A Micro-Simulation Study of the Generalized Proportional Allocation Traffic Signal Control

Gustav Nilsson and Giacomo Como

***Abstract*—We study the problem of controlling phase activations for signalized junctions in an urban transportation network using local feedback information consisting of measures of the queue-lengths at the incoming lanes of each junction. Our focus is on the validation and performance evaluation through micro-simulations of the recently proposed Generalized Proportional Allocation (GPA) controller. Previous theoretical work has provided provable performance guarantees in terms of stability, and throughput optimality of the GPA controller in a continuous averaged dynamical queueing network model. In this paper, we first provide and implement two discretized versions of the GPA controller in the SUMO micro simulator. We then compare, in an artificial Manhattan-like grid, the performance of the GPA controller with those of the MaxPressure controller, which is another distributed feedback controller that requires more information than the GPA. Finally, to show that the GPA controller is easily implementable in a real-world scenario, we apply it to a previously published realistic traffic scenario for the city of Luxembourg and compare its performance with the static controller provided with the scenario as well as with the cyclic MaxPressure controller. The simulations show that the GPA controller outperforms both the fixed time and the cyclic MaxPressure controllers for the Luxembourg scenario, and behaves better than the MaxPressure pressure controller in the Manhattan-grid when the demands are low.**

Index terms: Decentralized Traffic Signal Control, Microscopic Traffic Simulation

I. Introduction

While the first traffic signals were controlled completely in open loop, various approaches have been taken to adjust the green light allocation based on the current traffic situation. The problem of traffic signal control for a single junction has been well studied in, e.g., [1]. Although traffic signal control for a whole transportation network is more complex, several approaches have been undertaken. To mention a few, SCOOT [2], UTOPIA [3], and SCATS [4] all focus on controlling traffic signals in a whole transportation network. Other control theoretic approaches to traffic signal are, e.g., learning-based methods [5], [6], [7], hybrid control [8], and fuzzy control [9]. For a review of traffic control, see [10].

However, these approaches generally lack formal stability, optimality, and robustness guarantees. In [11], [12], a decentralized feedback controller for traffic control was proposed, referred to as Generalized Proportional Allocation (GPA), for which both stability and throughput optimality guarantees can be proved in a dynamical queueing network model. These promising theoretical results motivate the investigation of whether the GPA control performance can be validated in a micro-simulator with more realistic traffic dynamics, which is the subject of the present paper.

More specifically, in [11], [12], a GPA average control action for traffic signals in continuous time is studied. Under the assumptions that the controller can measure the whole queue lengths at each junction, the averaged controller is shown to be throughput-optimal from a theoretical perspective. With this, we mean that when the traffic dynamics is modeled as a network of point queues there exists no controller that can handle larger constant exogenous inflows to the network than the GPA controller. That throughput can be optimized by fully decentralized control policies using no global information on the network is in line with similar results in resilient control of dynamical flow networks [13], [14]. The throughput optimality property also implies that there are formal guarantees that the controller will not create gridlock situations in the network. Note that, as exemplified in [15], in general, feedback controllers performing well for a single isolated junction may cause gridlocks in a network setting.

At the same time, the GPA controller requires very little information about the network topology and traffic flow propagation. Indeed, all information the GPA controller needs to determine the phase activation in a junction is the queue lengths on the incoming lanes to that junction and the static set of phases. This makes the controller fully distributed, as no information is required on the global network structure or about the state in the other junctions in order to compute the control action in one junction. The GPA traffic signal controller also has the property that it adjusts the cycle lengths depending on the demand, extending it when the demand is higher and reducing it when it is lower. The fact that during higher demands, the cycle lengths should be longer to waste less service time due to phase shifts, has been suggested previously for open-loop traffic signal control, see e.g., [16].

Another feedback control strategy for traffic signal control is the MaxPressure controller [17], [15] that builds upon the same idea as the BackPressure controller, proposed for communication networks in [18]. While the BackPressure controller controls both the routing (to which servers the packets should proceed after received service) and the scheduling (which subset of queues that should be severed), the MaxPressure controller only controls the latter, i.e., the phase activation but not the routing. More recently, due to the rapid development

G. Nilsson is with the School of Electrical and Computer Engineering (ECE), Georgia Institute of Technology, Atlanta, GA. gustav.nilsson@gatech.edu

G. Como is with the Department of Mathematical Sciences, Politecnico di Torino, Italy, and the Department of Automatic Control, Lund University, Sweden. giacomo.como@polito.it

G. Como is member of the excellence centers LCCC and ELLIT. This research was carried on within the framework of the MIUR-funded *Progetto di Eccellenza* of the *Dipartimento di Scienze Matematiche G.L. Lagrange*, CUP: E11G18000350001, and was partly supported by the *Compagnia di San Paolo* and the Swedish Research Council (VR).

of autonomous vehicles, it has been proposed in [19] to utilize the routing control from the BackPressure controller in traffic networks as well. Like the GPA controller, the MaxPressure controller can also be proved to be throughput optimal. However, the MaxPressure controller requires information about the tuning ratios at each junction, i.e., how the vehicles (on average) propagate from one junction to the neighboring junctions. Although various techniques for estimating those turning ratios have been made, for example [20], with more and more drivers or autonomous vehicles doing their path planning through some routing service, it is likely to believe that the turning ratios can change in an unpredictable way when a disturbance such as an accident occurs in the transportation network. If the traffic signal controller has information about the turning ratios, other control strategies are possible as well, for instance, MPC-like as proposed in [21], [22], [23] and robust control as proposed in [24].

In a conference contribution [25] we presented the first discretization and validation results of the GPA in a microscopic traffic simulator. Although these preliminary results were promising, the validations were only performed on an artificial network and only compared with a fixed-time traffic signal controller. Moreover, the GPA was only discretized in a way such that all phases in a cycle are activated. In this paper, we substantially extend such preliminary results in [25] by showing another discretization that does not have to utilize the full cycle, and we also perform new validations. The new validations both compare the GPA to the MaxPressure controller on an artificial network (the reason for choosing an artificial network will be explained later), but also validate the GPA controller in a realistic scenario, namely for the Luxembourg city during a whole day. Since both the GPA and the MaxPressure are decentralized and throughput optimal, by comparing them side-to-side, we can investigate whether the additional information about the turning ratios that the MaxPressure controller requires yields any performance gain, and if this dependence on external information makes the MaxPressure less robust as compared to the GPA.

The outline of the paper is as follows: In Section II we present the model we are using for traffic signals, together with a problem formulation of the traffic signal control problem. In Section III we present two different discretizations of the GPA that we are using in this study and give brief descriptions of the MaxPressure controller and the Cyclic MaxPressure controller. In Section IV we compare the GPA controller with the MaxPressure controller on an artificial Manhattan-like grid, and in Section V we investigate how the GPA controller performs in a realistic traffic scenario, where it is also compared to the Cyclic MaxPressure controller. The paper is concluded with some ideas about further research.

II. Model and Problem Formulation

In this section, we describe the model for traffic signals that will be used throughout the paper, together with the associated control problem.

We consider an arterial traffic network with signalized junctions. Let $\mathcal{J}$ denote the set of such signalized junctions. For a junction $j \in \mathcal{J}$, we let $\mathcal{L}^{(j)}$ be the set of incoming lanes, on which the vehicles can queue up. The set of all lanes in the whole network will be denoted by $\mathcal{L} = \bigcup_{j\in\mathcal{J}} \mathcal{L}^{(j)}$. For a lane $l \in \mathcal{L}$, the queue-length at time t —measured in the number of vehicles— is denoted by $x_l(t)$.

Each junction has a predefined set of *phases* $\mathcal{P}^{(j)}$ of size n_j. For simplicity, we assume that phases $p_i \in \mathcal{P}^{(j)}$ are indexed by $i = 1, \dots, n_j$. A phase $p \in \mathcal{P}^{(j)}$ is a subset of incoming lanes to the junction j that can receive green light simultaneously. Throughout the paper, we will assume that for each lane $l \in \mathcal{L}$, there exists a unique junction $j \in \mathcal{J}$ with at least one phase $p \in \mathcal{P}^{(j)}$ such that $l \in p$.

