

# Two-layer hierarchical control for large-scale urban traffic networks\*

Anastasios Kouvelas<sup>1</sup>, Dimitris Triantafyllos<sup>2</sup>, Nikolas Geroliminis<sup>1</sup>

**Abstract**—Many efforts have been carried out to optimize the traffic signal settings in cities. Nevertheless, state-of-the-art and -practice strategies cannot deal efficiently with oversaturated conditions (i.e. queue spillbacks and partial gridlocks), as they are either based on application-specific heuristics or they fail to replicate accurately the propagation of congestion. An alternative approach for real-time network-wide control is the perimeter flow control (or gating). This can be viewed as an upper-level control layer, and be combined with other strategies (e.g. local or coordinated regulators) in a hierarchical control framework. In the current work, a recently developed perimeter control regulator is utilized for the upper-level layer. Another lower-level control layer utilizes the max-pressure regulator, which constitutes a local feedback control law, applied in coupled intersections, in a distributed systems-of-systems (SoS) concept. Different approaches are discussed about the design of the hierarchical structure of SoS and a traffic microsimulation tool is used to assess the impact of each approach to the overall traffic conditions. Preliminary results show that integrating a network-level approach within a local adaptive framework can significantly improve the system performance when spillback phenomena occur (a common feature of city centres with short links).

## I. INTRODUCTION

In recent years, most big cities around the world become denser and wider, and due to the lack of space for building new infrastructure, the problem of urban traffic management is steadily gaining momentum. During the peak hours, traffic networks face serious congestion problems and the performance of the infrastructure degrades significantly. Many efforts have been carried out about the optimization of traffic signal settings. Nevertheless, state-of-the-art and -practice strategies cannot deal efficiently with oversaturated conditions (i.e. queue spillbacks and partial gridlocks), as they are either based on application-specific heuristics or they fail to replicate accurately the propagation of congestion.

An alternative approach for real-time network-wide control, that has recently gained a lot of interest in the literature, is the perimeter flow control (or gating). The basic concept of such an approach is to partition heterogeneous large-scale cities into a small number of homogeneous regions (zones) and apply perimeter control to the inter-regional

flows along the boundaries between regions. The inter-transferring flows are controlled at the intersections located along the borders between zones, so as to distribute the congestion in an optimal way and minimize the total delay of the system. Previous research [1], has shown that the master-slave concept may arise, as some region can be “sacrificed” and led to congested states in favour of other regions that are more “important” for the total performance of the system. This can be viewed as an upper-level control layer, and be combined with other strategies (e.g. local or coordinated regulators) in a hierarchical control framework.

Local adaptive strategies might work efficiently if conditions are not severely congested, but in case of very high demand and short intersections, the fast dynamics of congestion might result in spillbacks and local close-to-gridlock conditions. Perimeter control can avoid this phenomena by preventing overcrowding for critical regions of the city. Nevertheless, as it controls only a subset of the intersections, it can create longer queues in the controlled boundary, while strong heterogeneity might arise inside the regions, which will have as a result to decrease the regional outflow. Local strategies can be complementary to the perimeter control and avoid local pockets of congestion.

In the current work we focus on the development of hierarchical control structures to tackle the problem described above and we study different architecture designs. A recently developed perimeter control regulator, which integrates model-based optimal control and online data-driven learning/adaptation, is utilized for the upper-level layer. Another lower-level control layer utilizes the max-pressure regulator, which has been also proposed recently and constitutes a local feedback control law ([2], [3]), applied in coupled intersections, in a distributed systems-of-systems (SoS) concept. Max-pressure has been proved to be among the most efficient controllers if the system can be stabilized, but in case of oversaturated conditions and short links, it might face difficulties.

Max-pressure controller will also require a large number of sensors to accurately measure queue lengths in all intersections in the network, which would increase the implementation cost. We show in this paper that if max-pressure for a subset of critical intersections is combined with perimeter control, the results in terms of total delays for the system can significantly improve. Different approaches are discussed about the design of the hierarchical structure of SoS, i.e. mutual interactions between the two control layers, activation/deactivation of each layer, mutually related objectives of the regulators, online versus offline selection of critical intersection for the lower-level control layer.

