DIT4TraM

Distributed Intelligence & Technology
for Traffic & Mobility Management

Passenger-oriented distributed control strategy for large-scale urban systems

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 953783.

Distributed Intelligence & Technology for Traffic & Mobility Management

Project	DIT4TraM
Grant Agreement No.	953783
Start Date of Project	01-09-2021
Duration of the Project	36 months
Deliverable Number	D3.1
Deliverable Title	Passenger-oriented distributed control strategy for large-scale urban systems
Dissemination Level	Confidential
Deliverable Leader	Swiss Federal Institute of Technology Lausanne
Submission Date	February 28 2023
Author	Zhenyu Yang
Co-author(s)	Nikolas Geroliminis, Ludovic Leclercq, Eleni I. Vlahogianni, Marco Rinaldi, Erik-Sander Smits, Dimitrios Tsitsokas, Manon Seppecher, Mina Khalesian, Jiajie Yu, Konstantinos Katzilieris, Emmanouil Kampitakis, Paula Vogg

Release Approval

Name	Role	Date
Marco Rinaldi	Project reviewer	February 22nd 2023
Oded Cat	Project reviewer	February 27nd 2023
Nikolas Geroliminis	Task and work package leader	February 27th 2023

Document History

Version	Description	Name
1	Versions for internal review among authors	DIT4tramWP3_1_deliverable_Ver1
2	Versions for internal review among project members	DIT4tramWP3_1_deliverable_Ver2
3	Versions ready for releases	DIT4tramWP3_1_deliverable_final

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no. 953783

Contents

Executive Summary

This report represents Deliverable 3.1 of the DIT4TraM Project, comprised of research and development work carried out in Task 3.1: *Passenger-oriented distributed control strategy for large-scale urban systems*, as part of Work Package 3: *Cooperative multi-class distributed traffic management* during the period M1–M18.

Task 3.1 focuses on developing distributed traffic control algorithms to govern transportation networks, that are geographically distributed and with multiple entities and operators.

At the intersection level, the Integrated Signal and Bus Lane Control (ISBLC) framework is developed, aiming to maximize passenger throughput and improve the level of service for public transport passengers. It adjusts traffic signals and assigns dedicated bus lanes based on traffic conditions and bus occupancy and delay information. A smart intersection (agent) controls the traffic signal phase, as well as the activation of the upstream bus lanes in both directions of a main artery based on the observed passenger delays, giving priority status (weight) to bus passengers. The agent takes an action about all three decisions every timestep. The objective of the agent is to minimise the weighted total passenger waiting time.

At the corridor level, the proposed Deep Reinforcement Learning (DRL) model promises to effectively manage traffic signals in multi-modal networks, by capturing the complex interactions between private car traffic and bus transit and adapting to different road configurations. The model simultaneously minimizes the traffic delay and bus headway variations and can accommodate different road layouts, including dedicated bus lanes and mixed traffic lanes. The cooperation between intersections is achieved by sharing action data among neighbours. Scalability and portability are demonstrated by transferring trained models to other similar intersections, reducing the training costs in an extensive transportation system. A large benchmark over the most representative methods including the centralized DRL method is performed in numerical experiments.

At the network level, a two-layer control framework is developed to improve traffic signal control performance in congested networks. This framework combines the benefits of Max Pressure(MP) and Perimeter Control (PC) strategies and adjusts the green times of intersections based on real-time queue measurements. In the upper layer, perimeter control is applied in an aggregated scale between a set of homogeneously congested regions. At the end of every control cycle, the controller, based on inputs of aggregated regional vehicle accumulation, specifies the target inter-regional exchange flows for the next cycle, which are translated into the respective inter-regional green times between every pair of adjacent regions. In the lower layer, distributed control based on Max Pressure regulator is applied to a set of eligible intersections, in the interior of the regions. This set can contain all or a fraction of signalized intersections of the region, with the exception of those

used for PC (if PC is applied in parallel).

As a clear instance of decentralisation, the two-layer signal control framework entails combining centralized, aggregated perimeter control strategy, with partial and effcient distributed control. MP controllers do not communicate with each other or with any central control unit, but operate independently based on queue measurements directly upstream and downstream the controlled intersections, by adjusting green times of the approaches accordingly, at the end of every control cycle. The control layers do not exchange information, however their combined effect is indirectly considered by both controllers through the real-time traffic measurements that they receive as inputs.

In a multi-modal context, we propose two bus prioritization strategies, Intermittent Dynamic Bus Lanes (IDBL) and Adaptive Bus Lane Density Control (BLDC), which are developed using reinforcement learning to allocate road space between buses and vehicles in real-time. The algorithms are trained to minimize bus delays and ensure that the travel time of vehicular traffic is not significantly affected. Reinforcement Learning gives the ability to bus-agents to learn the correlation between prevailing traffic conditions and expected transit delays. The results' analysis showed that both methods manage to minimise transit delays even in higher demand scenarios while maintaining the disturbance of vehicular traffic close to mixed traffic conditions. We also implement Dynamic Bus Lanes (DyBL) in combination with perimeter control and evaluate different control algorithms and system architectures in terms of network performance, aiming at minimizing total passenger travel time. A simple dynamic controller is configured for controlling DyBL car inflow and implemented via microsimulation for a realistic network and demand scenario, in combination with a simple two-region perimeter control.

The above-mentioned traffic control approaches exemplify the possible ways of enhancing existing distributed control structures in many existing distributed system, such as the Integrated network management (INM) system of the Utrecht pilot. In a multi-model, multi-scale transportation system, the successful implementation of these novel control strategies relies on the data and information transmission between different scales and modes. For instance, in Chapter 3, the hierarchical traffic signal control relies on the availability of data from network traffic monitoring with different granularity. Meanwhile, the dynamical bus lane approaches of IDBL and BLDC in Chapter 5 highlight the benefit of information exchanging between different modes. Whereas in many existing systems, such as INM, such interactions are often ignored or underestimated.

Motivated by this observation, the report envisions that the currently functioning system, Integrated network management (INM), in the Utrecht pilot will be further enhanced with the concept developed in DIT4TraM. First, traffic information from different travel modes can be integrated, such as bicycle traffic, to facilitate better inform control decisions with aspects of multimodality. Second, in order to be able to take full advantage of additional data sources, chiefly Floating Car Data (FCD) can be intergated. With more data, the selection and assignment of intersection tasks would be based on the status of the traffic network with the development in Task 3.2 of DIT4TraM, i.e., the Jam Tree approach. When a Jam Tree is detected, by tracking its spatiotemporal evolution, the approach will allow identifying which intersections contribute to the target bottleneck, and guide the appropriate distribution of tasks leading to optimal bottleneck resolution.

1 Introduction

In recent decades, significant progress has been made in developing real-time traffic management strategies that improve performance in heavily congested networks with time-dependent traffic characteristics. Local adaptive strategies that are commonly used are based on heuristic optimization techniques. Due to the accuracy and computational burden challenge, such methods have become less effective in handling congestion propagation and queue spillbacks. The distributed control architecture, which originated from communication systems and production processes has gained increasing attention in traffic control recently. In the context of passenger-oriented mobility, it has become increasingly relevant to develop more practicable decentralized network-wide control strategies. Such control schemes act locally in coupled intersections and have been proven (under certain conditions) to stabilize the network conditions.

1.1 Scope of this deliverable

This deliverable is part of the DIT4TraM's work package 3 "Cooperative multi-class distributed traffic management" and corresponds to task 3.1 "Passenger-oriented distributed control strategy for large-scale urban systems ". The objectives of Task 3.1 are to design and implement a set of distributed algorithms that can govern a transportation network. We focus on developing local regulators that communicate with each other to reach an agreement on their control decisions, in order to ensure local cooperation towards the achievement of city-level mobility goals.

The approach is based on distributed optimization and takes into account human-oriented considerations, which makes it go beyond current practices. Information exchange between the local regulators will be carefully designed to ensure that interaction effects are integrated at the local level. The system will be designed with flexibility in mind, allowing for the self-organization of the individual components while ensuring sufficient compliance. At the same time, the operator will have the ability to manage the system at the overall level and optimize capacity utilization.