The phases are usually constructed such that the vehicles' paths in a junction do not cross each other, in order to avoid collisions. Examples of this will be shown later in the paper. After a phase has been activated, it is common to signalize to the drivers that the traffic signal is turning red and give time for vehicles that are in the middle of the junction to leave it before the next phase are activated. Such time is usually referred to as clearance time. Throughout the paper, we shall refer to those phases containing yellow traffic signals as *clearance phases* (in contrast to regular phases corresponding to configurations when lanes receive green traffic light). We will assume that a clearance phase activation follows each phase activation. We will let the phase activation time vary and make the natural assumption that the clearance phases has to be activated for a fixed time.

For a given junction $j \in \mathcal{J}$, the set of phases can be described through a phase matrix $P^{(j)}$ of dimension $|\mathcal{L}^{(j)}| \times |\mathcal{P}^{(j)}|$, where

$$P^{(j)}_{il} = \begin{cases} 1 & \text{if} \quad \text{lane } l \text{ belongs to the } i\text{-th phase} \\ 0 & \text{if} \quad \text{otherwise}\,. \end{cases}$$

While neither the phase matrix nor the corresponding set of phases $\mathcal{P}^{(j)}$ contain the clearance phases, to each phase $p \in \mathcal{P}^{(j)}$ we will associate a clearance phase, denoted p'. We denote the set of real phases and their corresponding clearance phases by $\bar{\mathcal{P}}^{(j)}$.

The controller's task in a signalized junction is to define a *signal program*, $\mathcal{T}^{(j)} = \{(p, t_{\text{end}}) \in \bar{\mathcal{P}}^{(j)} \times \mathbb{R}_+\}$, where the phase p is activated until t_{end}. When $t = t_{\text{end}}$, the phase p', where $(p', t'_{\text{end}}) \in \mathcal{T}^{(j)}$, with smallest $t'_{\text{end}} > t$ is activated. Formally, we can define the function $c^{(j)}(t)$ that gives the phase that is activated at time t as follows

$$c^{(j)}(t) = \{p : (p, t_{\text{end}}) \in \mathcal{T}^{(j)} \mid \\ t_{\text{end}} > t \text{ and } t_{\text{end}} \le t'_{\text{end}} \text{ for all } (p', t'_{\text{end}}) \in \mathcal{T}^{(j)}\}\,.$$

What $c^{(j)}(t)$ is doing is to find the phase with the smallest end-time greater than the current time.

Finally, we let

$$T^{(j)} = \max\{t_{\text{end}} \mid (p, t_{\text{end}}) \in \mathcal{P}^{(j)}\}$$

denote the time when the signal program for junction j ends, and hence a new signal timing program has to be determined.

Example 1: Consider the junction in Figure 1 with the incoming lanes numbered as in the figure. In this case, the

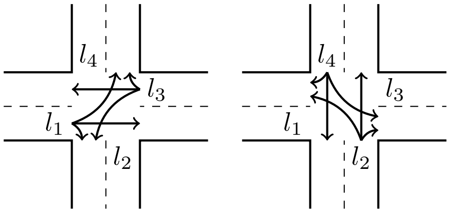

Fig. 1. The phases for the junction in Example 1. This junction has four incoming lanes and two phases, $p_1 = \{l_1, l_3\}$ and $p_2 = \{l_2, l_4\}$. Hence there is no specific lane left-turning left.

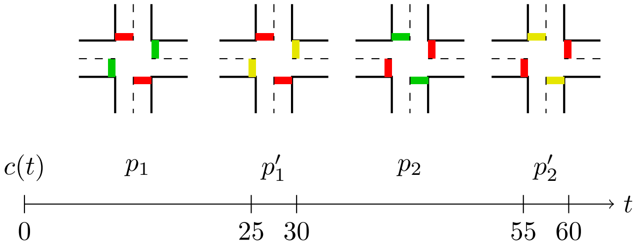

Fig. 2. Example of a signal program for the junction in Example 1. In this example the signal program is $\mathcal{T} = \{(p_1, 25), (p_1', 30), (p_2, 55), (p_2', 60)\}$.

drivers turning left have to solve collision avoidance by themselves. The phase matrix is

$$P^{(j)} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} .$$

An example of signal program is shown in Figure 2. Here the program is $\mathcal{T} = \{(p_1, 25), (p_1', 30), (p_2, 55), (p_2', 60)\}$. which means that both the phases are activated for 25 seconds each, and the clearance phases are activated for 5 seconds each.

III. Feedback Controllers

In this section, we present four different traffic signal controllers that all determine the signal program. The first two are to be interpreted as different discretizations of the GPA controller, where the first one makes sure that all the clearance phases are activated during one cycle, and the second one only activates the clearance phases if their corresponding phase has been activated. The third controller is the MaxPressure controller, which by default does not guarantee cyclic phase activation. The fourth controller is a modification of MaxPressure, presented in [26], which ensures a cyclic phase activation.

All the four controllers are feedback-based, i.e., when one signal program has reached its end, the current queue lengths are used to determine the upcoming signal program. Moreover, the GPA controllers are fully distributed, in the sense that in order to determine the signal program in one junction, the controller only needs information about the queue-lengths on the incoming lanes for that junction. The MaxPressure controller is also distributed in the sense that it does not require network-wide information, but it requires queue-length information from the neighboring junctions as well.

For the sake of simplicity, we shall assume that after a phase has been activated, a clearance phase has to be activated for a fixed amount of time $T_w > 0$, that is independent of which phase that has just been activated, for all the controllers presented in this section.

A. GPA with Full Clearance Cycles

For this controller, we assume that all the clearance phases have to be activated for each cycle. When $t = T^{(j)}$ for a junction j, a new signal program is computed by solving the following convex optimization problem:

$$\begin{aligned} \underset{\substack{\nu \in \mathbb{R}_+^{n_j} \\ w \in \mathbb{R}_+}}{\text{maximize}} \quad & \sum_{l \in \mathcal{L}^{(j)}} x_l(t) \log \left(((P^{(j)})^T \nu)_l \right) + \kappa \log(w) , \\ \text{subject to} \quad & \sum_{1 \le i \le n_j} \nu_i + w = 1 , \\ & w \ge \bar{w} . \end{aligned} \tag{1}$$

In the optimization problem above, $\kappa > 0$ and $\bar{w} \ge 0$ are tuning parameters for the controller, and their interpretation will be discussed later. Recall that $x_l(t)$ is the measured queue length on lane l at time t. Also observe that the for each junction $j \in \mathcal{J}$, the control action depends exclusively on the queues on the incoming lanes. To stress out this dependence we let $x^{(j)}$ denote the vector of queue lengths for the incoming lanes to junction j, i.e., $x^{(j)}$ is the projection of $x(t)$ on the lanes in $\mathcal{L}^{(j)}$. The objective function is the one used for the GPA controller in [11], [12], where it is shown that this choice of objective function achieves throughput-optimality for an entire traffic network.

The entries of the vector ν in the solution of the optimization problem above determine the fraction of the cycle time that each phase should be activated. The variable w represents the fraction of the cycle time that should be allocated to the clearance phases. Because ν and w are determining fractions of cycle, the entries of ν together with w have to sum up to 1. Observe that, as long as the queue lengths are finite, w will be strictly greater than zero. Since we assume that each clearance phase has to be activated for a fixed amount of time, $T_w > 0$, the total cycle length T_{cyc} for the upcoming cycle can be computed by

$$T_{\text{cyc}} = \frac{n_j T_w}{w} .$$

With the knowledge of the full-cycle length, the signal program for the upcoming cycle can be computed according to Algorithm 1.

