Two-level hierarchical control for large-scale urban traffic networks

Anastasios Kouvelas

Nikolas Geroliminis

École Polytechnique Fédérale de Lausanne **May 2017**

STRC | **17th Swiss Transport Research Conference**
Monte Verità / Ascona, May 17 – 19, 2017

École Polytechnique Fédérale de Lausanne

Two-level hierarchical control for large-scale urban traffic networks

Anastasios Kouvelas, Nikolas Geroliminis
Urban Transport Systems Laboratory
École Polytechnique Fédérale de Lausanne
GC C2 390, Station 18, 1015 Lausanne, Switzerland
phone: +41-21-69-35397
fax: +41-21-69-35060
{tasos.kouvelas, nikolas.geroliminis}@epfl.ch

May 2017

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).

Keywords

Urban traffic control; hierarchical control; macroscopic fundamental diagram (MFD); perimeter control; distributed control; max-pressure.

1 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 (Kouvelas *et al.* (2017)), 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.

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, 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, 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.

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 local 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.

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. Finally, the last section concludes the paper with some closing remarks about this work.

2 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.

2.1 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

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

holds for every node n, where $k = 0, 1, 2, \ldots$ 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

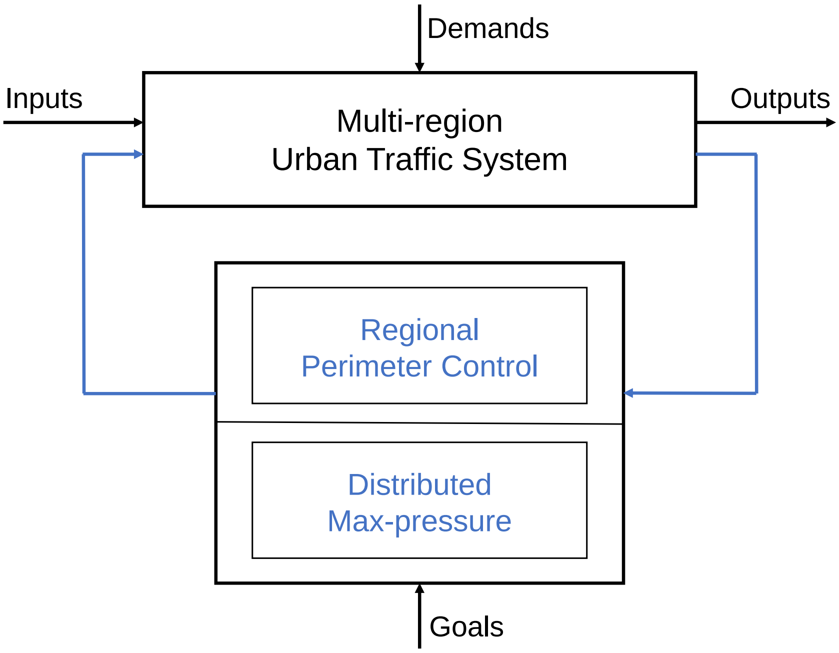
$$g_{n,j}(k) \geq g_{n,j,\min},\ \ j \in F_n \tag{2}$$

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 $n \in N$.

2.2 Framework description

Figure 1 describes the basic principle of the hierarchical control framework. Basically, we have two distinct controllers that act independently, without communicating of 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 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 sections.

Figure 1: Write here the caption of the figure.



2.3 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 Kouvelas *et al.* (2015, 2017) 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 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 Keyvan-Ekbatani *et al.* (2012), Geroliminis *et al.* (2013), Aboudolas and Geroliminis (2013), Ramezani *et al.* (2015).