\*This work was supported by ERC (European Research Council) Starting Grant “META-FERW: Modelling and controlling traffic congestion and propagation in large-scale urban multi-modal networks” (Grant #338205)

<sup>1</sup>Anastasios Kouvelas and Nikolas Geroliminis are with Urban Transport Systems Laboratory (LUTS), School of Architecture, Civil & Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland {tasos.kouvelas, nikolas.geroliminis}@epfl.ch

<sup>2</sup>Dimitris Triantafyllos is with TSS – Transport Simulation Systems, S.L., Ronda de la Universitat 22 B, 08007 Barcelona, Spain dimitris.triantafyllos@aimsun.com

A traffic microsimulation tool is utilized in order to assess the impact of each hierarchical control design to the overall traffic conditions. An urban network is simulated with a realistic OD matrix and different hierarchical control schemes are compared to the default signal settings that are currently used in the field. Each approach is also evaluated in comparison to a scenario of distributed control in all the intersections or standalone perimeter control. Simulation investigations demonstrate the advantages of hierarchical control structures over different other disconnected regulator schemes. Preliminary results show that integrating a network-level approach within a local adaptive framework can significantly improve the system performance when spillback phenomena occur (a common feature of city centres with short links). An efficient multi-layer control design can significantly improve traffic congestion, leading to lower delays and higher production for the system. This has positive economic and environmental implications and large-scale cities are able to provide better quality of service to the users of the infrastructure.

One work in a similar direction is presented in [4] where the authors combine perimeter control (or gating) with local traffic-responsive control for intersections inside the protected region. The difference in the current work is that we propose a methodology for the selection of intersections and also the applied local control is max-pressure. Finally, some recent works ([5], [6]) use similar control techniques for route reservation approaches.

The next section presents the methodological part of this work. The hierarchical control framework is described and each of the two levels is presented in details. The subsections describe the upper level controller (i.e. perimeter regulator that is applied at the boundaries between regions) and the lower level distributed controller (i.e. max-pressure) which is applied to the intersections inside the network regions. Section III presents some preliminary simulation results for different control scenarios. Finally, the last section concludes the paper with some closing remarks about this work.

## II. HIERARCHICAL TRAFFIC SIGNAL CONTROL

This section describes the hierarchical scheme that it is used in this work to control the intersections of an urban network. The scheme consists of two mutually exclusive but interactive layers. The first upper layer (perimeter control) has more macroscopic characteristics, as it reflects the modelling of the network as a multi-region systems and controls the flows that transfer among the regions. The second one has a more microscopic flavour, as it models all the details of local intersections and acts on a link level basis to prevent local congestion phenomena. Both layers are applied simultaneously (with equal control cycles), and, although they do not exchange any kind of information, they are correlated as they have similar objectives.

### A. Notations

The arterial network is represented as a directed graph with links  $z \in Z$  and nodes  $n \in N$ . For each signalized intersection  $n$ , we define the sets of incoming  $I_n$  and outgoing  $O_n$  links.

It is assumed that the offsets and the cycle time  $C_n$  of node  $n$  are fixed or calculated in real-time by another algorithm. In addition, to enable network offset coordination, it is quite usual to assume that  $C_n = C$  for all intersections  $n \in N$  but this is not the case here as the coordination problem is not considered. The signal control plan of node  $n$  (including the fixed lost time  $L_n$ ) is based on a fixed number of stages that belong to the set  $F_n$ , wherein  $v_j$  denotes the set of links that receive right of way at stage  $j \in F_n$ . Finally, the saturation flow  $S_z$  of link  $z \in Z$  and the turning movement rates  $\beta_{i,w}$ , where  $i \in I_n$  and  $w \in O_n$ , are assumed to be known and can be constant or time varying.