Although the optimization problem (1) can be solved in real-time using convex solvers, the optimization problem can also be solved analytically in some special cases. One such case is when the phases are orthogonal, i.e., every incoming lane only belongs to one phase. If the phases are orthogonal, then $P^T \mathbb{1} = \mathbb{1}$. In the case of orthogonal phases and $\bar{w} = 0$, the solution to the optimization problem in (1) is given by

$$\begin{aligned} \nu_i(x(t)) &= \frac{\sum_{l \in \mathcal{L}^{(j)}} P_{il} x_l(t)}{\kappa + \sum_{l \in \mathcal{L}^{(j)}} x_l(t)} , \qquad i = 1, \dots, n_j , \\ w(x(t)) &= \frac{\kappa}{\kappa + \sum_{l \in \mathcal{L}^{(j)}} x_l(t)} . \end{aligned} \tag{2}$$

Algorithm 1: GPA with Full Clearance Cycles

```
Data: Current time $t$, local queue lengths $x^{(j)}(t)$,
      phase matrix $P^{(j)}$, clearance time $T_w$, tuning
      parameters $\kappa, \bar{w}$
Result: Signal program $\mathcal{T}^{(j)}$
$\mathcal{T}^{(j)} \leftarrow \emptyset$
$n_j \leftarrow$ Number of rows in $P^{(j)}$
$(\nu, w) \leftarrow$ Solution to (1) given $x^{(j)}(t), P^{(j)}, \kappa, \bar{w}$
$T_{\text{cyc}} \leftarrow n_j \cdot T_w / w$
$t_{\text{end}} \leftarrow t$
for $i \leftarrow 1$ to $n_j$ do
    $t_{\text{end}} \leftarrow t_{\text{end}} + \nu_i \cdot T_{\text{cyc}}$
    $\mathcal{T}^{(j)} \leftarrow \mathcal{T}^{(j)} + (p_i, t_{\text{end}})$        ▷ Add phase $p_i$
    $t_{\text{end}} \leftarrow t_{\text{end}} + T_w$
    $\mathcal{T}^{(j)} \leftarrow \mathcal{T}^{(j)} + (p_i', t_{\text{end}})$   ▷ Add clearance phase $p_i'$
end
```

From the expression of w above, a direct expression for the total cycle length can be obtained

$$T_{\text{cyc}} = T_w n_j + \frac{T_w n_j}{\kappa} \sum_{l \in \mathcal{L}^{(j)}} x_l(t) \,.$$

From the expressions above we can observe a few things. First, we see that the fraction of the cycle time that each phase is activated is proportional to the queue lengths in that phase, and this explains why we call this control strategy generalized proportional allocation. Moreover, we get an interpretation of the tuning parameter κ, as it determines how the cycle length T_{cyc} should scale with the current queue lengths. If κ is small, even small queue lengths will cause longer cycles, while if κ is large, the cycles will be short even for large queues. Hence, too small κ may give too long cycles, which can result in lanes getting more green light than needed and the controller ending up giving the green light to empty lanes, while vehicles in other lanes are waiting for service. On the other hand, too large a κ may make the cycle lengths so short that the fraction of the cycle time that each phase gets activated is too short for the drivers to react on.

Remark 1: In [12] we showed that the averaged continuous-time GPA controller can stabilize the network and hence keep the queue-lengths bounded. Moreover, this averaged version is throughput-optimal, which means that no controller can handle more exogenous inflow to network than this controller.

However, when implementing the GPA controller, the control signal has to be discretized, i.e., the traffic signals are either green or red for an amount of time. The following example shows that an upper bound on the cycle length, i.e., $\bar{w} > 0$ is required to guarantee stability even for an isolated junction when the control signal is discretized. This bound is not required for the averaged controller in [12].

Example 2: Consider a junction with two incoming lanes with unit flow capacity, both having their own phase, i.e., either one of the lanes can receive green light, but not both lanes simultaneously. Let the exogenous inflows of vehicles to each of the two lanes be fixed such that $\lambda_1 = \lambda_2 = \lambda$. Moreover, we let the parameters $T_w = 1$, $\bar{w} = 0$, and we let the initial state be $x_1(0) = A > 0$, and $x_2(0) = 0$. The control signals and the cycle time for the first cycle is then given by

$$\nu_1(x(0)) = \frac{A}{A+\kappa}\,, \quad \nu_2(x(0)) = 0\,, \quad T_{\text{cyc}}(x(0)) = \frac{A+\kappa}{\kappa}.$$

Observe that the cycle time $T_{\text{cyc}}(x(0))$ is strictly increasing with the parameter A. After one full service cycle, i.e., at time $t_1 = T(x(0))$ the queue lengths are

$$\begin{aligned} x_1(t_1) &= A + T(x(0)) \left(\lambda - \frac{A}{A+\kappa} \right) \\ &= \max \left(\overbrace{A + \lambda \frac{A+\kappa}{\kappa} - \frac{A}{\kappa}}^{f(A)}, 0 \right), \\ x_2(t_1) &= T(x(0)) \lambda = \lambda \left(\frac{A+\kappa}{\kappa} \right). \end{aligned}$$

If A and κ are chosen such that $x_1(t_1) = 0$, then due to symmetry, the analysis of the system can be repeated in the same way with a new initial condition. To make sure that one queue always gets empty during the service cycles, it must hold that $f(A) \leq 0$. Moreover, to make sure that the other queue grows larger than the previous queue, it must also hold that $x_2(t_1) > A$ which can be equivalently expressed as

$$\begin{aligned} A\kappa + \lambda(A+\kappa) - A &\leq 0\,, \\ A\kappa - \lambda(A+\kappa) &< 0\,. \end{aligned}$$

The choice of $\lambda = \kappa = 0.1$ and $A = 1$ is one set of parameters satisfying the constraints above, and will hence make the queue lengths and cycle times grow unboundedly. How the queue lengths and cycle times evolve in this case is shown in Figure 3.

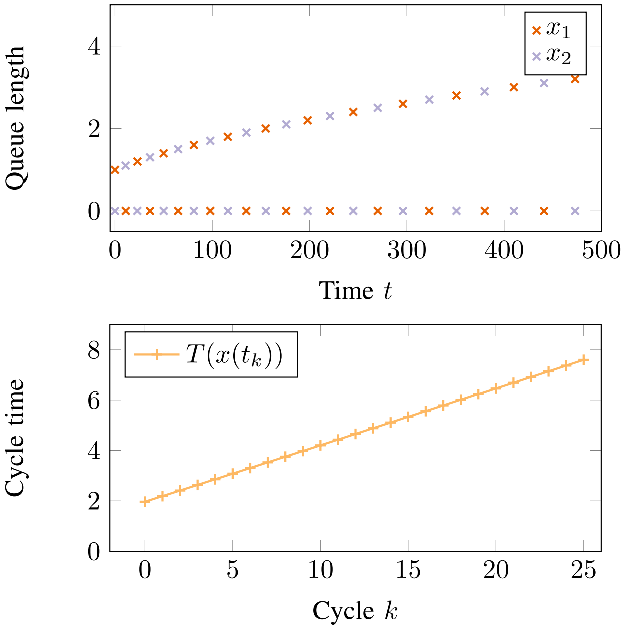

Fig. 3. How the queue lengths evolve in time together with the cycle times for the system in Example 2. We can observe that the cycle length increases for each cycle.

While imposing an upper bound on the cycle length, and hence a lower bound on w, is necessary to obtain the stability, it will shrink the throughput region. In other words, by setting $\bar{w} > 0$ the controller will not be able to handle as large traffic flows as if $\bar{w} = 0$. An upper bound on the cycle length may occur naturally, due to the fact that the sensors cover a limited area and hence the measurements will saturate. However, we will later observe in the simulations that letting $\bar{w} > 0$ may improve the performance of the controller when it is simulated in a realistic scenario, even when saturation of the queue length measurements is possible.

B. GPA with Shorted Cycles

One possible drawback of the controller in Section III-A is that it has to activate all the clearance phases in one cycle. This property implies that if the junction is empty when the signal program is computed, it will take $n_j T_w$ seconds until a new signal program is computed. Motivated by this shortcoming, we also present a version of the GPA where only the clearance phases get activated if their corresponding phases have been activated. If we let n'_j denote the number of phases that will be activated during the upcoming cycle, the total cycle time is given by

$$T_{\text{cyc}} = \frac{n'_j T_w}{w} .$$

How to compute the signal program in this case, is shown in Algorithm 2.