2.3.1 Regulator for perimeter control

To this end, the regional multivariable feedback PI regulator derived in Kouvelas *et al.* (2017) reads in vector form

$$\mathbf{u}(k) = \mathbf{u}(k-1) - \mathbf{K}_{\mathrm{P}}\left[\mathbf{n}(k) - \mathbf{n}(k-1)\right] - \mathbf{K}_{\mathrm{I}}\left[\mathbf{n}(k) - \hat{\mathbf{n}}\right] \tag{3}$$

where $k = 0, 1, 2, \ldots$ 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, $\mathbf{K}_{\mathrm{P}}, \mathbf{K}_{\mathrm{I}} \in \mathbb{R}^{M \times N}$ 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 Kouvelas *et al.* (2017).

2.3.2 Derivation of the green time durations

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 $n_{i,\mathrm{start}}$ and $n_{i,\mathrm{stop}}$[1] 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 Leclercq *et al.* (2014), Ortigosa *et al.* (2014) and Ampountolas and Kouvelas (2015).

Equation (3) calculates the fraction of flows $\mathbf{u}(k)$ to be allowed to transfer between neighbouring regions. The obtained values are then used to derive the green time durations for the stages of the signalized intersections located at the inter-regional boundaries. As described in Kouvelas *et al.* (2017), for a given pair of sending and receiving regions, i, j, respectively, the ordered transferring flow by the controller is $u_{ij}(k)M_{ij}(n_i(k))$ vehicles per time unit (where $M_{ij}(n_i(k))$ denotes the MFD with the transferring flow from region i to region j). This flow is distributed to the corresponding intersections proportionally to the saturation flows of the controlled links (i.e., typically, links with more lanes are anticipated to accommodate more flow). To this end, every link z is required to transfer $u_{ij}(k)M_{ij}(n_i(k))S_z/S_{ij}$ flow, where S_z is the saturation flow of the link and S_{ij} the summation of all saturation flows related to the $i \rightarrow j$ movement. The green

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

stage duration g_z is given by $g_z(k) = u_{ij}(k)M_{ij}(n_i(k))C_{ij}/S_{ij}$, where C_{ij} defines the cycle time; without loss of generality C_{ij} is assumed to be equal for all intersections included in the $i \rightarrow j$ movement (i.e. the equation can be readily modified if this assumption does not hold).

Note that the real transferring flows may be different than the ordered ones for different reasons (e.g. low demand, spillback from downstream links); however, the regulator is robust to these occurrences due to its feedback structure (i.e. the differences will be integrated into the measurements of the following control cycles). Finally, in Kouvelas *et al.* (2017) the gain matrices $\mathbf{K}_\mathrm{P}, \mathbf{K}_\mathrm{I}$ and set-points $\hat{\mathbf{n}}$ of the controller are optimized in real-time by a learning/adaptive algorithm based on real performance measurements, ensuing a more realistic set-up of the problem. In the current work, we utilize the final regulator that is obtained after the solution of Riccati and the fine-tuning process.

2.4 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 (Varaiya (2013)). Several variations of the basic method that can be applied in real-time (depending on the available infrastructure) are presented.

2.4.1 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

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

where $x_{z,\max}$ 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

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

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

2.4.2 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

$$G_n = C_n - L_n - \sum_{j \in F_n} g_{n,j,\min}, \; n \in N \tag{6}$$

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 Kouvelas *et al.* (2014) is utilized here, 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

$$\tilde{g}_{n,j}(k) = \frac{P_j(k)}{\sum\limits_{i \in F_n} P_i(k)} G_n, \;\; j \in F_n \tag{7}$$

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

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

This procedure 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.

3 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 more critical. From an implementation point of view it is considered too expensive and time consuming to control all the intersection, and as a result a methodology is needed that can define the critical intersections for each region.

3.1 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 (Fig. 2), 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 3 presents the test network partitioned in 4 homogeneous regions. For the partitioning the algorithm presented in Saeedmanesh and Geroliminis (2016) has been used. The result 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, which are noted with circles in Fig. 4.