By definition, the constraint

$$\sum_{j \in F_n} g_{n,j}(k) + L_n = (\text{or } \leq) C_n \quad (1)$$

holds for every node  $n$ , where  $k = 0, 1, 2, \dots$  is the control discrete-time index and  $g_{n,j}$  is the green time of stage  $j$ . Inequality in equation 1 may be useful in cases of strong network congestion to allow for all-red stages (e.g. for strong local gating). In addition, the constraint

$$g_{n,j}(k) \geq g_{n,j,\min}, \quad j \in F_n \quad (2)$$

where  $g_{n,j,\min}$  is the minimum permissible green time for stage  $j$  in node  $n$  and is introduced in order to guarantee allocation of sufficient green time to pedestrian phases. The control variables of the problem are  $g_{n,j}(k)$  and depict the effective green time of every stage  $j \in F_n$  of every intersection  $n \in N$ .

### B. Framework description

Figure 1 describes the basic principle of the hierarchical control framework. Basically, we have two distinct controllers that act independently, without communicating or exchanging information. At each control cycle (which is the same for both layers, e.g. 90 seconds) the controllers receive all the required measurements from the system and apply their equations. The two layers run in parallel and the decisions of each layer are not affected by the other. Each layer has its own objectives that are related to the global (upper) or the local (lower) performance of the network. They

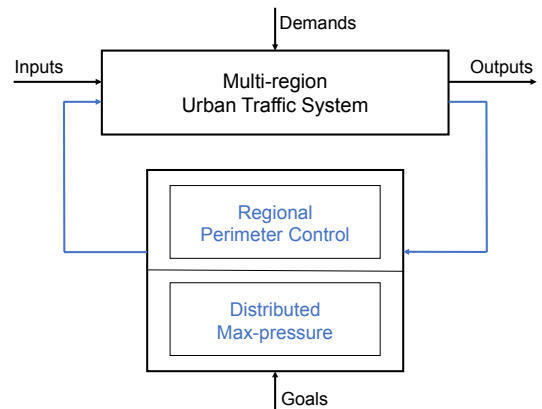


Fig. 1. Schematic description of the hierarchical control framework.

utilize different kind of input data (aggregated accumulations versus local queue measurements) and the green times that are calculated are applied to different signalised intersections. To this end, the intersections that are selected for the upper control layer (perimeter control) are excluded from the set of intersections to be studied for the application of the lower control method (max-pressure). The details of each regulator are explained in details in the next subsections.

### C. Upper level controller

For the upper control level (i.e. regional perimeter control) a multivariable feedback PI regulator from the literature is utilized. The regulator is introduced in [7], [1] and controls the transferring flows between the antagonistic regions in an optimal way (i.e. so as to minimize the total delay of the system). Based on the distribution of vehicles in each region and an objective criterion about the total performance of the system one can regulate the flows that try to transfer from one region to all its neighbouring regions. The actuators are all the traffic lights that are located in the boundaries between regions and this can create inter-regional queues that affect the homogeneity of congestion. This is the reason that we apply here the lower level controller, i.e. to homogenise the traffic and improve the queues and congestion inside each region. Other works in the literature that utilize similar perimeter control structures can be found in [8], [9], [10], [11].

1) *Regulator for perimeter control:* To this end, the regional multivariable feedback PI regulator derived in [1] reads in vector form

$$\mathbf{u}(k) = \mathbf{u}(k-1) - \mathbf{K}_P [\mathbf{n}(k) - \mathbf{n}(k-1)] - \mathbf{K}_I [\mathbf{n}(k) - \hat{\mathbf{n}}] \quad (3)$$

where  $k = 0, 1, 2, \dots$  is the discrete control time index,  $\mathbf{u}(k) \in \mathbb{R}^M$  denotes the vector with the control inputs (i.e. all transferring flows from any region  $i$  to all neighbouring regions  $j$ ),  $\mathbf{n}(k) \in \mathbb{R}^N$  denotes the state vector of all region accumulations, and  $\hat{\mathbf{n}}$  is the vector with the set points for each region  $i$ . Finally,  $\mathbf{K}_P, \mathbf{K}_I \in \mathbb{R}^{M \times N}$  are the proportional and integral gains of the regulator, which are computed by the solution of the corresponding discrete-time Riccati equation of the system and then fine-tuned in real-time, as demonstrated in [1].