C. MaxPressure

As mentioned in the introduction, the MaxPressure controller is another throughput-optimal feedback controller for traffic signals. The MaxPressure controller computes the difference between the queue lengths and their downstream queue lengths in each phase in order to determine each phase's pressure. It then activates the phase with the largest pressure for a fixed time interval. To compute the pressure, the controller needs information about where the outflow from every queue will proceed. To model this, we introduce the routing matrix R of dimension $|\mathcal{E}| \times |\mathcal{E}|$, whose entries R_{ij} coincide with the fraction of vehicles that will proceed from lane i in the current junction to lane j in a downstream junction.

With the knowledge of the routing matrix and under the assumption that the flow rates are the same for all phases, the pressure ω_i of each phase $p_i \in \mathcal{P}^{(j)}$ can then be computed as

$$\omega_i = \sum_{l \in p_i} \left(x_l(t) - \sum_k R_{lk} x_k(t) \right) . \tag{3}$$

The phase that should be activated is then any phase in the set $\operatorname{argmax}_i \omega_i$.

Apart from the routing matrix, the MaxPressure controller has one tuning parameter, the phase duration $d > 0$. That parameter determines how long a phase should be activated for, and hence how long it should take until the pressures are resampled and a new phase activation decision is made.

How to compute the signal program with the MaxPressure controller is shown in Algorithm 3.

Algorithm 2: GPA with Shorted Cycles

```
Data: Current time $t$, local queue lengths $x^{(j)}(t)$,
      phase matrix $P^{(j)}$, clearance time $T_w$, tuning
      parameters $\kappa, \bar{w}$
Result: Signal program $\mathcal{T}^{(j)}$
$\mathcal{T}^{(j)} \leftarrow \emptyset$
$n_j \leftarrow$ Number of rows in $P^{(j)}$
$(\nu, w) \leftarrow$ Solution to (1) given $x^{(j)}(t), P^{(j)}, \kappa, \bar{w}$
▷ Compute the number of phases to be activated
$n'_j \leftarrow 0$
for $i \leftarrow 1$ to $n_j$ do
    if $\nu_i > 0$ then
        $n'_{p_j} \leftarrow n'_j + 1$
    end
end
if $n'_j > 0$ then
    ▷ If vehicles are present on some phases, activate
      those
    $T_{\text{cyc}} \leftarrow n'_j \cdot T_w / w$
    $t_{\text{end}} \leftarrow t$
    for $i \leftarrow 1$ to $n_j$ do
        if $\nu_i > 0$ then
            $t_{\text{end}} \leftarrow t_{\text{end}} + \nu_i \cdot T_{\text{cyc}}$
            ▷ Add phase $p_i$
            $\mathcal{T}^{(j)} \leftarrow \mathcal{T}^{(j)} + (p_i, t_{\text{end}})$
            $t_{\text{end}} \leftarrow t_{\text{end}} + T_w$
            ▷ Add clearance phase $p'_i$
            $\mathcal{T}^{(j)} \leftarrow \mathcal{T}^{(j)} + (p'_i, t_{\text{end}})$
        end
    end
else
    ▷ If no vehicles are present, hold a clearance
      phase for one time unit
    $\mathcal{T}^{(j)} \leftarrow (p'_1, t + 1)$
end
```

D. Cyclic MaxPressure

While the standard MaxPressure controller does not preserve the service cycle for a junction, a modified version that makes sure that all the clearance phases get activated has been presented in [26]. The controller computes the pressure for each phase just like the MaxPressure controller, but then activates each phase in proportion to

$$\nu_i = \frac{\exp \eta \omega_i}{\sum_j \exp \eta \omega_j} ,$$

where ω are the pressures form (3) and $\eta > 0$ is a tuning parameter. Observe that, differently from the GPA controller, the Cyclic MaxPressure controller assumes a fixed cycle length, and hence $\sum_{1 \leq i \leq n_j} \nu_i = 1$.

IV. Comparison Between GPA and MaxPressure

A. Simulation setting

In order to compare the proposed GPA controller with the MaxPressure controller, we simulate both controllers on

Algorithm 3: MaxPressure

```
Data: Current time $t$, local queue lengths $x(t)$, phase
      matrix $P^{(j)}$, routing matrix $R$, phase duration $d$
Result: Signal program $\mathcal{T}^{(j)}$
$\mathcal{T}^{(j)} \leftarrow \emptyset$
$n_j \leftarrow$ Number of rows in $P^{(j)}$
for $i \leftarrow 1$ to $n_j$ do
    for $l \in \mathcal{L}^{(j)}$ do
        if $l \in p_i^{(j)}$ then
            $\omega_i \leftarrow \omega_i + x_l(t) - \sum_k R_{lk} x_k(t)$
        end
    end
end
$i \leftarrow \operatorname{argmax}_i \omega_i$
▷ Add phase $p_i$
$\mathcal{T}^{(j)} \leftarrow \mathcal{T}^{(j)} + (p_i, t+d)$
▷ Add clearance phase $p_i'$
$\mathcal{T}^{(j)} \leftarrow \mathcal{T}^{(j)} + (p_i', t+d+T_w)$
```

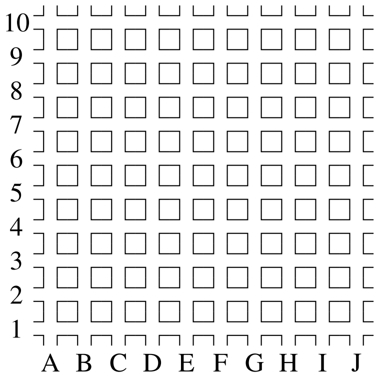

Fig. 4. The Manhattan-like network used in the comparison between GPA and MaxPressure.

an artificial Manhattan-like grid with artificial demand. We shall use the open-source micro-simulator SUMO [27], that simulates every single vehicle's behavior in the traffic network. A schematic drawing of the network is shown in Figure 4. In a setting like this, we can elaborate with the turning ratios, and provide the MaxPressure controller both correct and incorrect turning ratios. This allows us to investigate the robustness properties of both the controllers.

The Manhattan grid in Figure 4 has ten bidirectional north-to-south streets (indexed A to J) and ten bidirectional east-to-west streets (indexed 1 to 10). All streets with an odd number or indexed by letter A, C, E, G or I consist of one lane in each direction, while the others consist of two lanes in each direction. The speed limit on each lane is 50 km/h. The distance between each junction is 300 meters. Fifty meters before each junction, every street has an additional lane, reserved for vehicles that want to turn left. Due to the varying number of lanes, there exist four different junction topologies, all shown in Figure 5, together with the set of possible phases. Each junction is equipped with sensors on the incoming lanes that can measure the number of vehicles queuing up to fifty meters from the junction. The sensors measure the queue lengths by the number of stopped vehicles.

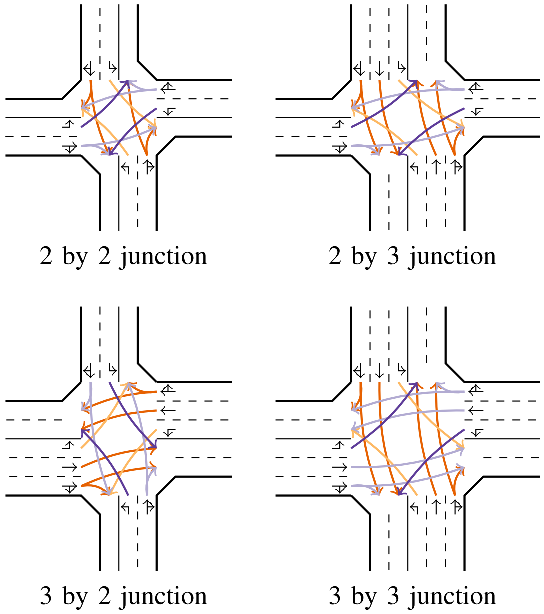
2 by 2 junction
2 by 3 junction
3 by 2 junction
3 by 3 junction

Fig. 5. The four different types of junctions present in the Manhattan grid, together with theirs phases.