Part of the algorithms and mechanisms will be tested and demonstrated both in various simulation environments and in the Utrecht pilot project, with a focus on distributed traffic signal control for multi-modal transportation. EPFL will work closely with NTUA and UGE to design the algorithms and mechanisms for various system configurations, which will be integrated into the unified framework of Task 3.4.

Furthermore, this report emphasizes designing the proper distributed architectures and information exchange requirements for different DIT4TraM subsystems, which achieves better efficiency and resiliency. Motivated by the benefits of distributed approaches, we develop distributed control architecture for different applications and modes in transportation systems, including passenger-oriented distributed control, auctioning schemes for multimodal mobility services, cooperative schemes for local bottleneck control and finally traffic management platforms.

1.2 Structure of the deliverable

In this report, Chapter 2 presents a passenger-oriented intersection level framework of Integrated Signal and Bus Lane Control (ISBLC). Chapter 3 introduces a distributed traffic signal control model with deep reinforcement learning (DRL) for a multi-modal corridor-level network consisting of private car traffic and bus transit. Chapter 4 examines the effectiveness of the network-wide parallel application of perimeter control (PC) and (max pressure) MP strategies embedded in a two-layer control framework in a link-based macroscopic simulation environment. Chapter 5 develops two learnable, highly scalable and transferable bus prioritisation strategies using reinforcement learning under a connected environment in urban corridors. Chapter 6 describes the platform and the location for the pilot of Utrecht and envisions the potential extensions based on the existing system. Finally, Chapter 7 includes the conclusions.

2 Intersection-level control

2.1 Introduction

During the last years, the advent of connected vehicle technology and smart infrastructure [1] has created new opportunities for smarter and more efficient intersection-level control that leverages Vehicle-to-all (V2X) real-time communication capabilities for traffic information exchange, enabling more informed decision-making [2]. A significant amount of research is focused on creating adaptive schemes that aim to improve vehicle traffic at intersections (e.g., [3], [4]). However, most approaches do not distinguish between flows of different modes and are oriented towards the optimisation of vehicle traffic, not taking into account the challenging dynamics of a multimodal system. The future of transport, driven by the growing recognition of the benefits of a more sustainable, efficient, and equitable transportation system, is likely to be increasingly multimodal, especially as the population grows and cities become more congested. Therefore, in order to meet the diverse needs of all users in a fair way, passenger-oriented traffic and network management frameworks need to be explored.

In this chapter, we propose a novel passenger-oriented intersection level framework, namely Integrated Signal and Bus Lane Control (ISBLC), which aims to maximise the passenger throughput of intersections, while ensuring the stability and smooth operation of public transport. One would expect these two objectives to always be aligned. It is not unusual, however, especially at local intersection level, that the number of vehicle passengers significantly exceeds the number of bus passengers, leading to a misalignment of interests between the two. The ISBLC dynamically adjusts traffic signals and simultaneously assigns dedicated bus lanes to upstream lanes of the intersection, based on the observed traffic conditions, as well as the information it receives from V2I communication regarding vehicle/bus passenger occupancy and bus delays. With that information in hand and combined with a state-of-the-art reinforcement learning algorithm, ISBLC objective is to reduce total passenger waiting time, while providing the highest level of service for public-transport passengers. Results of ISBLC are compared with traditional traffic signal and bus prioritisation approaches to provide insights into the effectiveness of the proposed approach and inform future research towards passenger-oriented and multi-modal traffic control schemes.

2.2 Integrated Signal and Bus Lane Control

2.2.1 Concept

In the ISBLC scheme, a smart intersection-agent, operating in a connected environment controls the traffic signal as well as the activation-deactivation of the upstream bus lanes in both directions based on the prevailing traffic conditions and information on passenger occupancy per vehicle and bus delays defined as the deviation from a given static bus timetable. An illustrative example of a configuration of an ISBLC intersection is depicted in Figure 1.

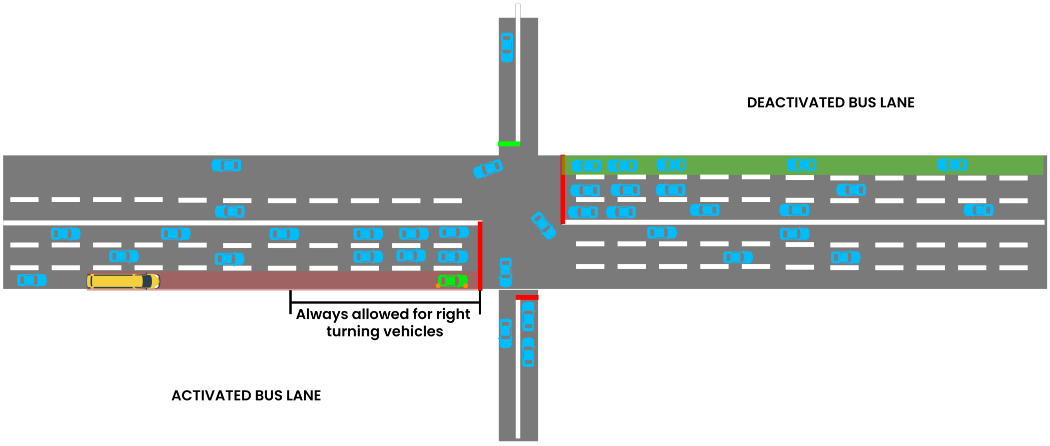


Figure 1: Configuration of an ISBLC intersection. The red and green stripes represent an activated and deactivated upstream bus lane, respectively. Right turn vehicles are always allowed within the last 80 metres.

The way ISBLC operates is described as follows. The intersection starts receiving information from incoming buses at a predefined time t before their projected arrival (100 seconds) to the relevant upstream edge of the intersection. This provides the traffic signal agent with crucial additional time to evacuate the bus lane if needed. The incoming buses report the number of passengers and an already accumulated delay from previous locations of a hypothetical traffic network "outside" the intersection. When the bus enters the intersection, it reports to the ISBLC traffic signal its passenger volume and delay based on a baseline bus timetable for a bus stop that is placed on the downstream approach of the intersection (including the delay outside the intersection). According to the above described observation combined with the prevailing passenger-oriented traffic conditions, the traffic signal decides which will be the next green phase and if the upstream bus lanes will be activated or not, separately.

If a bus lane gets activated, all relevant vehicles are informed. Vehicles in the adjacent lanes are not allowed to enter the bus lane anymore. Vehicles within the bus lane check their adjacent lane for following and leading neighbouring vehicles. If they observe a gap of sufficient length between the leading and trailing vehicles (Figure 2), they get instructed to change lane. If the gap is not sufficient, they continue straight until lane changing is possible again. If a vehicle got an instruction to change lane but could not follow through, the vehicle continues straight until lane changing is possible again. Moreover, if the lane area upstream of the bus is not activated as a bus lane by the traffic signal for an upcoming bus, all vehicles are allowed to enter behind the bus. When the bus lane is deactivated, vehicles in the adjacent lanes and vehicles on the (possibly) previously activated bus lane are informed that they are allowed again to enter or stay on the bus lane, respectively, and no other restrictions take place. It should be noted, that right turn vehicles are allowed within the last 80 meters of an activated upstream bus lane and thus they are excluded from any exit-instruction. This approach is designed in such a way that it can easily be extended to large corridors with consecutive ISBLC intersections, where every intersection controls its traffic signal and the activation or deactivation of its upstream bus lanes.

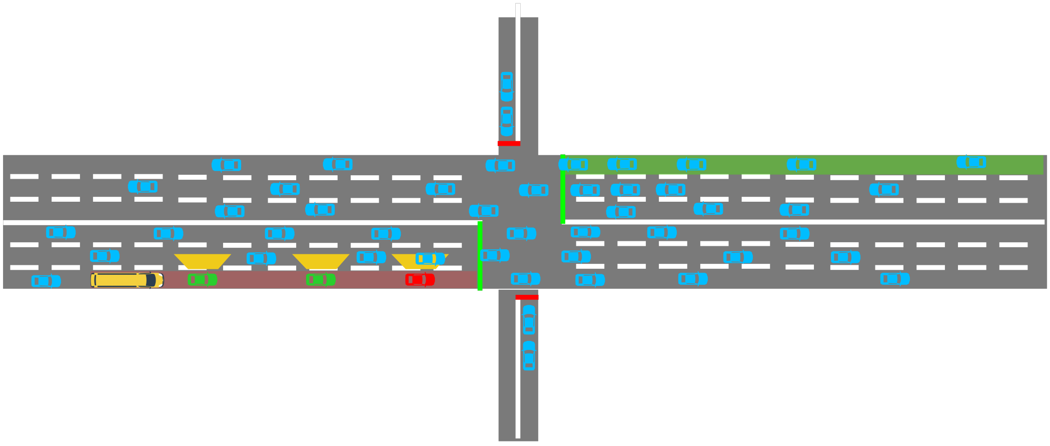


Figure 2: Evacuation of an upstream bus lane in an ISBLC intersection.