The state feedback regulator (3) is activated in real-time at each control interval  $T$  and only within specific time windows based on the current accumulations  $\mathbf{n}(k)$  (i.e. by use of two thresholds  $n_{i,\text{start}}$  and  $n_{i,\text{stop}}$ <sup>1</sup> and real-time measurements). The required real-time information of the vehicle accumulations  $\mathbf{n}(k)$  can be directly estimated via loop detector time-occupancy measurements. Furthermore, in cases where only sparse measurements are available, different approaches to estimate MFD related state variables with real data are described in [12], [13] and [14].

<sup>1</sup>In practical applications usually  $n_{i,\text{stop}} < n_{i,\text{start}}$  is selected in order to avoid frequent activations/deactivations of the controller.

### D. Lower level controller

In this section the max-pressure control for arterial networks is introduced. This decentralized controller does not require any knowledge of the mean current or future demands of the network (in contrast to other model predictive control frameworks). Max-pressure stabilizes the network if the demand is within certain limits, thus it maximizes network throughput. However, it does require knowledge of mean turn ratios and saturation rates, albeit an adaptive version of max-pressure will have the same performance, if turn movements and saturation rates can be measured. It only requires local information at each intersection and provably maximizes throughput [2]. Several variations of the basic method that can be applied in real-time (depending on the available infrastructure) are presented.

1) *Max-pressure:* The state of each link  $x_z(k)$  is defined by the number of vehicles waiting in the queue to be served for each control index  $k$  (i.e. at the beginning of time period  $[kC_n, (k+1)C_n]$ ). Given that we are provided with real-time measurements or estimates of all the states we can compute the pressure  $p_z(k)$  that each link exerts on the corresponding stage of node  $n$  at the beginning of cycle  $k$  as follows

$$p_z(k) = \left[ \frac{x_z(k)}{x_{z,\text{max}}} - \sum_{w \in O_n} \frac{\beta_{z,w} x_w(k)}{x_{w,\text{max}}} \right] S_z, \quad z \in I_n \quad (4)$$

where  $x_{z,\text{max}}$  is the storage capacity of link  $z$  (in vehicles). Storage capacity is used in the denominator of equation 4 in order to take into account the length of the links, so that the pressure of a short link with a number of vehicles waiting to be served is higher than the pressure of a longer link with the same number of vehicles. The measurements (or estimates)  $x_z(k), \forall z \in Z$  represent a feedback from the network under control, based on which the new pressures are calculated via equation 4 in real-time.

The pressure of link  $z$  during the control cycle  $k$  is the queue length of the link (first term in equation 4 within the brackets) minus the average queue length of all the output links (second term in equation 4 within the brackets). Regarding the second term as the (average) downstream queue length and the first as the upstream queue length, the definition of the pressure is simply the difference between the upstream and downstream queue lengths. It should be noted, that in the case where all output links are exiting the network (we assume that exit links have infinite capacity, i.e., they do not experience any downstream blockage), the second term in equation 4 becomes zero. Hence, the pressure of the link is simply the queue length multiplied by its corresponding saturation rate. Note that in this paper we select the second term to be always equal to 0, i.e. we do not consider the downstream queues but only the upstream (demand) queues. This is a variation of max-pressure that only looks at the queues of local intersections and not the downstream destinations of the traffic streams.

If equation 4 is applied  $\forall z \in I_n$  the pressures of all incoming links of node  $n$  are calculated. The pressure of each stage  $j$  of the intersection can then be computed as

follows

$$P_{n,j}(k) = \max \left\{ 0, \sum_{z \in v_j} p_z(k) \right\}, \quad j \in F_n \quad (5)$$

and this metric can be used to calculate the splits for the different conflicting stages of the intersection.