Since the scenario is artificial, we can generate demand with prescribed turning ratios and hence let the MaxPressure controller run in an ideal setting. For the demand generation, we assume that at each junction, a vehicle will turn left with probability 0.2, go straight with probability 0.6, and turn right with probability 0.2. We do assume that all vehicles depart from lanes connected to the boundary of the network, and all vehicles will also end their trips when they have reached the boundary of the network. In other words, no vehicles will depart or arrive inside the grid. We will study the controllers' performance for three different demands determined by the probability that a vehicle departs from each boundary lane at each second. We denote this probability δ, where the probabilities for the three different demands are $\delta = 0.05$, $\delta = 0.1$ and $\delta = 0.15$. We generate vehicles for 3600 seconds and then simulate until all vehicles have left the network.

We also compare the results for the GPA controller and the MaxPressure controller with a standard fixed time (FT) controller and a proportional fairness (PF) controller, i.e., the GPA controller with full clearance cycles, but with $\kappa = 0$ and a prescribed fixed cycle length. The reason for comparing with those two controllers is to ensure that the GPA improves the performance as compared to the standard fixed time controller in SUMO and to illustrate that utilizing the proportional fairness controller without adjusting the cycle length does not necessarily improve the performance of the transportation network. For the fixed time controller, the phases containing a straight movement are activated for 30 seconds, and the phases only containing left or right turn movements are activated for 15 seconds. The clearance time for each phase is still set to 5 seconds. This means that the cycle lengths for each of the four types of junctions will be 110 seconds. This is also the fixed cycle time we are using for the proportional fairness controller.

TABLE I
GPA WITH SHORTED CYCLES - MANHATTAN SCENARIO

κ	δ	Total Travel Time [h]
1	0.05	1398
5	0.05	715
10	0.05	699
15	0.05	696
20	0.05	690
1	0.10	7636
5	0.10	1898
10	0.10	1992
15	0.10	2263
20	0.10	2495
1	0.15	$+\infty$
5	0.15	5134
10	0.15	4498
15	0.15	5140
20	0.15	6050

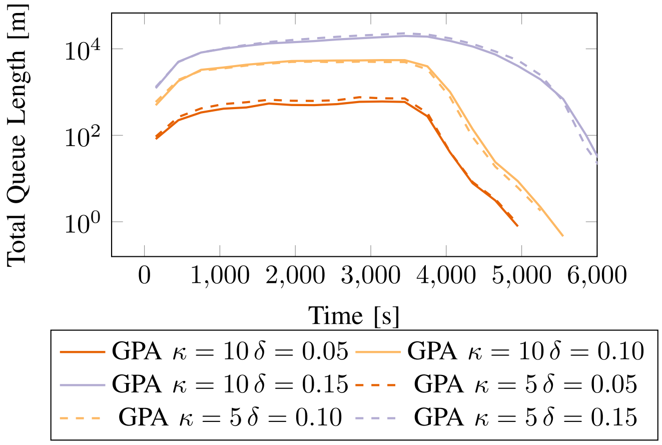

Fig. 6. How the queue lengths vary with time when the GPA with shorted cycles are used in Manhattan grid. The GPA is tested with two different values of $\kappa = 5, 10$ for the three demand scenarios $\delta = 0.05, 0.10, 0.15$. To improve the readability of the results, the queue-lengths are averaged over 300 seconds intervals.

B. GPA Results

Since the phases in this scenario are all orthogonal, the expressions in (2) can be used to solve the optimization problem in (1). The tuning parameter $\bar{w}$ is set to $\bar{w} = 0$ for all simulations. In Table I, we show how the total travel time varies for the GPA controller with shorted cycles for different values of κ. For the demand, $\delta = 0.15$ and $\kappa = 1$ a gridlock situation occurs, probably because vehicles back-spill into upstream junctions. We can see that a value $\kappa = 10$ seems to be the best choice for $\delta = 1$ and $\delta = 0.15$, while a higher κ slightly improves the total travel time for the lowest demand investigated. Letting $\kappa = 10$ is reasonable for other demand scenarios in the same network setting, as observed in [25]. How the total queue lengths varies with time for $\kappa = 5$ and $\kappa = 10$ is shown in Figure 6.

C. MaxPressure Results

The MaxPressure controller decides its control action based on not only queue-lengths on the incoming lanes but also the downstream lanes. It is not always clear in which downstream lane a vehicle will end up in after leaving the junction. If a vehicle can choose between several lanes that are all valid for its path, the vehicle's lane choice will be determined during the simulation and depends upon how many other vehicles are occupying the possible lanes. Because of this, we assume that if a vehicle can choose between several lanes, it will try to join the shortest one. To exemplify how the turning ratios are estimated in those situations, assume that the overall probability that a vehicle is turning right is 0.2, and going straight is 0.6. If a vehicle going straight can choose between lanes l_1 and l_2, but l_2 is also used by vehicles turning right, the probability that the vehicle going straight will queue up in lane l_1 is assumed to be 0.4 and that the probability that the vehicle will queue up in lane l_2 is estimated to be 0.2.

TABLE II
MAXPRESSURE - MANHATTAN SCENARIO

d	δ	TTT correct TR [h]	TTT incorrect TR [h]
10	0.05	858	856
20	0.05	1 079	1 102
30	0.05	1 172	1 193
10	0.10	1 865	1 864
20	0.10	2 254	2 312
30	0.10	2 690	2 718
10	0.15	3 511	3 488
20	0.15	3 992	4 102
30	0.15	5 579	5 590

To also investigate the MaxPressure controller's robustness with respect to the routing information, we perform simulations both when the controller has the correct information about the turning probabilities, i.e., that a vehicle will turn right with probability 0.2, continue straight with probability 0.6 and turn left with probability 0.2. For the simulations when the MaxPressure has the wrong turning information, the controller instead has the information that with probability 0.6 the vehicle will turn right, with probability 0.3 the vehicle will proceed straight and with probability 0.1 the vehicle will turn left. In the simulations, we consider three different phase durations, $d = 10$ seconds, $d = 20$ seconds and $d = 30$ seconds.

How the total queue lengths vary over time for the different demands is shown in Figure 7, Figure 8, and Figure 9. The total travel time, both when the MaxPressure controller is operating with the right, and the wrong turning ratios are shown in Table II. From these results, we can conclude that a shorter phase duration, i.e., $d = 10$, is the most efficient for all demands. This has probably to do with the fact that, with a longer phase duration, the activation time is becoming larger than the time it takes to empty the measurable part of the queue. Another interesting observation is that if the MaxPressure controller has wrong information about the turning ratios, its performance does not decrease significantly.

D. Summary of the Comparison

To better observe the difference between the GPA and MaxPressure, we have plotted the total queue length with the GPA controller with $\kappa = 5$ and $\kappa = 10$, and the best MaxPressure configuration with $d = 10$. The results are shown in Figures 10, 11, and 12. In these figures, we have also

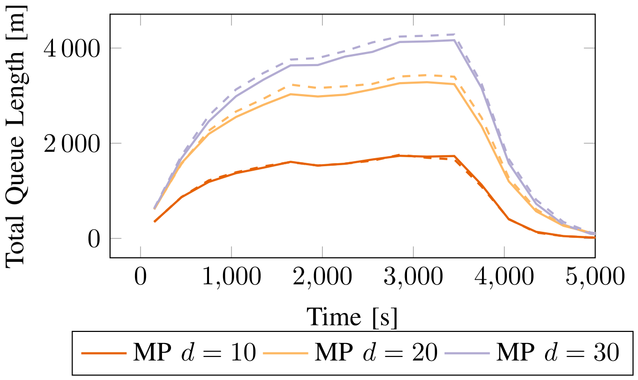

Fig. 7. How the total queue lengths vary over time in the Manhattan grid with the MaxPressure (MP) controller with correct turning ratios (solid) and wrong turning ratios (dashed). The demand is $\delta = 0.05$. To improve the readability of the results, the queue-lengths are averaged over 300 seconds intervals.