Drivers are assumed to be fully compliant to the given instructions. This means that drivers outside the activated bus lane will never enter and drivers inside will always seek to change lanes when requested, but conduct a lane change only when conditions are suitable for a safe lane change, i.e. they operate as they would normally do. Finally, it is assumed that drivers are truthful and don't exhibit malicious behaviour. This simply means that they will never misreport the number of passengers inside the vehicle.

2.2.2 Deep Reinforcement Learning ISBLC Controller

In this work, the Proximal Policy Optimization (PPO) [5] algorithm is utilised as a reinforcement learning approach to address the problem of traffic signal control at intersections. Proximal Policy Optimization is a state-of-the-art on-policy and model-free reinforcement learning algorithm that belongs in the policy-gradient methods. It aims to find a balance between exploration and exploitation in order to improve the training stability by updating the policy in a local, proximal manner, avoiding "too severe" policy updates.

The structural components of the agent, namely the state, action and reward, are defined as follows. The agent observes a state s in each step (step duration = $5sec$). The state s can be divided in two parts. A traffic signal related and a bus lane related one. Concerning the traffic signal related state, per green phase (4 green phases in total) the following are given: the number of total passengers $N_p^{[phase]}$, the total number of waiting passengers $q_p^{[phase]}$ and the total waiting time of passengers $W_p^{[phase]}$. All described values are measured within a distance from the traffic light (200 metres) and the bus passengers are also included. Moreover, it is given if the phase was activated or not (1 or 0) during the previous step $act_{prev}^{[phase]}$, if the phase during the current step is activated or not (1 or 0) $act_{cur}^{[phase]}$ and finally the number of consecutive activations of the phase $n_{act}^{[phase]}$. Concerning the bus lane related state, per bus lane (2 in total) the following are given: the previous bus lane activation state (0: activated, 1: deactivated) bl_{prev}, the current bus lane activation state (0: activated, 1: deactivated) bl_{cur}, if the related to the bus lane phase was activated or not (0 or 1) during the previous step $phase_{prev}^{bl}$, if the related to the bus lane phase was activated or not (0 or 1) during the current step $phase_{cur}^{bl}$, a noisy value of the remaining time until the entrance of an upcoming bus in the relevant bus lane of the intersection rte, the distance of the bus from the traffic signal d_{tl}, the expected arrival delay to the downstream bus stop $delay$, the number of vehicles in front of the bus veh_f and finally the number of passenger on the bus lane n_{bp}. The state space S is continuous and more precisely $S \in R^{42}$.

The traffic signal agent observes a state every 5 seconds and based on that takes an action a. As was mentioned the agent has to take simultaneously 3 actions. One related to the traffic signal (which will be the next green phase) and another two for the bus lanes (0: activation and 1: deactivation). To avoid a large discrete action space by combining the aforementioned actions, the action was designed to be a multi discrete one. This means that from an RL perspective 3 actions are taken simultaneously. The action can be defined as:

$$a = \begin{cases} a_{tl} & \in \{0,1,2,3\} \\ a_{bl_1}, a_{bl_1} & \in \{0,1\} \\ a_{bl_2}, a_{bl_2} & \in \{0,1\} \end{cases} \tag{1}$$

Thus, the action space A is defined as $A = \{0,1,2,3\} \times \{0,1\} \times \{0,1\}$.

The reward is the most crucial component that drives the agent towards the desired objective. The objective of the traffic signal agent is to minimise the passenger average waiting time while also minimising the reported bus delays based on a given bus schedule. To that end, the reward consists of seven components as follows:

$$reward = r_p + r_b + r_{cbl} + r_{gt}^{min} + r_{gt}^{max} + r_o + r_c \tag{2}$$

The first component is the negative average waiting time of all passengers which accounts for the minimization of the passenger waiting time,

$$r_p = -\frac{\sum_{phase=1}^{n} W_p^{[phase]}}{\sum_{phase=1}^{n} q_p^{[phase]}}\ ,\ n = 4 \tag{3}$$

The second component is the negative expected arrival delay per bus lane, which is taken into consideration only when a bus has entered the intersection's edges, weighted by the number of bus passengers and is the component that accounts for bus prioritisation,

$$r_b = -\left(\frac{2}{1+\frac{1}{n_{bp_i}}}\right)\sum_{j=1}^{k}|delay_j|\ ,\ \forall\ bus\ j\ on\ bus\ lane\ i,\ i \in \{1,2\} \tag{4}$$

To avoid severe bus lane state changing (for both bus lanes), a small negative reward (third component) was given every time that the agent's action resulted in bus lane activation state change,

$$r_{cbl} = \begin{cases} -5,\ if\ bl_{prev} \neq bl_{cur} \\ 0,\ otherwise \end{cases}\ ,\ \forall\ bus\ lane\ i,\ i \in \{1,2\} \tag{5}$$

The fourth component is to ensure a minimum green time for each phase by giving a negative reward when the number of consecutive phase activations is less than 2.

$$r_{gt}^{min} = \begin{cases} -50,\ if\ n_{act}^{[cp]} < 2 \\ 0,\ otherwise \end{cases}\ ,\ cp = current\ active\ phase \tag{6}$$

The fifth component is to also ensure a maximum green per phase by giving a negative reward that increases with the number of consecutive phase activations. This gives more flexibility to the traffic signal to exceed the maximum green in order to prioritise a bus if needed.

$$r_{gt}^{max} = \left(n_{act}^{[cp]} - 10\right)^{1.3} \times 30,\ cp = current\ active\ phase \tag{7}$$

The sixth component is the left turn overflow penalty. It is important, in order to maximise the efficiency of a prominent bus lane activation, to avoid left turn overflows that will create gridlock phenomena. Thus, a negative reward is given for left turns when overflow is observed.

$$r_o = \begin{cases} -10, if\ overflow = True \\ 0,\ otherwise \end{cases}, \forall\, protected\, left\, turn\, i,\ i \in \{1,2\} \tag{8}$$

Finally, the seventh and last component concerns the constraint of the option of bus lane activation when there is no bus on the upstream edges of the intersection or at least approaching. For that reason, a negative reward is given when any of the bus lanes is activated in a situation like the aforementioned one.

$$r_c = \begin{cases} -10, if\ no\ bus\ is\ detected \\ 0,\ otherwise \end{cases}, \forall\, bus\, lane\, i,\ i \in \{1,2\} \tag{9}$$

2.3 Implementation

2.3.1 Description of the testbed intersection

SUMO (Simulation of Urban MObility) [6] micro-simulation framework is used to create an artificial 4-leg intersection (Figure 3). The approaches of the main road have a length of 400 metres and consist of 3 lanes plus one extra reserved lane for left turn with an 80 metres length. The secondary roads' approaches have a length of 250 metres and consist of 2 lanes. One bus stop is placed on each of the main road's downstream edges. The traffic signal has a 4-phase program with protected left turns.

2.3.2 Baseline Traffic Demand Scenario and Bus Timetable Derivation

Initially, a two-hour baseline traffic demand scenario, which resembles to normal traffic conditions and corresponds to approximately 400 veh/hour per lane for the main artery and 150 veh/hour per lane for the secondary roads is initialised. The derivation of the bus timetable is based on the baseline scenario. In more detail, a two-hour simulation using the baseline scenario's demand with a warm-up period of 15 minutes was run for 100 times with random seeds. The bus timetable resulted as the mean measured arrival time for each bus stop (2 in total).