2) *Green time calculation*: Given that the pressure of each stage has been computed by equation 5, the total effective green time  $G_n$  that is available to be distributed in node  $n$

$$G_n = C_n - L_n - \sum_{j \in F_n} g_{n,j,\min}, \quad n \in N \quad (6)$$

can be split to all stages in many different ways. One approach, is to select the stage with the maximum pressure and activate it for the next control cycle  $C_n$ . This implies that all the available effective green time  $G_n$  will be given to this stage. In the next cycle the queues of the system are updated, the new pressures are calculated and the stage with the maximum pressure is selected to be activated and so forth. This approach may not be the optimal one, as the control cycle may be large and queues can grow unexpectedly at the links that are not activated. Alternatively, max-pressure can be called several times within a cycle  $C_n$ . Every time the stage with the maximum pressure is activated, however, the frequency of the measurements/control is now higher. The frequency of max-pressure application to an intersection depends on two main factors: (a) the available infrastructure and communications (i.e. the appropriate measurements or estimates of queue lengths should be provided in real-time), and (b) an optimal frequency of max-pressure application which needs to be investigated and defined (and could be dependent on the special characteristics of each site).

Another approach that has been proposed in [3] is utilized here, which is to call max-pressure at the end of each cycle and split the green time  $G_n$  proportionally to the computed pressure of each stage. That is, for each decision variable  $\tilde{g}_{n,j}(k)$  (where  $\tilde{g}_{n,j}$  depicts the green time of stage  $j$  on the top of  $g_{n,j,\min}$ ) the following update rule is applied

$$\tilde{g}_{n,j}(k) = \frac{P_j(k)}{\sum_{i \in F_n} P_i(k)} G_n, \quad j \in F_n \quad (7)$$

Thus, the total amount of green time allocated for each control variable  $g_{n,j}(k)$  for cycle  $k$  is given by

$$g_{n,j}(k) = \tilde{g}_{n,j}(k) + g_{n,j,\min}, \quad j \in F_n \quad (8)$$

Finally, the following constraint is applied for operational reasons

$$|g_{n,j}(k) - g_{n,j}(k-1)| \leq g_{n,j}^R, \quad j \in F_n \quad (9)$$

where  $g_{n,j}^R$  is a user-defined parameter that is used to make sure that there are no big fluctuations in the green times between two consecutive cycles.

The procedure explained above is repeated periodically (for every cycle) and requires minimum communication specifications, as the local controller is called once per cycle. This local regulator is applied to the lower level of the hierarchical control structure to the simulation results presented in the next section.

### III. SIMULATION EXPERIMENTS AND PRELIMINARY RESULTS

This section presents some preliminary results for the application of the two-level hierarchical framework to microsimulation. First of all, an analysis is conducted to the data of all the intersections of the network in order to define the critical ones that max-pressure is going to be applied. All the network contains 565 signalised intersections and the purpose of this analysis is to select 10-15 intersections for each region that are the most important. From an implementation point of view it is considered too expensive and time consuming to control all the intersections, and as a result a methodology is needed that can define the critical intersections for each region.

#### A. Network description

In the current work, we use as a case study network a replica of the city center of Barcelona in Spain. For this network, we have a well calibrated microsimulation model in Aimsun (Figure 2(a)), which is used for the simulation experiments. The purpose of the study is to run different control scenarios and investigate the effect of the controllers based on the statistics gathered from the microsimulation engine and empirical observations from the simulations (e.g. network load, traffic congestion, local gridlocks, etc.).

Figure 2(b) presents the test network partitioned in 4 homogeneous regions. For the partitioning the algorithm presented in [15] has been used. The result of this algorithm is to get 4 clusters (zones) that are as homogeneous as possible and with compact shapes. In this multi-region system perimeter control can be applied to regulate the inter-transferring flows between the regions. The methodology presented in the previous section is applied here. The perimeter controller acts on the intersections located in the boundaries between the four regions.

#### B. Selection of the critical intersections

In order to apply the lower level controller (max-pressure) the critical intersections of every region need to be selected first. In order to do so, we need a methodology to rank the intersections according to their importance and decide which are the most critical. These intersections are controlled then based on the max-pressure control scheme that has been described earlier.

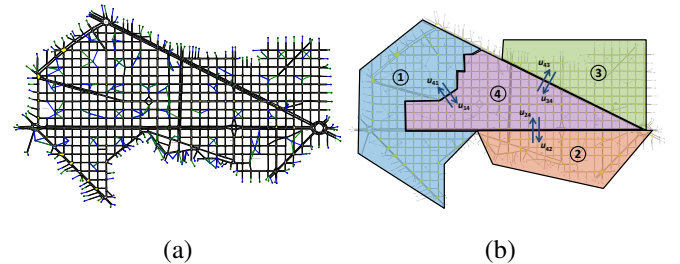


Fig. 2. (a) Aimsun microsimulation model for the CBD of Barcelona; (b) network partitioned in 4 homogeneous regions and acting perimeter controllers.