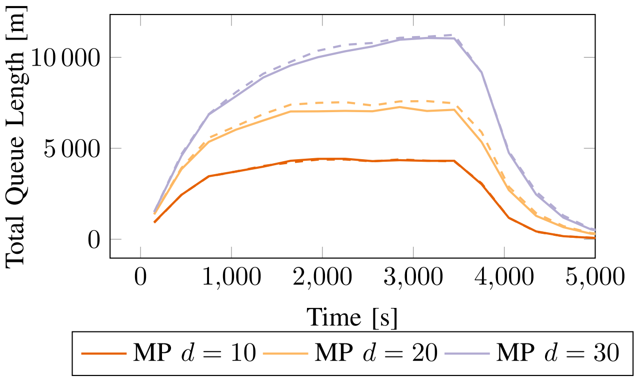

Fig. 8. How the total queue lengths vary over time in the Manhattan grid with the MaxPressure (MP) controller with correct turning ratios (solid) and wrong turning ratios (dashed). The demand is $\delta = 0.10$. To improve the readability of the results, the queue-lengths are averaged over 300 seconds intervals.

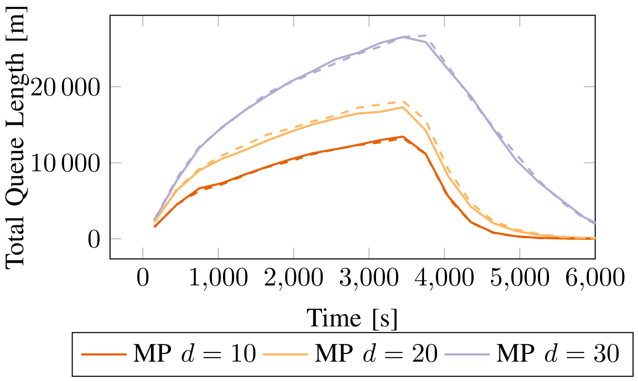

Fig. 9. How the total queue lengths vary over time in the Manhattan grid with the MaxPressure (MP) controller with correct turning ratios (solid) and wrong turning ratios (dashed). The demand is $\delta = 0.15$. To improve the readability of the results, the queue-lengths are averaged over 300 seconds intervals.

TABLE III
FIXED TIME AND PROPORTIONAL FAIR CONTROL - MANHATTAN SCENARIO

Controller	δ	Total Travel Time [h]
FT	0.05	1201
FT	0.10	2555
FT	0.15	4642
PF	0.05	1694
PF	0.10	4165
PF	0.15	$+\infty$

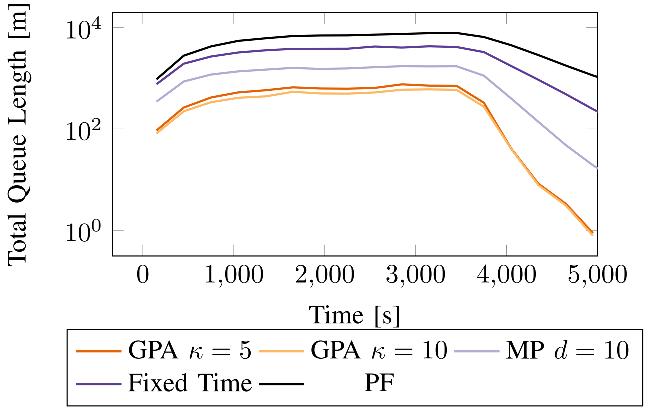

Fig. 10. A comparison between different control strategies for the Manhattan grid with the demand $\delta = 0.05$.o improve the readability of the results, the queue-lengths are averaged over 300 seconds intervals.

included for reference the total queue lengths for the fixed time controller and the proportional fairness controller. The total travel times for those controllers are given in Table III. When the demand is $\delta = 0.15$, a gridlock situation occurs with the proportional fairness controller, just as with the GPA controller with $\kappa = 1$. From the simulations we can conclude that, for this scenario and during high demands, the MaxPressure controller performs better than the GPA controller, while during low demands, the GPA performs better. One explanation for this could be that during low demands, adapting the cycle length is critical, while during high demands when almost all the sensors are covered, it is more important to keep the queue balanced between the current and downstream lanes. The proportional fairness controller that does not adapt its cycle length, always performs the worst. In most of the cases, a fixed time controller performs second worst. It is just for the demand $\delta = 0.15$, and during the draining phase that the fixed time controller performs better than the GPA controller.

V. SIMULATIONS IN THE LUST SCENARIO

To test the GPA controller in a realistic scenario, we make use of the Luxembourg SUMO Traffic (LuST) scenario presented in [28][1]. The scenario models the city center of Luxembourg during a full day, and the authors of [28] have made several adjustments from some given population data when creating the scenario, to make it as realistic and well-calibrated as possible. For more information about the calibration of this scenario, we refer the reader to [28].

[1] The scenario files are obtained from https://github.com/lcodeca/LuSTScenario/tree/v2.0

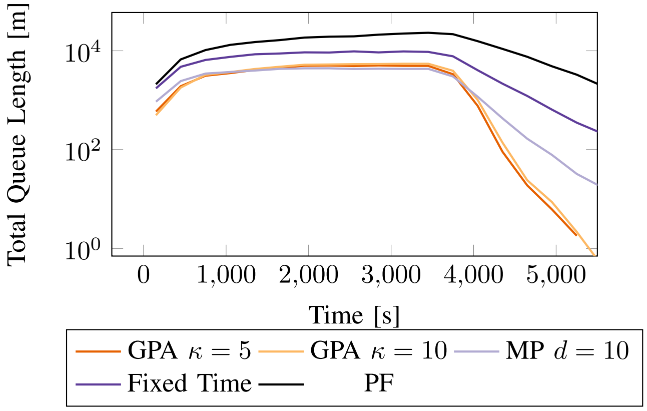

Fig. 11. A comparison between different control strategies for the Manhattan grid with the demand $\delta = 0.10$. To improve the readability of the results, the queue-lengths are averaged over 300 seconds intervals.

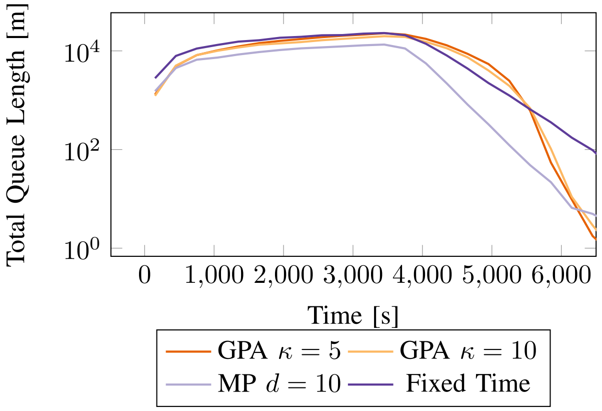

Fig. 12. A comparison between different control strategies for the Manhattan grid with the demand $\delta = 0.15$. Since the proportional fairness controller (PF) creates a gridlock, it is not included in the comparison. To improve the readability of the results, the queue-lengths are averaged over 300 seconds intervals.