2.3.3 Training Setting

The PPO traffic signal agent was trained by interacting repeatedly with the two hour simulation. For the creation of vehicles' passengers a discrete distribution was made with 1.6 mean passengers per vehicle. Regarding the generation of bus passengers we hypothesise that traffic demand follows roughly the same pattern

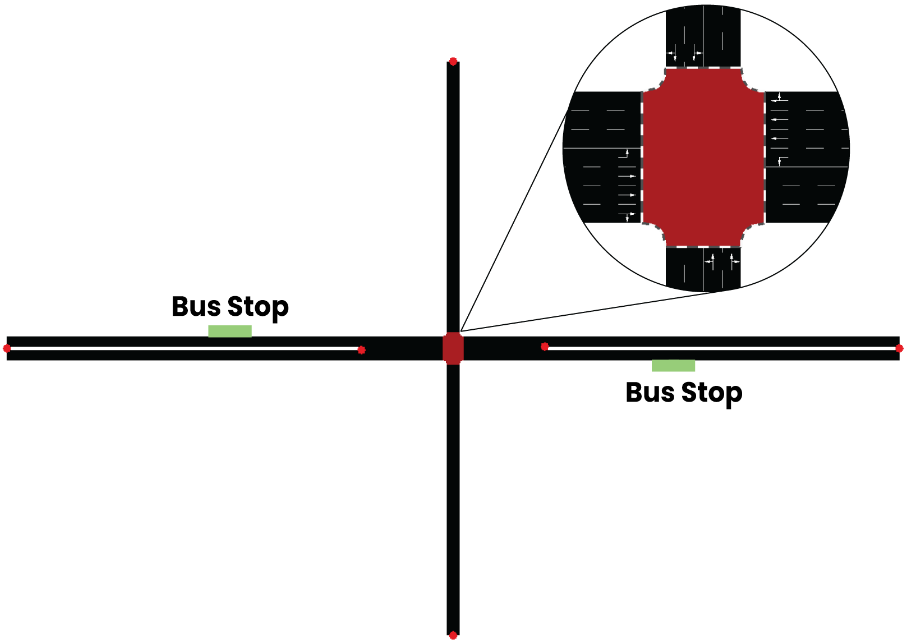


Figure 3: Layout of the ISBLC intersection.

with bus passenger demand, i.e. when traffic demand increases, the bus passenger demand increases also. Thus, bus passengers are generated from a gaussian distribution with mean equal to 50 passengers and a standard deviation of 20 passengers. The generated value is clipped by the minimum and maximum number of passengers. The maximum number of passengers is bounded by the capacity of the buses which was set to be equal to 100 (12 metres bus). The minimum number of passengers was set to 30. We call the above described distribution, baseline distribution and corresponds to the baseline scenario.

In order to increase generalisation, we follow a structured way of designing episodes explained below. First, a 12 buses per hour demand (per direction) was created and the bus timetables were derived as discussed. In order to avoid the overfitting of the agent to the bus schedule in each episode each created bus was chosen using a Bernoulli distribution. Thus, each bus was entering the simulation with probability 0.5. Second, the chosen buses to enter the simulation were entered with a delay that follows a truncated gaussian distribution from 0 to 120 seconds with mean delay equal to 60 seconds and a standard deviation of 20 seconds. Third, the demand of each episode is generated as a random scale of the baseline scenario demand ranging from 1.0-1.8. The same logic is followed for the generation of bus passengers where the baseline distribution is shifted proportionally to the demand scale. Of course, the maximum number of passengers is always constrained by the given bus capacity. Finally, the demand scale was changing randomly (1.0-1.8) during the simulation (every 15 minutes).

The benefits of parallel environments in reinforcement learning are well known [7], [8]. Parallelisation can greatly boost the training process as more samples are collected in unit time. Moreover the independent collection of data can lead to more diverse and comprehensive training data, leading to better generalisation, improved performance, faster convergence, and an important reduction in training time. For that reason, a synchronous environment parallelisation is performed, which resulted in less than 90 minutes training time until convergence.

2.3.4 Evaluation Process

To evaluate the developed method, 9 two-hour simulation scenarios, with a bus demand of 12 buses per hour per direction on the main artery and an initial traffic demand scale varying from 1.0 to 1.8 (step 0.1) are executed 10 times each (with random seeds) and the extracted results are averaged. Moreover, the buses are initialised with a random delay with a mean delay equal to 60 seconds (with a standard deviation of 10 seconds) to test if the proposed method can reduce the accumulated delays “outside” the intersection. Finally, both vehicle and bus passengers are generated following the same principles that were discussed in the training setting subsection.

The proposed method is compared with four alternative well established methods to evaluate its efficiency; static traffic signal control optimised by Webster’s formula [9], Webster traffic signal control combined with dedicated bus lanes, actuated traffic signal control [10] and finally, actuated traffic signal control combined with dedicated bus lanes. Webster’s method is a commonly used approach for traffic signal control. It involves using a fixed schedule to control the traffic signals, which is optimised based on Webster’s formula to determine the optimum cycle length and green times per phase of a traffic signal. Actuated signal control relies on sensors, such as inductive loops or cameras, to detect the presence of vehicles at the intersection. Based on the detected presence, the traffic signal timing is adjusted accordingly to optimise the traffic flow, reduce delays, and improve the overall performance of the intersection.

2.4 Findings

To evaluate the compliance with the derived bus timetable, the Mean Arrival Time Deviation is depicted in Figure 4. ISBLC is by far the best method keeping the bus delay below 75 seconds for all demand scales. With an initial mean delay of 60 seconds it can be extracted that for demand scale below 1.6 alleviates completely the delay occurred to the bus by the intersection and for demand scale below 1.5 manages to mitigate the initial delay “outside” the intersection reaching approximately a 20 seconds reduction for demand scale 1.0.

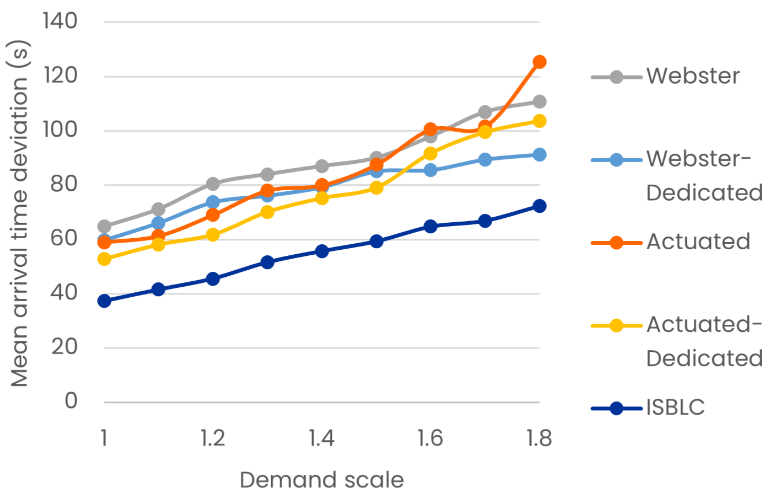


Figure 4: Mean arrival time deviation from timetable.

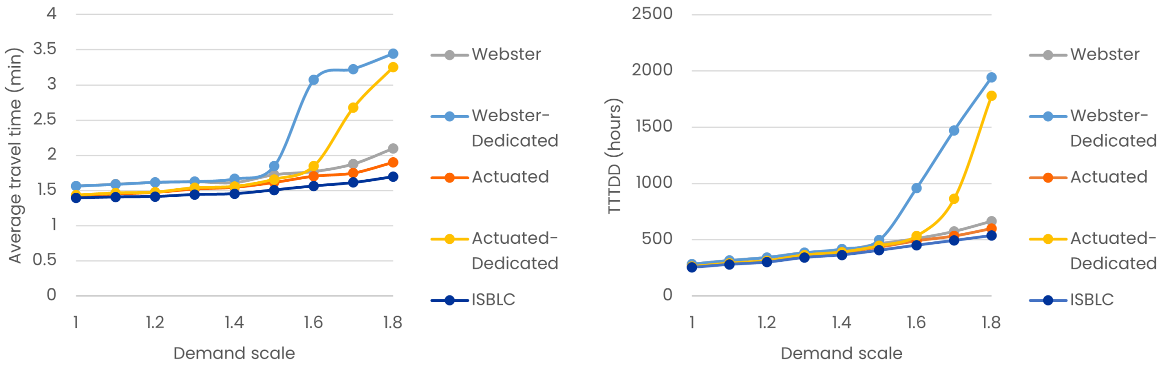
The Total Travel Time and Depart Delay (TTTDD) [11] metric is considered to be one of the most suitable metrics for a fair comparison of methods which are tested in a fixed simulation time regarding vehicular traffic and is defined as:

$$TTTDD = TotalTravelTime + TotalDepartDelay \tag{10}$$

Where $TotalTravelTime$ denotes the total travel time of passengers inserted in the simulation with the passengers generated during the warm up period excluded and the $TotalDepartDelay$ denotes the total departure delay of the inserted passengers as well as of the passengers waiting to be inserted into the simulation. Again, the passengers generated within the warm up period are excluded.

