedarsani, J. Walrand, and Y. Zhong, "Robust scheduling and congestion control for flexible queueing networks," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE, 2014, pp. 467–471.
- [21] N. Mehr, R. Horowitz, and R. Pedarsani, "Low-complexity ramp metering for freeway congestion control via network utility maximization," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, p. to appear.
- [22] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking (ToN)*, vol. 8, no. 5, pp. 556–567, 2000.
- [23] M. J. Powell, "A method for non-linear constraints in minimization problems". UKAEA, 1967.
- [24] C. C. Moallemi and B. Van Roy, "Convergence of min-sum message-passing for convex optimization," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 2041–2050, 2010.
- [25] M. Hintermüller, K. Ito, and K. Kunisch, "The primal-dual active set strategy as a semismooth newton method," *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 865–888, 2002.
- [26] S. Koehler, C. Danielson, and F. Borrelli, "A primal-dual active-set method for distributed model predictive control," *Optimal Control Applications and Methods*, 2016.
- [27] S. S. Kia, "Distributed optimal resource allocation over networked systems and use of an  $\epsilon$ -exact penalty function," *IFAC-PapersOnLine*, vol. 49, no. 4, pp. 13–18, 2016.