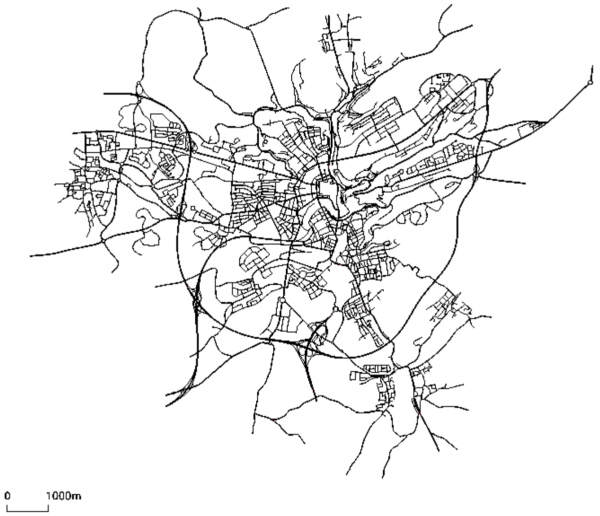

Fig. 13. The traffic network of Luxembourg city

The LuST network is shown in Figure 13. To each of the 199 signalized junctions, we have added a lane area detector to each incoming lane. The length of the detectors are 100 meters, or as long as the lane lasts if the lane is shorter than 100 meters. Those sensors are added to give the controller real-time information about the queue-lengths at each junction.

As input to the system, we are using the Dynamic User Assignment demand data. For this data-set, the drivers take their shortest path (with respect to travel time) between their current position and their destination. It is assumed that 70 percent of the vehicles can recompute their shortest path while driving, and will do so every fifth minute. This rerouting possibility is introduced to model the fact that more and more drivers are using online navigation with real-time traffic state information. If so, the drivers will get updates about what the optimal route choice is during their trips.

In the LuST scenario, the phases are constructed in a bit more complex way and are not always orthogonal. For non-orthogonal phases, it is not always the case that all lanes receive yellow light when a clearance phase is activated. If the lane receives a green light in the next phase as well, it will receive green light during the clearance phase too. Since we do not change the phase structure with respect to the original scenario, it means that just activate the phase following the phase just activated is not enough to make sure that all lanes receive a yellow light before a red light. For this reason, we choose to implement the controller which activates all the clearance phases in the cycle, i.e., the controller that is given in Section III-A and Section III-D.

As mentioned, the phases in the LuST scenario are not orthogonal in each junction. Hence we have to solve the convex optimization problem in (1) to compute the phase activation. The computation is done by using the solver CVXPY[2] in Python. Although the controller can be implemented in a distributed manner, the simulations in this paper are performed on a single computer. Despite the size of the network and the fact that the communication via TraCI between the controller written in Python and SUMO slows down the simulations significantly, the simulations are still running about 2.5 times faster than real-time on a regular CPU. The computation times suggest that there is no problem with running this controller in a real-time setting.

Since the demand is high during the peak hours in the scenario, gridlock situations occur. Those kinds of situations are unavoidable since there will be conflicts in the car-following model. To make the simulation continue to run, SUMO has a teleporting option that is utilized in the original LuST scenario. The original LuST scenario is configured in such a way that if a vehicle has been looked for more than 10 minutes, it will teleport along its route until there is free space. It is therefore important when we evaluate the control strategies that we keep track of the number of teleports, to make sure that the control strategy will not create a significantly larger amount of gridlocks, compared to the original fixed time controller. In Table IV, the number of teleports are reported for each controller. It is also reported how many of those teleports that

[2] https://cvxpy.org

TABLE IV
COMPARISON OF THE DIFFERENT CONTROL STRATEGIES

	κ	$\bar{w}$	Teleports (jam)	Total Travel Time [h]
GPA	10	0	76 (6)	49 791
GPA	10	0.05	65 (1)	49 708
GPA	10	0.10	37 (0)	49 519
GPA	10	0.15	57 (19)	49 408
GPA	10	0.20	50 (10)	49 380
GPA	10	0.25	35 (0)	49 265
GPA	10	0.30	30 (0)	48 930
GPA	10	0.35	25 (1)	48 922
GPA	10	0.40	51 (0)	48 932
GPA	10	0.45	49 (5)	49 076
GPA	10	0.50	42 (15)	49 383
GPA	5	0	668 (76)	57 249
GPA	5	0.05	234 (62)	54 870
GPA	5	0.10	68 (10)	52 038
GPA	5	0.15	47 (9)	50 696
GPA	5	0.20	50 (6)	49 904
GPA	5	0.25	41 (3)	49 454
GPA	5	0.30	23 (0)	48 964
GPA	5	0.35	30 (1)	48 643
GPA	5	0.40	35 (5)	48 445
GPA	5	0.45	39 (1)	48 503
GPA	5	0.50	42 (10)	48 772
Fixed time	–	–	122 (80)	54 103
Cyclic MP	$\eta = 0.05$		40 (0)	56 049
Cyclic MP	$\eta = 0.1$		55 (2)	55 367
Cyclic MP	$\eta = 0.5$		93 (7)	55 574
Cyclic MP	$\eta = 1$		1923 (1277)	61 566

are caused directly due to traffic jam, but one should have in mind that, e.g., a gridlock caused by two vehicles that want to swap lanes is often a consequence of congestion.

The total travel time and the number of teleports for different choices of tuning parameters are shown in Table IV. For the fixed time controller, we keep the standard fixed time plan provided with the LuST scenario. How the queue lengths vary with time for different $\bar{w}$ is shown in Figure 14 for $\kappa = 5$ and in Figure 15 for $\kappa = 10$.

From the results, we can see that any controller with $\kappa = 10$ and $\bar{w}$ within the range of investigation will improve the traffic situation. However, the controller that yields the overall shortest total travel time is the one with $\kappa = 5$ and $\bar{w} = 0.40$. This result suggests that tuning the GPA only with respect to κ, and keep $\bar{w} = 0$, may not lead to the best performance concerning total travel time, although it gives higher theoretical throughput.

For comparison, we have also implemented the Cyclic MaxPressure. For this controller, we are using the same cycle length in each junction as the provided for the Fixed Time controller, and as an estimate for the routing matrix, we assume that the vehicles split up equally among the downstream lanes. The results for different choices of η is shown in Table IV. From Table IV and Figure 14–15 it can be concluded that the Cyclic MaxPressure does not improve the traffic situation, while the GPA with the right choice of parameters does.

VI. CONCLUSION

In this paper, we have discussed the implementational aspects of the Generalized Proportional Allocation controller. The controller's performance was compared to the MaxPressure controller both on an artificial Manhattan-like grid and for a real scenario. In comparison with MaxPressure, it was shown that the controller performs better than the MaxPressure controller when the demand is low, but the MaxPressure performs better during high demand. Those observations hold even if the MaxPressure controller does not have correct information about the turning ratios in each junction.

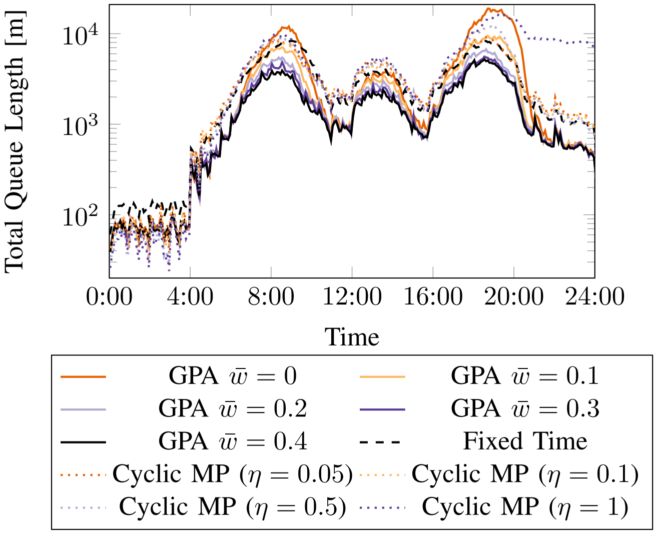

Fig. 14. How the queue lengths vary with time when the traffic signals in the LuST scenario are controlled with three different controllers: GPA controller, Cyclic MaxPressure (MP) controller, and the standard fixed-time controller. For the GPA controller the parameters $\kappa = 5$ and different values of $\bar{w}$ are tested. To improve the readability of the results, the queue-lengths are averaged over 300 seconds intervals.