Thus, to evaluate the impact of ISBLC on vehicular traffic the average travel time and the TTTDD metric of vehicle passengers are depicted in Figure 5. For both metrics, ISBLC is shown to be the best method for all the demand scales concerning the vehicular disturbance.

To investigate the favouring of ISBLC for bus passengers over vehicle passengers in the context of transit prioritisation, the average waiting time of vehicle passengers and bus passengers is depicted in Figure 6. As one can observe, the average waiting time of bus passengers is negligible and the difference in average waiting time is retained above 12 seconds for all demand scales. Comparing bus passenger average travel time and TTTDD between all methods (Figure 7), ISBLC outperforms the other methods. The succeeded transit prioritisation of ISBLC can be also extracted by comparing the results of Figure 5 and 7 where the difference in improvement for bus passenger from vehicle passenger results, is noticeably greater. Moreover, from these two figures is evident that ISBLC is the best method regarding the overall passenger traffic.



(a) (b)

Figure 5: Average travel time (a) and TTTDD (b) of vehicle passengers.

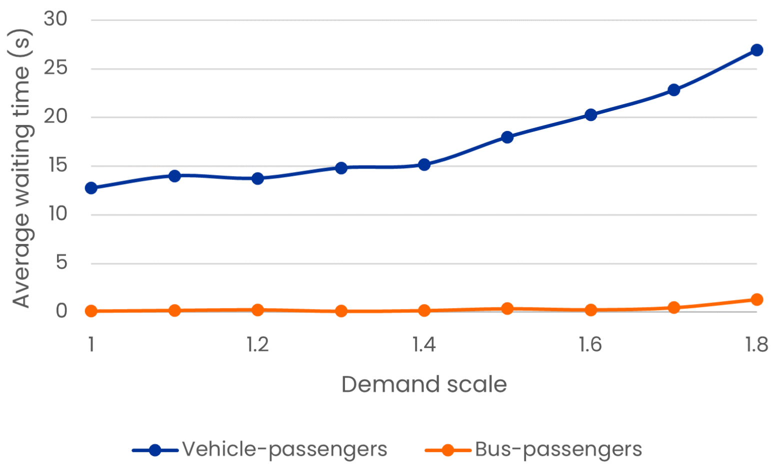
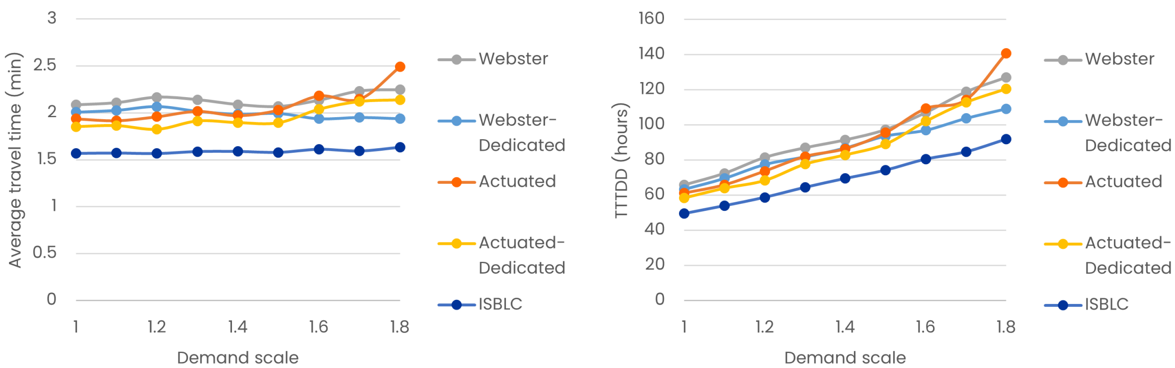


Figure 6: Average waiting time of bus passengers and vehicle passengers for ISBLC.



(a) (b)

Figure 7: Average travel time (a) and TTTDD (b) of bus passengers.

Finally, an analysis is conducted for the bus lane activation by ISBLC method which is depicted in Figure 8. It can be seen (Figure 8 (a)) that there is a rather descending pattern regarding the number of bus lane activations as the demand scale ascends. This can be explained by the fact that the impact of bus lanes on vehicular traffic is more intense as demand increases. Moreover, the ratio of vehicle passengers to bus passengers is increasing and the objective of the ISBLC except for the compliance with timetables is also to improve overall passenger traffic. On the other hand, in Figure 8 (b) the percentage of consecutive bus lane activations for larger demand scales seems to also increase. Although, this is difficult to analyse in detail because the results came from a complex training process, it seems to be beneficial to reduce the total number of activations but lengthen the time of activation as this confines the disturbance on vehicular traffic (frequent short-timed activations with high traffic demand create intense lane changing behaviour). Another reason for the prolonged bus lane activations for higher demand scales, is that the evacuation of a bus lane, with the increase of traffic demand, requires more time due to the limited space in the adjacent lanes which subsequently results in fewer successful lane changes.

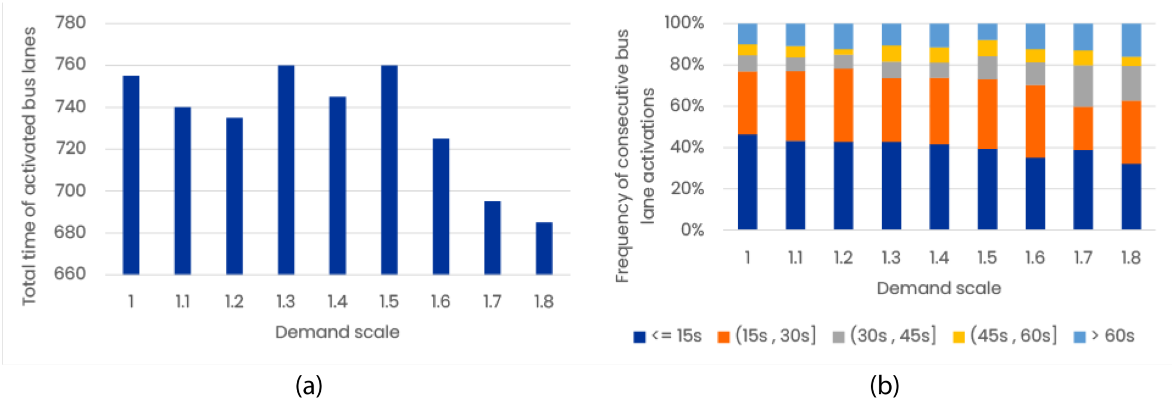


Figure 8: Bus lane activation analysis. (a) Number of bus lane activations per demand scale; (b) Frequency of consecutive bus lane activations per time interval.

2.5 Concluding remarks

This work proposed a novel approach for intelligent intersection-level control (ISBLC) by combining traffic signal and upstream bus lane control with the objective to maximise passenger throughput while prioritising transit by minimising reported transit delays based on a given schedule. Reinforcement learning is used because of its ability to learn in complex dynamics, such as traffic dynamics, that are difficult to be modelled by analytic methods. It allows the system to learn and adapt to continuously changing traffic conditions and demands in real-time while enabling the proposed scheme to continuously improve its decisions based on the consequences of previous actions, leading to more efficient and effective traffic management.