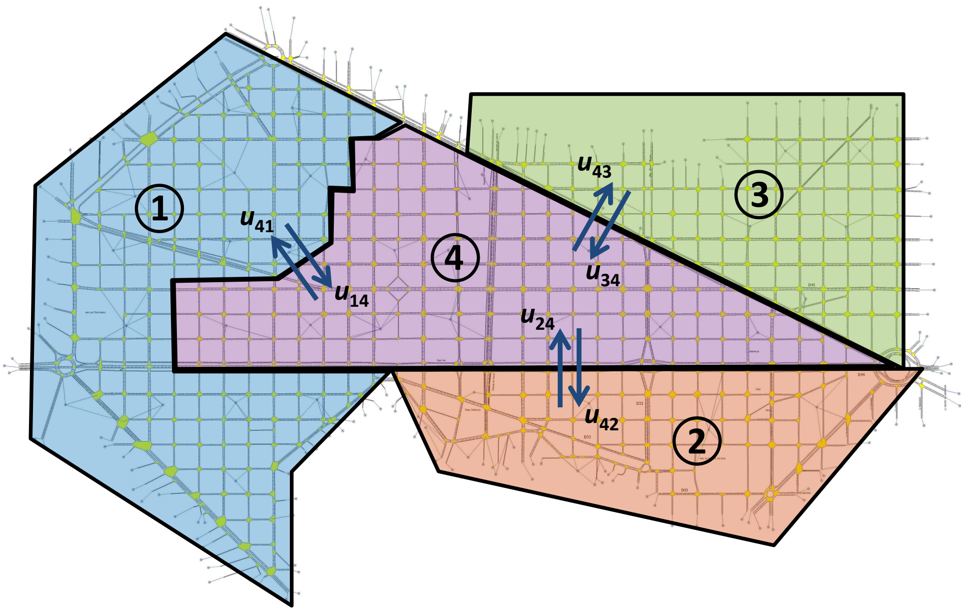
3.2 Selection of the critical intersections

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 important. These intersections are controlled then based on the max-pressure control scheme that was described earlier.

To this end we do an analysis of the simulation data for the fixed-time plan of the traffic lights. This scenario is called no control (NC) and it reflects all the pre-timed signal plans of the city of Barcelona. For this control scenario 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

Figure 2: Microsimulation model of the urban test network in Aimsun.

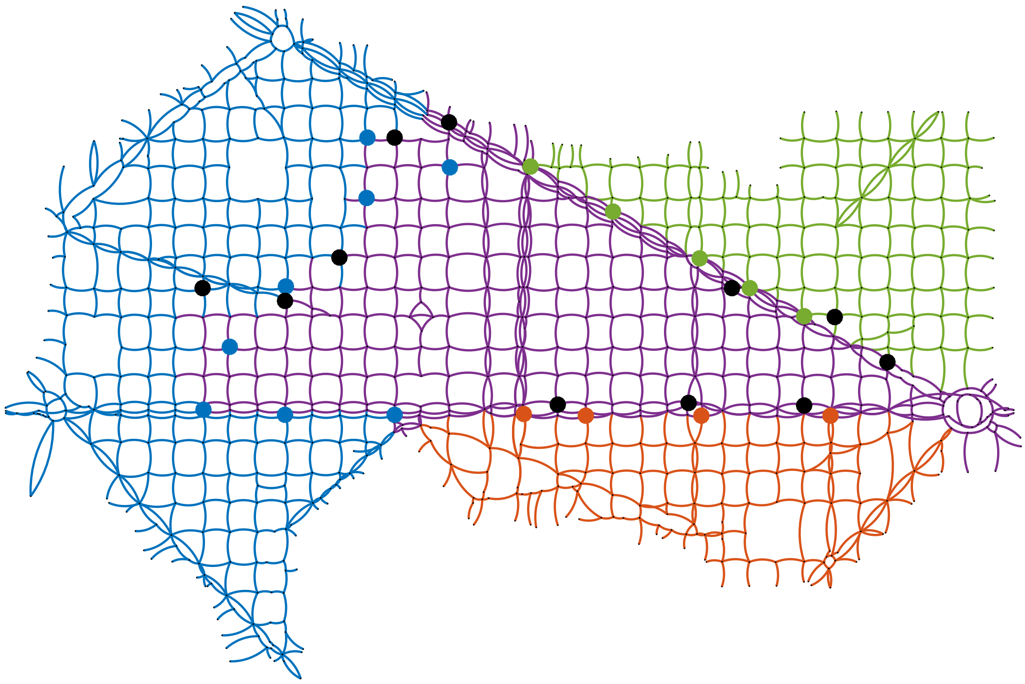
Figure 3: The test network partitioned in 4 regions to apply perimeter control.