To this end we do an analysis of the simulation data for the fixed-time plan of the traffic lights and for the case where perimeter control is applied. The first scenario is called fixed-time control (FTC) and it reflects all the pre-timed signal plans of the city of Barcelona, while the latter is called perimeter control (PC). For both control scenarios and for a realistic OD demand profile we run a replication and collect all the detectors data for occupancy and flow measurements. This data is a replication of what we can obtain in the real world and represent proxies of the network density and flow respectively.

Here, we utilize occupancy measurements by the detectors to estimate the number of vehicles that exist in every link of the network and thus calculate a proxy of the queue length. By summing up over all the input links of an intersection, we can perform an analysis of the level of congestion as well as the homogeneity of each intersection (which is actually analogous to the variance of the queues). If we perform this analysis to our data and we average over time we can get a good flavour of the importance of each intersection and how critical it is to the propagation of congestion in the network. The more congested an intersection is the more critical for the overall congestion. Nevertheless, in order for the perimeter control to be more effective, the traffic around an intersection needs to be as homogeneous as possible. A level of homogeneity is the variance of the queues for all the incoming links.

Figure 3 presents the results of this analysis for the FTC scenario. The x-axis represents the variance of all the queues around an intersection (averaged over time) and the y-axis the level of congestion for all the input links to the studied intersection (averages over space and time). Every point in these graphs represents an intersection, i.e. region 1 has 158, region 2 has 60, region 3 has 76, and region 4 has 227 intersections. These results refer to the 1.5 hours of peak traffic after excluding the beginning and end of simulation where the traffic loads are very low. The results presented in this figure are for the fixed-time control strategy where the

pre-timed plans are applied in all intersections.

Figure 4 presents the same analysis for the scenario where perimeter control is applied to the city (PC scenario). The perimeter controller acts in the boundaries of the regions and all the measurements of the network are collected. In both plots we make a classification of intersections to three categories, i.e. critical, busy, and uncongested. The critical intersections are the outliers (outside the red dashed line) and correspond to junctions that have either big variance of queue measurements or high congestion levels. The busy intersections are in the band between the red and green dashed lines and they are intersections that carry heavy loads of traffic and can become critical easily. Finally, the uncongested intersections are below the green dashed line and are the ones that do not carry a lot of traffic and it is quite difficult for them to become critical. After this analysis, we choose to run the lower level controller (max-pressure) in all the critical intersections of each scenario.

Figures 5(a) and (b) present a spacial visualization of the above analysis on the network level. One can see the critical intersections (red) and the busy intersections (orange) for each scenario. The uncongested intersections are not marked with any color. Nevertheless, it is interesting to observe by comparing the two figures that after the application of perimeter control the number of critical intersections in regions 1 and 4 is reduced, and, equivalently, in regions 2 and 3 is increased. This happens because the perimeter controller achieves a more homogeneous distribution of congestion in the four regions and the classification thresholds do not change. Moreover, the number of total busy intersections is reduced significantly after the application of perimeter control, something that demonstrates the superiority of PC over FTC as a control scenario.

### C. Comparing the fixed-time case with other simulated scenarios

In this section the simulation results for the different control scenarios are presented. As mentioned earlier, for

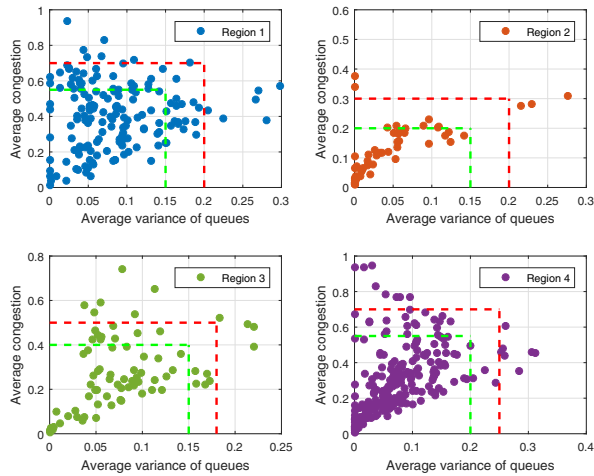


Fig. 3. Selection of critical intersections for the case that fixed-time control is applied to the network.