The results indicated the excellent performance of ISBLC as it manages to balance between optimising passenger traffic and transit prioritisation. However, there are some strong assumptions and limitations. First, a fully commented environment is assumed for the provision of necessary information as input to the ISBLC method. Thus, to implement such a scheme in the field, a number of prerequisites must be met, including a great penetration of connected vehicles, smart infrastructure, and robust communication networks. Moreover, a 100% drivers' compliance rate is assumed. This could lead to optimistic results as in reality, drivers may deviate from the intended behaviour due to various reasons such as personal preferences, etc. These drawbacks of course could be surmounted under Connected and Automated Vehicles (CAVs) which eliminate the human factor. Another limitation is that ISBLC was tested in a single intersection.

In future research, the proposed method will be tested on various connected vehicles penetration and drivers' compliance rates. Moreover, this work will be extended to an urban corridor level and various bus demands will be tested with fixed and dynamic bus timetables. Another interesting direction is to extend ISBLC to intersections where a bus lane exists in all directions.

3 Corridor-level control

In this chapter, we present a distributed traffic signal control model with Deep Reinforcement Learning (DRL) for a multi-modal corridor-level network consisting of private car traffic and bus transit, which is developed by UGE. The model simultaneously minimizes the traffic delay and bus headway variations and can accommodate different road layouts, including dedicated bus lanes and mixed traffic lanes. Cooperation between intersections is achieved by sharing action data among neighbours. Scalability and portability are demonstrated by transferring trained models to other similar intersections, reducing the training costs in an extensive transportation system. A large benchmark over the most representative methods, including the centralized DRL method, is performed in numerical experiments.

In the following subsections, the formulations of the agent's design, numerical test settings, sensitivity analysis of the tradeoff between two objectives, and test results are described.

3.1 Agent design

In the proposed distributed control algorithm, the agent is a traffic signal controller associated with an intersection in a corridor network equipped with bus lines. The agent's action involves allocating the green to one of the predefined phases for the following decision step. If the action chosen differs from the last action, the yellow and integral red phase of five seconds is activated. Agent's state and reward need to be modeled with both real-time car traffic state and bus operation information.

3.1.1 Agent's state

The agent's state space $S_{i,t} = [S_{i,t}^{traffic}, S_{i,t}^{transit}, S_{i,t}^{coop}]$ consists of the state observation for three components: car traffic, bus transit, and cooperativeness.

The car traffic state $S_{i,t}^{traffic} = [D_{i,t}, O_{i,t}]$ represents the total waiting time of vehicles, $D_{i,t}$, and the occupancy of each incoming leg, $O_{i,t}$, of the intersection i at decision step t. As depicted by Equation (11), $D_{i,t}$ results from the total number of stopped vehicles for all incoming legs within decision step t.

$$D_{i,t} = \sum_{\forall t' \in t} \sum_{\forall m \in M_i} n_{i,t'}^{m} \tag{11}$$

where $n_{i,t'}^{m}$ is the number of halting vehicle on the mth incoming leg of intersection i at time t'.

The bus transit state $S_{i,t}^{transit} = [h_{i,t}^f, h_{i,t}^b]$ is the forward and backward space headway of the bus on the incoming leg of intersection i at decision step t. Note that the optimal bus service is usually obtained when time headways are homogeneous. This also applies to space headways. The bus control strategy tries to equalize forward and backward space headways for all buses.

If the bus is too far from the signal, the signal's latest action does not contribute to bus service, so the agent should take action regardless of the bus state. When a bus distance to the traffic signal is below the control distance (for bus transit), $d_{i,t}$, the agent has to find the tradeoff between car traffic-related and bus transit-related objectives by activating the reward for bus transit. The control distance, $d_{i,t}$, is a parameter defining the maximal distance at which the agent i needs to consider the impacts of incoming buses while computing the upcoming decision/action.

The calculations of control distance in dedicated bus lanes and mixed traffic lanes are different. The control distance for bus lanes is defined by the maximum value between the distance for a bus decelerating to stop with an acceptable deceleration, according to the movement equation, and the distance for a bus moving forward at maximum speed during a decision step, defined by Equation (12):

$$d'_{i,t} = \max\{\frac{v_{max}^2}{2a_{dec}}, v_{max}\Delta t\} \tag{12}$$

where v_{max} is the speed limit for buses, a_{dec} is the acceptable deceleration for buses approaching to stops and Δt is the time span of a decision step.

For the mixed lane, the control distance is defined according to the speed of the bus, as Equation (13) displays. v_{cri} is the predefined critical speed for a bus. If the bus is at a low speed and queuing in a line, the action of the signal can influence the queue and then the bus, no matter how far the bus is from the intersection. Thus, the control distance is the distance between the bus and the intersection, $x_{i,t}^b$. If the bus is running at a normal or high speed, the control distance is calculated based on the formula in Equation (12). The control distance in the mixed traffic lane is given by:

$$d''_{i,t} = \begin{cases} x_{i,t}^b, & v_{i,t} < v_{cri} \\ \max\{\frac{v_{max}^2}{2a_{dec}}, v_{max}\Delta t\}, & v_{i,t} \geq v_{cri} \end{cases} \tag{13}$$

If the bus is outside the control distance or there is no bus on the incoming leg, the forward and backward space headways are set to 0. When several buses are on the same leg, only the state of the bus closest to the signal is collected. To exclude the situation where a dwelling bus is regarded as queuing, the critical position at which the bus speeds up to v_{cri} departing from a stop is defined. If the bus distance is greater than the critical position distance to the traffic signal, the bus state is set to 0. Thus, if $x^b_{i,t} > x^s_i - \frac{v^2_{cri}}{2aacc}$, $S^{transit}_{i,t} = [0, 0]$.

$S^{transit}_{i,t}$ can be concluded as Equation (14):

$$S^{transit}_{i,t} = \begin{cases} [h^f_{i,t}, h^b_{i,t}], & x^b_{i,t} < \min\{d_{i,t}, x^s_i - \frac{v^2_{cri}}{2aacc}\} \\ [0, 0], & otherwise \end{cases} \quad (14)$$

The cooperativeness state $S^{coop}_{i,t} = [a_{i-1,t-1}, a_{i,t-1}, a_{i+1,t-1}, n^y_{i,t}]$ consists of 3 components:

- the last action of agent i: $a_{i,t-1}$;
- the set of actions taken by the immediate neighborhood of agent i: here, the two neighbors $(a_{i-1,t-1}, a_{i+1,t-1})$; and
- the number of switches from one phase to another during the last ten action-steps for agent i: $n^y_{i,t}$, this variable monitors how frequently the controller changes phases. If the switches are too frequent, a large amount of the available capacity is lost due to yellow and integral red phases. It is important for controllers to consider this information in their decision-making process as it is not reflected in the other state variables.

3.1.2 Agent's reward

The same variables as the agent's state are used to define the reward function to enhance traffic efficiency at the intersection and to homogenize (keep constant) bus headways. Thus, the reward provides feedback regarding three dimensions, $r_{i,t} = r^{traffic}_{i,t} + r^{agent}_{i,t} + cr^{transit}_{i,t}$, matching with a performance evaluation according to car traffic, $r^{traffic}_{i,t} = r'^{traffic}_{i,t} + r''^{traffic}_{i,t}$, the bus system, $r^{transit}_{i,t}$, and the capacity loss due to the agent's past actions, $r^{agent}_{i,t}$. c is a predefined positive integer representing the weight of bus transit in the reward function.