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

Figure 4: Boundary intersections that the perimeter control is applied.

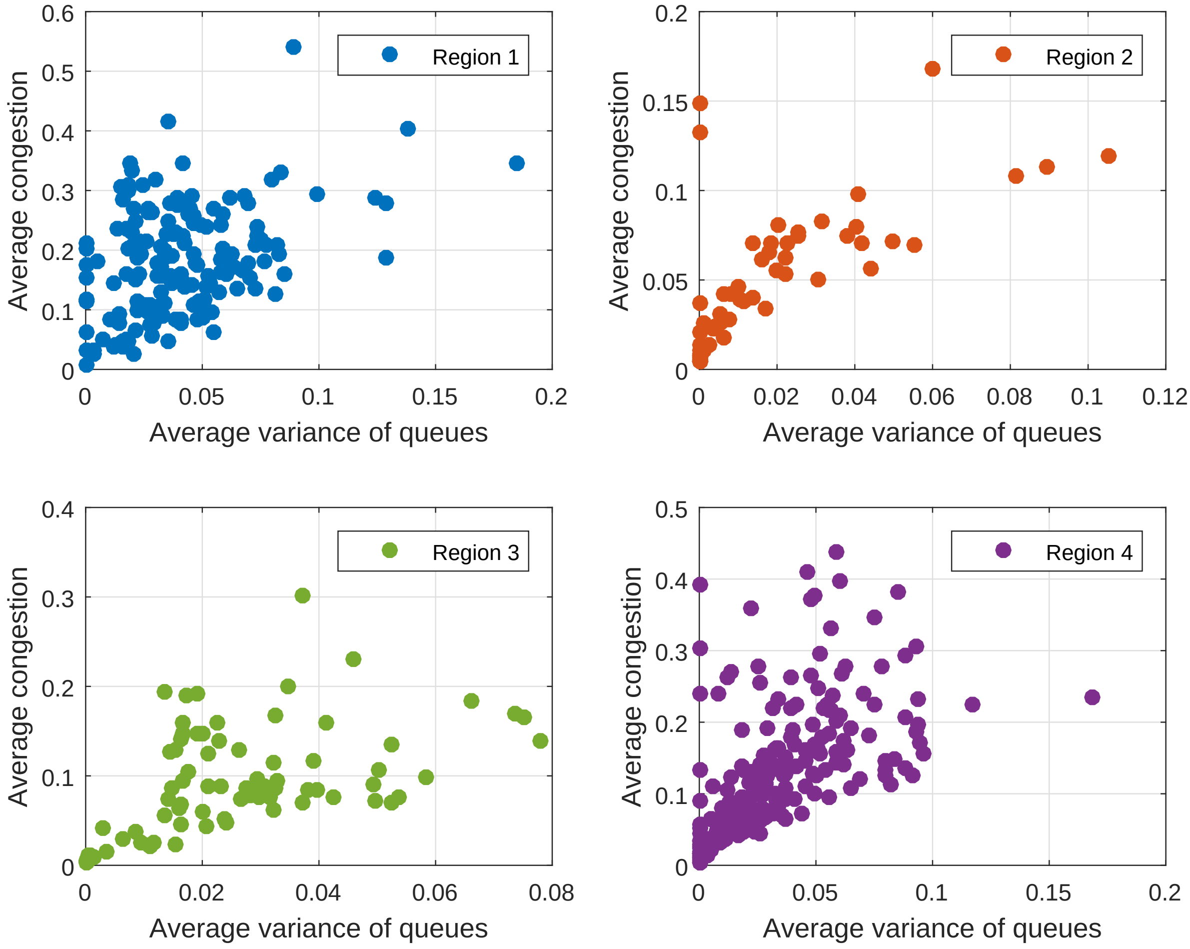


congestion as well as the homogeneity of each intersection (which is actually analogous to the variance of the queues). If we perform this analysis to our data and we average over time we can get a good flavour of the importance of each intersection and how critical it is to the propagation of congestion in the network. The more congested an intersection is the more critical for the overall congestion. Also, 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 5 presents the results of this analysis. 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 whole simulation horizon of 6.5 hours. Figure 6 presents the same results but focusing only on the 1.5 hours of peak traffic and excluding the beginning and end of simulation where the traffic loads are very low.

In these two plots we are looking for outliers, i.e. intersections with high variance of queues, high level of congestion, or, alternatively with high values for both metrics.

Figure 5: Write here the caption of the figure.



3.3 The fixed-time case and other simulated scenarios

In this section the simulation results for the no control case are presented. As mentioned earlier, for this control scenario, the actual fixed-time plans of the traffic control center of the city are applied in all the intersections. Table 1 presents all the statistics that we obtain from the simulation software for this scenario. These results need to be compared with all the other control scenarios which are the following: (a) only max-pressure controller, (b) only perimeter controller, and (c) the hierarchical scheme that combines both control approaches.

Figure 6: Write here the caption of the figure.