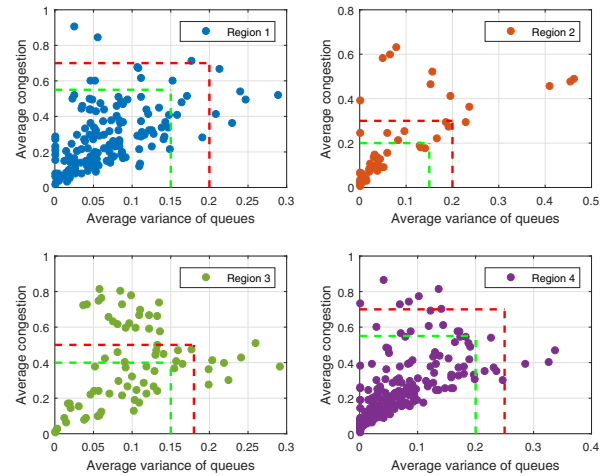


Fig. 4. Selection of critical intersections for the case that perimeter control is applied to the network.



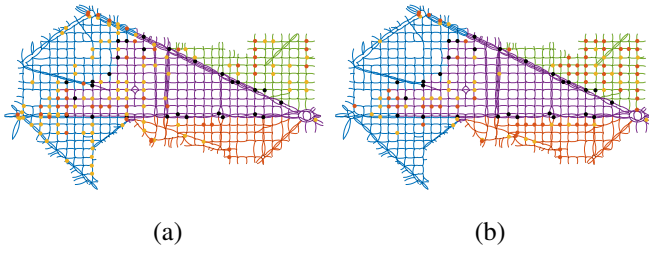


Fig. 5. Data for the partitioned network (a) FTC scenario, (b) PC scenario. Black dots: intersections that run perimeter controller; red dots: critical intersections that run max-pressure; orange dots: busy intersections that can become critical.

the fixed-time control scenario (FTC), the actual pre-timed plans of the traffic control center of the city are applied in all the intersections. Table I presents all the statistics that we obtain from the simulation software for this scenario and for 6 hours of simulated traffic. The other control scenarios that are presented for comparison are HC1, where the hierarchical framework is applied with max-pressure controller applied to all the critical intersections of Figure 5(a), and HC2, where max-pressure is applied to the critical intersections of Figure 5(b). It is clear that the hierarchical control framework improves the statistics (e.g. some 20-25% improvement in the total delay). Moreover, when perimeter control is first applied and then we do the selections of the critical intersections the simulation results are slightly better than selecting the intersections for the FTC case.

Finally, Figure 6 demonstrates a comparison of the trajectories of vehicle accumulations for the FTC and HC2 scenarios. The difference between the solid and dashed lines represents an approximation of the total delay improvement for each region. Regions 1 and 4 improve significantly while regions 2 and 3 slightly deteriorate. The hierarchical control scheme manages to improve the overall traffic conditions in terms of congestion and queue spillbacks.

#### IV. CONCLUSIONS

A hierarchical framework has been presented which consists of two existing controllers in the literature. The two controllers are combined in two different layers, the one applying a macroscopic approach (perimeter control) and the other an approach at the microscopic level (max-pressure controller). Another contribution of this paper is a methodology to select critical intersections inside urban regions. Simulation data is studied and an analysis is performed about the level of congestion as well as the homogeneity of local intersections. This methodology can define which intersections are crucial about the global traffic conditions and the propagation of congestion. The results of different control schemes remains to be compared via microsimulation in order to assess the different approaches and evaluate the hierarchical framework.