For car traffic, the agent aims to minimize the total waiting time and to limit the density of stopped vehicles on each incoming leg. Therefore, the total waiting time and the occupancy at the current decision step and the last step are compared to decide the reward, as Equations (15) and (16) show, where O_{cri} is the predefined critical occupancy. At intersection i, the $r''^{traffic}_{i,t}$ has to be calculated for all incoming legs with a red phase assigned during the last decision step.

$$r'^{traffic}_{i,t} = \begin{cases} 1, & D_{i,t} < D_{i,t-1} \\ -1, & D_{i,t} \geq D_{i,t-1} \end{cases} \quad (15)$$

$$r''^{traffic}_{i,t} = \begin{cases} 0, & O_{i,t} < O_{cri} \\ -1, & O_{i,t} \geq O_{cri} \end{cases} \tag{16}$$

Agents try to avoid wasting green time due to frequent phase switches. Thus, the reward for the agent's action is given by Equation (17), where y is a predefined integer ($0 < y < 10$) representing the acceptable number of phase switches in 10 steps.

$$r''^{traffic}_{i,t} = \begin{cases} 0, & n^{y}_{i,t} \leq y \\ -1, & n^{y}_{i,t} > y \end{cases} \tag{17}$$

For bus transit, if $h^{f}_{i,t} > h^{b}_{i,t}$, the bus needs to be prioritized to shorten the forward headway, thus equalizing the forward and backward headways. In this case, the transit reward is positive if the agent gives green priority to the bus-incoming lane in the following decision step. Similarly, in the case $h^{f}_{i,t} < h^{b}_{i,t}$, the bus needs to be held to equalize the forward and backward headways. The bus system reward can be concluded as Equation (14), where g^{bus}_{i} is the action that gives green priority to the bus-incoming leg at intersection i.

$$r^{transit}_{i,t} = \begin{cases} 1, & h^{f}_{i,t-1} > h^{b}_{i,t-1} \; and \; a_{i,t-1} = g^{bus}_{i} \\ 1, & h^{f}_{i,t-1} < h^{b}_{i,t-1} \; and \; a_{i,t-1} \neq g^{bus}_{i} \\ 0, & h^{f}_{i,t-1} = h^{b}_{i,t-1} \\ -1, & otherwise \end{cases} \tag{18}$$

3.2 Numerical test settings

In the following subsections, comparisons with four alternative traffic control strategies are implemented in our MARL framework. These traffic control strategies are selected because of their proximity to our approach or their wide use in the literature or field operation:

- Fixed control [9],
- Longest queue first [12],
- Max pressure [13],
- Centralized RL method [14].

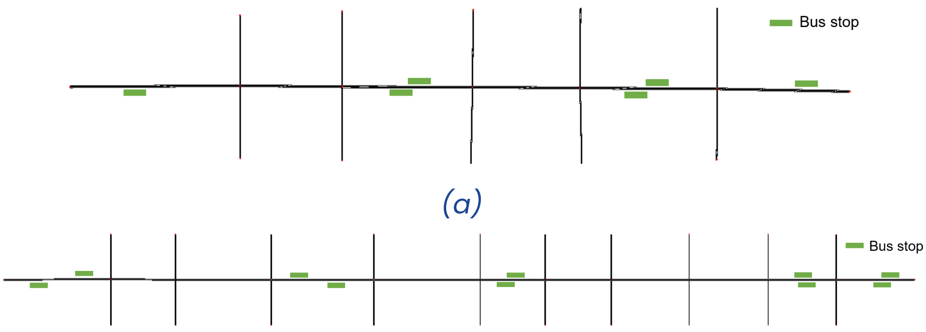
The analysis is performed according to two scenarios implemented in the SUMO simulation framework [6]. The scenarios differ in the bus lane layout (dedicated bus lane or mixed traffic lane).

Scenario 1: corridor networks with dedicated bus lanes:

This scenario includes one bus line driving on a dedicated bus lane in the corridor networks. To highlight the transferability of the strategies learned by the agents, the algorithm is trained and tested on distinct road configurations but with agents managing similar intersections. Therefore, there are five signalized intersections in the training corridor network (Figure 9a) and ten in the test arterial (Figure 9b). Four buses loop around the corridor network for the training process. All buses take a U-turn at the terminal of the arterial and continue the route in the opposite direction. In the test network, the number of buses is increased to eight to maintain a similar average space headway with the training network.

Scenario 2: corridor networks with mixed lanes:

The training and test networks are the same as the training network for scenario 1, except that the bus lanes are converted into mixed lanes.



(b)

Figure 9: Network structures in scenarios 1 and 2. (a) Training arterial, (b) test arterial.

3.3 Sensitivity analysis of the tradeoff between car traffic and bus transit in scenario 1

Before comparing the performance of the proposed algorithm with the benchmarks, an optimal tradeoff between the objectives related to car traffic and the objectives related to bus transit, modeled by the parameter c, needs to be determined. Consequently, a sensitivity analysis of the tradeoff is performed. Different values of c (ranging from 1 to 5) and two extreme cases are trained and tested with scenario 1. In the two extreme cases, only the reward of car traffic or bus transit is taken into account, represented by 'RL – traffic only' and 'RL – bus only', respectively. The reward curves for all agents in each training model are shown in Figure 10. In each case, they converge around 40 episodes of training.

The average queue length and standard deviation of space/time headways are com-

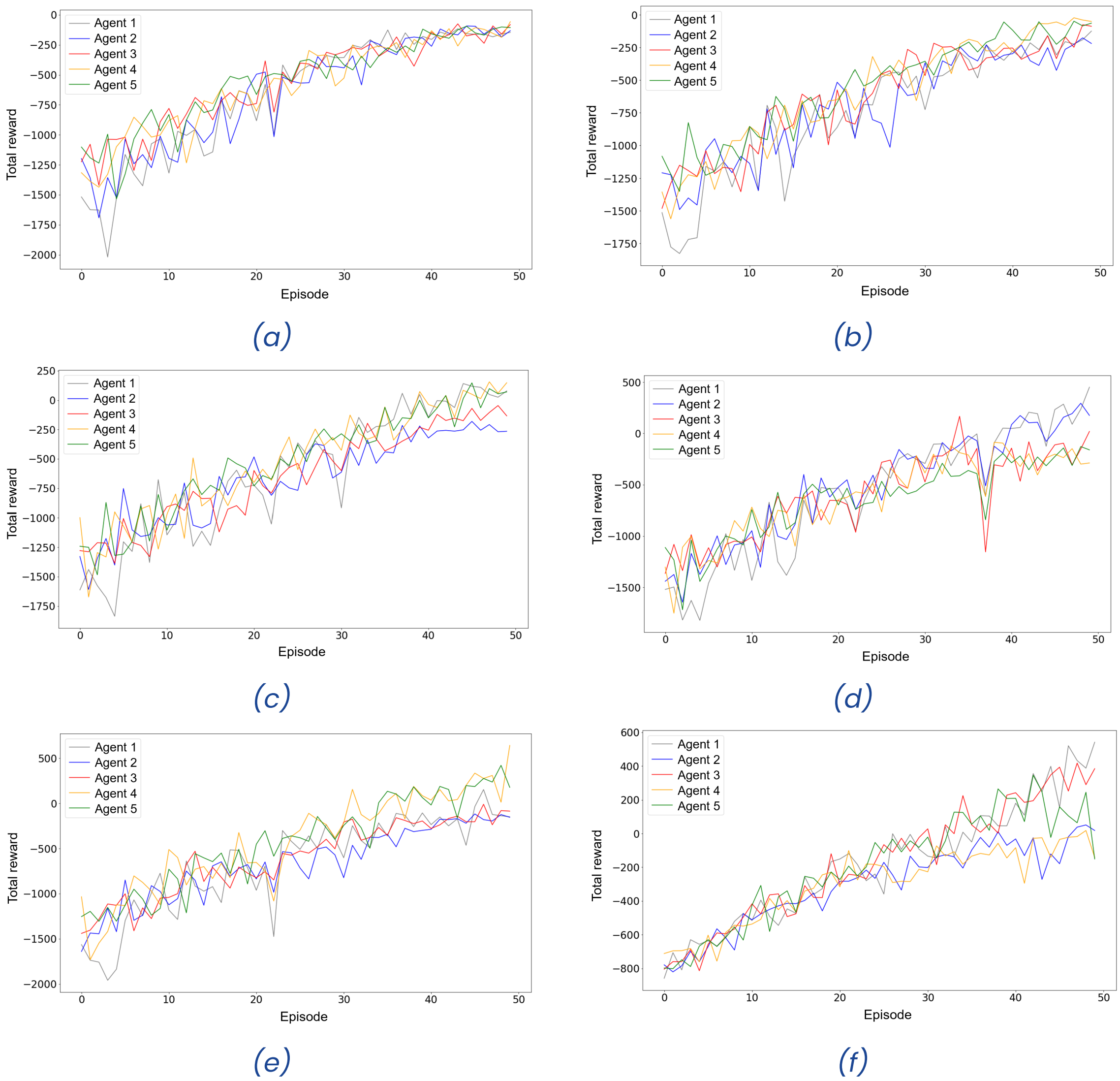


Figure 10: Total reward of each agent during the training episodes in RL-based models of (a) traffic only, (b) c = 1, (c) c = 2, (d) c = 3, (e) c = 4, and (f) bus only.

pared among these cases. The results are shown in Figure 11 and Table 1. Different random seeds are set to obtain various traffic demands during the tests. The notation of 'RL - n' refers to the proposed RL model with $c = n$. The average space headway in Table 1 is always 1516 m because the buses travel in a loop, and the space headway between the first and last bus is also taken into account. Thus, the average space headway is always the total length of a round trip divided by the number of buses, also denoted as *nominal headway*.