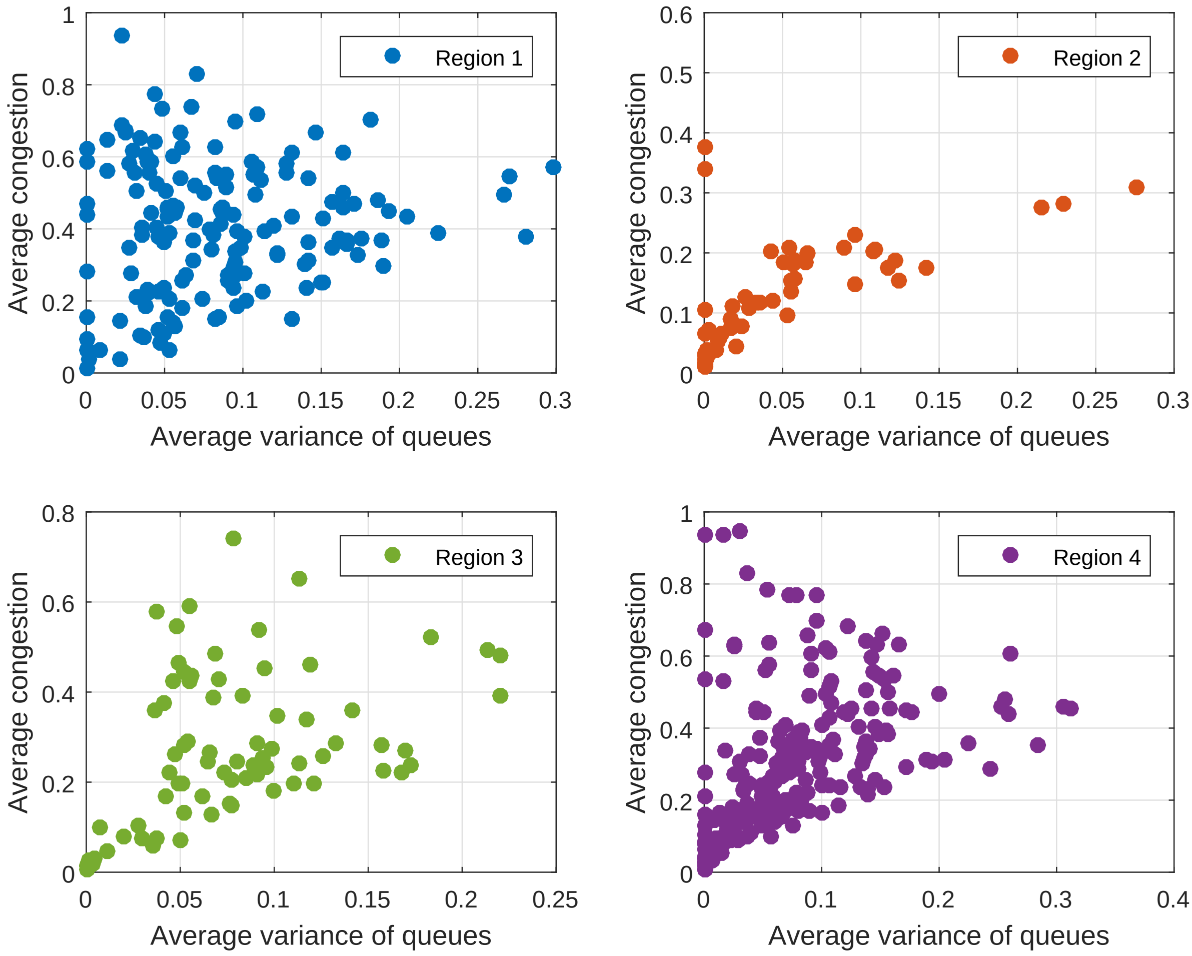


Table 1: Simulation statistics for the no control case.

Criteria	Value	Units
Average Delay	425.57	sec/km
Mean Speed	7.34	km/h
Mean Queue	6130.38	veh
Mean Virtual Queue	5662.32	veh
Total Travel Time	56742.25	h
Total Travelled Distance	404251.1	km
Vehicles Served	202428	veh

4 Summary and Important Notes

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. The performance of local distributed control is compared to perimeter control between zones, as well as to the combination of both distributed and centralized control.

5 References

Aboudolas, K. and N. Geroliminis (2013) Perimeter and boundary flow control in multi-reservoir heterogeneous networks, *Transportation Research Part B*, **55**, 265–281.

Ampountolas, K. and A. Kouvelas (2015) Real-time estimation of critical vehicle accumulation for maximum network throughput, paper presented at the *American Control Conference*, 2057–2062.

Geroliminis, N., J. Haddad and M. Ramezani (2013) Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach, *IEEE Transactions on Intelligent Transportation Systems*, **14** (1) 348–359.

Keyvan-Ekbatani, M., A. Kouvelas, I. Papamichail and M. Papageorgiou (2012) Exploiting the fundamental diagram of urban networks for feedback-based gating, *Transportation Research Part B*, **46** (10) 1393–1403.

Kouvelas, A., J. Lioris, S. A. Fayazi and P. Varaiya (2014) Maximum pressure controller for stabilizing queues in signalized arterial networks, *Transportation Research Record*, (2421) 133–141.

Kouvelas, A., M. Saeedmanesh and N. Geroliminis (2015) Feedback perimeter control for heterogeneous urban networks using adaptive optimization, paper presented at the *18th IEEE International Conference on Intelligent Transportation Systems*, 882–887.

Kouvelas, A., M. Saeedmanesh and N. Geroliminis (2017) Enhancing model-based feedback perimeter control with data-driven online adaptive optimization, *Transportation Research Part B: Methodological*, **96**, 26–45.

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

Ortigosa, J., M. Menendez and H. Tapia (2014) 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*, **3** (3-4) 245–266.

Ramezani, M., J. Haddad and N. Geroliminis (2015) Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control, *Transportation Research Part B*, **74**, 1–19.

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

Varaiya, P. (2013) Max pressure control of a network of signalized intersections, *Transportation Research Part C*, **36**, 177–195.