#### REFERENCES

[1] A. Kouvelas, M. Saeedmanesh, and N. Geroliminis, "Enhancing model-based feedback perimeter control with data-driven online adaptive optimization," *Transportation Research Part B*, vol. 96, pp. 26–45, 2017.

TABLE I

PERFORMANCE INDICATORS FOR DIFFERENT SIMULATED SCENARIOS.

Evaluation criteria	FTC	HC1	HC2	Units
Delay	539.48	432.22	412.15	sec/km
Space-mean Speed	5.96	7.25	7.55	km/h
Stop Time	450.52	353.71	334.36	sec/km
Total Travel Time	73923.62	56224.3	54363.82	h
Total Travelled Distance	425745.89	398687.84	396924.88	km
Vehicles Served	201706	201706	201706	veh

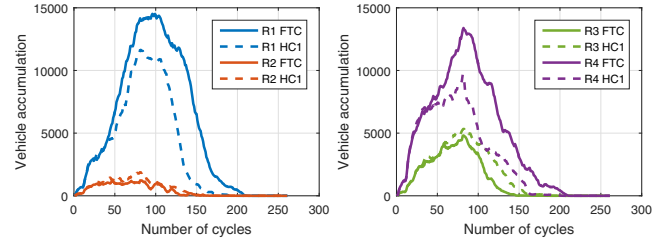


Fig. 6. Time-series of vehicle accumulations for the four regions and the HC1 control scenario.

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

[3] A. Kouvelas, J. Lioris, S. A. Fayazi, and P. Varaiya, "Maximum pressure controller for stabilizing queues in signalized arterial networks," *Transportation Research Record*, no. 2421, pp. 133–141, 2014.

[4] M. Keyvan Ekbatani, X. Gao, V. Gayah, and V. L. Knoop, "Combination of traffic-responsive and gating control in urban networks: Effective interactions," in *Transportation Research Board 95th Annual Meeting*, 2016.

[5] C. Menelaou, S. Timotheou, P. Kolios, and C. G. Panayiotou, "Improved road usage through congestion-free route reservations," *Transportation Research Record*, no. 2621, pp. 71–80, 2017.

[6] C. Menelaou, P. Kolios, S. Timotheou, C. Panayiotou, and M. Polycarpou, "Controlling road congestion via a low-complexity route reservation approach," *Transportation Research Part C*, vol. 81, pp. 118–136, 2017.

[7] A. Kouvelas, M. Saeedmanesh, and N. Geroliminis, "Feedback perimeter control for heterogeneous urban networks using adaptive optimization," in *18th IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 882–887.

[8] M. Keyvan-Ekbatani, A. Kouvelas, I. Papamichail, and M. Papageorgiou, "Exploiting the fundamental diagram of urban networks for feedback-based gating," *Transportation Research Part B*, vol. 46, no. 10, pp. 1393–1403, 2012.

[9] N. Geroliminis, J. Haddad, and M. Ramezani, "Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 348–359, 2013.

[10] K. Aboudolas and N. Geroliminis, "Perimeter and boundary flow control in multi-reservoir heterogeneous networks," *Transportation Research Part B*, vol. 55, pp. 265–281, 2013.

[11] M. Ramezani, J. Haddad, and N. Geroliminis, "Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control," *Transportation Research Part B*, vol. 74, pp. 1–19, 2015.

[12] L. Leclercq, N. Chiabaut, and B. Trinquier, "Macroscopic fundamental diagrams: A cross-comparison of estimation methods," *Transportation Research Part B*, vol. 62, pp. 1–12, 2014.

[13] J. Ortigosa, M. Menendez, and H. Tapia, "Study on the number and location of measurement points for an MFD perimeter control scheme: a case study of Zurich," *EURO Journal on Transportation and Logistics*, vol. 3, no. 3-4, pp. 245–266, 2014.

[14] K. Ampountolas and A. Kouvelas, "Real-time estimation of critical vehicle accumulation for maximum network throughput," in *American Control Conference*, 2015, pp. 2057–2062.

[15] M. Saeedmanesh and N. Geroliminis, "Clustering of heterogeneous networks with directional flows based on "Snake" similarities," *Transportation Research Part B*, vol. 91, pp. 250–269, 2016.