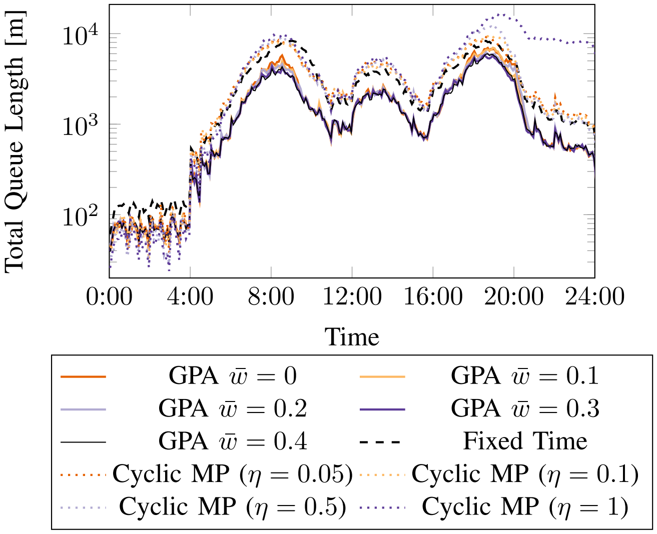

Fig. 15. How the queue lengths vary with time when the traffic lights in the LuST scenario are controlled with the GPA controller, Cyclic MaxPressure (MP) controller, and the standard fixed-time controller. For the GPA controller the parameters $\kappa = 10$ and different values of $\bar{w}$ are tested. To improve the readability of the results, the queue-lengths are averaged over 300 seconds intervals.

While the information about the turning ratios and the queue lengths at neighboring junctions are needed for the MaxPressure controller, the GPA controller does not require any such information. This makes the GPA controller easier to implement in a real scenario, where the downstream junction may not be signalized and equipped with sensors. We showed that it is possible to both implement the GPA controller in a realistic scenario covering the city of Luxembourg and that it improves the traffic situation compared to a standard fixed time controller. The GPA controller also performs better than the Cyclic MaxPressure controller.

In all simulations, we have used the same tuning parameters for all junctions in the LuST scenario, while the fixed time controller is different for different junction settings. Hence the GPA controller's performance can be improved even more by tuning the parameters individually for each junction. Ideally, this should be done with some auto-tuning solution, but it may also be worth to take static parameters into account, such as the sensor lengths. This is a topic for future research.

References

[1] R. Mohajerpoor, M. Saberi, and M. Ramezani, "Analytical derivation of the optimal traffic signal timing: Minimizing delay variability and spillback probability for undersaturated intersections," *Transportation Research Part B: Methodological*, vol. 119, pp. 45 – 68, 2019.

[2] 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.

[3] V. Mauro and C. Di Taranto, "Utopia," in *Control, computers, communications in transportation*, pp. 245–252, Elsevier, 1990.

[4] A. G. Sims and K. W. Dobinson, "The Sydney coordinated adaptive traffic (SCAT) system philosophy and benefits," *IEEE Transactions on vehicular technology*, vol. 29, no. 2, pp. 130–137, 1980.

[5] J. Jin and X. Ma, "A decentralized traffic light control system based on adaptive learning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5301 – 5306, 2017. 20th IFAC World Congress.

[6] D. Srinivasan, M. C. Choy, and R. L. Cheu, "Neural networks for real-time traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, pp. 261–272, Sep. 2006.

[7] P. LA and S. Bhatnagar, "Reinforcement learning with function approximation for traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 412–421, June 2011.

[8] A. Kouvelas, K. Aboudolas, M. Papageorgiou, and E. B. Kosmatopoulos, "A hybrid strategy for real-time traffic signal control of urban road networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 884–894, Sep. 2011.

[9] B. P. Gokulan and D. Srinivasan, "Distributed geometric fuzzy multiagent urban traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 714–727, Sep. 2010.

[10] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[11] G. Nilsson, P. Hosseini, G. Como, and K. Savla, "Entropy-like Lyapunov functions for the stability analysis of adaptive traffic signal controls," in *The 54th IEEE Conference on Decision and Control*, pp. 2193–2198, 2015.

[12] G. Nilsson and G. Como, "On generalized proportional allocation policies for traffic signal control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9643–9648, 2017.

[13] G. Como, E. Lovisari, and K. Savla, "Throughput optimality and overload behavior of dynamical flow networks under monotone distributed routing," *IEEE Transactions on Control of Networked Systems*, vol. 2, no. 1, pp. 57–67, 2015.

[14] G. Como, "On resilient control of dynamical flow networks," *Annual Reviews in Control*, vol. 43, pp. 80 – 90, 2017.

[15] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[16] R. P. Roess, E. S. Prassas, and W. R. McShane, *Traffic engineering*. Prentice Hall, 2011.

[17] P. Varaiya, "The max-pressure controller for arbitrary networks of signalized intersections," in *Advances in Dynamic Network Modeling in Complex Transportation Systems*, pp. 27–66, Springer, 2013.

[18] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[19] A. A. Zaidi, B. Kulcsár, and H. Wymeersch, "Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 2134–2143, Aug 2016.

[20] S. Coogan, C. Flores, and P. Varaiya, "Traffic predictive control from low-rank structure," *Transportation Research Part B: Methodological*, vol. 97, pp. 1–22, 2017.

[21] Z. Hao, R. Boel, and Z. Li, "Model based urban traffic control, part i: Local model and local model predictive controllers," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 61 – 81, 2018.

[22] Z. Hao, R. Boel, and Z. Li, "Model based urban traffic control, part ii: Coordinated model predictive controllers," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 23 – 44, 2018.

[23] P. Grandinetti, C. Canudas-de Wit, and F. Garin, "Distributed optimal traffic lights design for large-scale urban networks," *IEEE Transactions on Control Systems Technology*, 2018.

[24] G. Bianchin and F. Pasqualetti, "A network optimization framework for the analysis and control of traffic dynamics and intersection signaling," in *57th IEEE Conference on Decision and Control*, pp. 1017–1022, Dec 2018.

[25] G. Nilsson and G. Como, "Evaluation of decentralized feedback traffic light control with dynamic cycle length," *IFAC-PapersOnLine*, vol. 51, no. 9, pp. 464–469, 2018.

[26] T. Le, P. Kovács, N. Walton, H. L. Vu, L. L. Andrew, and S. S. Hoogendoorn, "Decentralized signal control for urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 431–450, 2015.

[27] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, December 2012.

[28] L. Codecá, R. Frank, S. Faye, and T. Engel, "Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 2, pp. 52–63, 2017.

Gustav Nilsson received his M.Sc. in Engineering Physics and Ph.D. in Automatic Control from Lund University in 2013 and 2019, respectively. He is currently a Postdoctoral Associate at Georgia Institute of Technology, GA, USA. During his PhD studies, he has been a visiting researcher at the Institute of Pure and Applied Mathematics (IPAM), UCLA, CA, USA and at Department of Mathematical Sciences, Politecnico di Torino, Turin, Italy. Between October 2017 and March 2018, he did an internship at Mitsubishi Electric Research Laboratories in Cambridge, MA, USA. His primary research interest lies in modeling and control of dynamical flow networks with applications in transportation networks.

Giacomo Como is an Associate Professor at the Department of Mathematical Sciences, Politecnico di Torino, Italy, and at the Automatic Control Department of Lund University, Sweden. He received the B.Sc., M.S., and Ph.D. degrees in Applied Mathematics from Politecnico di Torino, in 2002, 2004, and 2008, respectively. He was a Visiting Assistant in Research at Yale University in 2006-2007 and a Postdoctoral Associate at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, from 2008 to 2011. He currently serves as Associate Editor of IEEE-TCNS and IEEE-TNSE and as chair of the IEEE-CSS Technical Committee on Networks and Communications. He was the IPC chair of the IFAC Workshop NecSys'15 and a semiplenary speaker at the International Symposium MTNS'16. He is recipient of the 2015 George S. Axelby Outstanding Paper Award. His research interests are in dynamics, information, and control in network systems with applications to cyber-physical systems, infrastructure networks, as well as social and economic networks.