According to Table 1, the best performances on traffic delay and bus headway control are always from 'RL – traffic only' and 'RL – bus only', respectively, which decreases the average queue length and standard deviation of bus headway by 18.59% and 33.74% compared to the best performance among the benchmarks. This tests and proves the effectiveness of the proposed model both on car traffic control and on bus line control. A larger c value forces the model to give more weight to the bus than to the car traffic performance in the reward. Thus, the car traffic performance might be affected. Figure 11 compares the total queue length of all control strategies along the simulation process. The continuous increase in total queue length is observed from 'RL – traffic only' to 'RL - n' and then 'RL – bus only'. When c is set to 3, car traffic and bus transit performances appear to be well-balanced in all three tests. For the other values of c, only car traffic or bus performance is satisfied. Thus, $c = 3$ is chosen in the sequel. With this setting, the car traffic and bus transit rewards vary from -3 to 1 and -3 to 3, respectively. Bus transit has the same weight as car traffic in negative reward and a greater weight in positive reward.

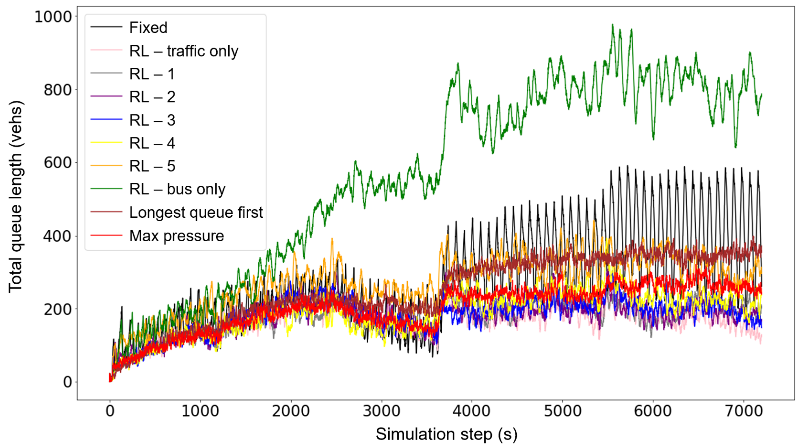
3.4 Test results for scenario 1

The performance of the proposed algorithm is further compared to the benchmark approaches. Table 1 presents the results of various simulation seeds in scenario 1. The average queue length of 'RL - 3' is always shorter than all the benchmark approaches. In an average of the three tests, 'RL - 3' decreases the average queue length by 30.91%, 19.59%, and 6.80% compared to the fixed control, longest queue first, and max pressure, respectively. According to Figure 11, the max pressure method performs slightly better than 'RL – 3' during the first half of the simulation. When the demand in crossroads increases during the second half of the simulation, 'RL – 3' becomes better than the max pressure method and obtains a better overall performance.

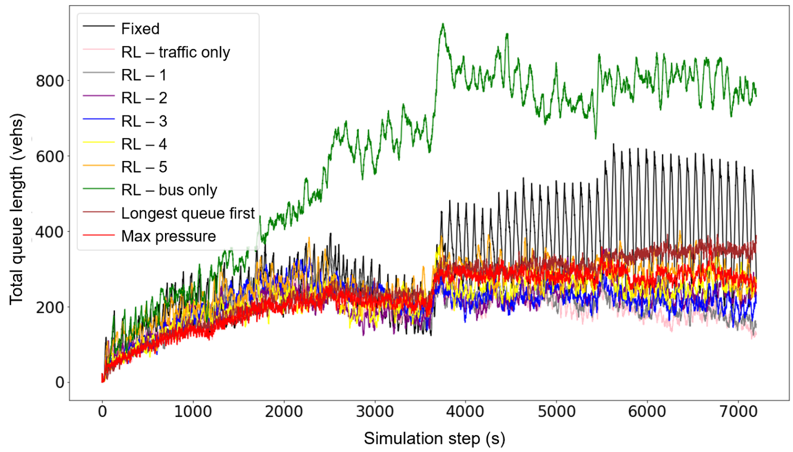
For bus headway control, 'RL – 3' also outperforms all benchmark strategies. The space headways of the control strategies in the test with a random seed=25000 can be seen in Figure 12. Figure 12 displays the travel distance of each bus during the simulation. Two lines approaching each other or even crossing each other means that bus bunching occurs. This phenomenon is observed in the trajectories of buses 1 and 2, buses 4 and 5 in the longest queue first approach, while it is observed for buses 1 and 2, buses 7 and 0 in the max pressure strategy. On the contrary, the RL-based methods effectively prevent buses from bunching. The percentage of small headways (less than 50% nominal headway) is calculated for each control strategy. They are 17.64%, 16.12%, 6.96%, and 3.03% for longest queue first, max pressure, 'RL – 3', and 'RL – bus only', respectively. The proposed approaches provide a significant improvement.

3.5 Test results for scenario 2

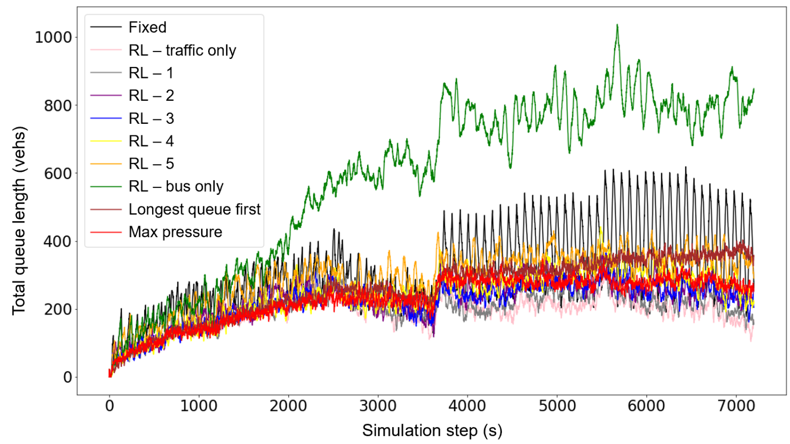
In this scenario, a centralized RL method from [14] is tested to compare with the proposed distributed RL algorithm since their goals are similar, namely: attempt-



(a)



(b)



(c)

Figure 11: Comparisons of queue length during the simulation in scenario 1 when (a) random seed = 15000, (b) random seed = 20000, and (c) random seed = 25000.

Table 1: Test results of each control method in scenario 1.

Random seed	Control method	Average queue length (vehs)	Space headway Average (m)	Space headway Standard deviation	Time headway Average (s)	Time headway Standard deviation
15000	RL - traffic only	163.11	1516	858.72	208.01	105.74
	RL - 3	179.59		517.94	194.64	49.89
	RL - bus only	563.33		394.29	207.30	42.74
	Longest queue first	250.48		872.79	197.20	103.59
	Max pressure	200.36		595.03	200.74	68.67
	Fixed	273.41		778.38	204.44	95.67
20000	RL - traffic only	190.62	1516	1098.14	207.85	143.25
	RL - 3	210.41		501.20	193.66	49.49
	RL - bus only	596.62		417.91	207.79	41.34
	Longest queue first	248.38		742.21	196.60	85.60
	Max pressure	224.94		620.76	200.83	69.16
	Fixed	301.24		773.84	204.44	95.20
25000	RL - traffic only	189.39	1516	818.38	203.81	100.57
	RL - 3	219.02		508.05	191.33	50.07
	RL - bus only	585.20		389.02	207.71	41.41
	Longest queue first	258.18		748.32	197.20	86.88
	Max pressure	227.16		779.26	201.34	92.51
	Fixed	305.35		778.64	204.44	95.63

ing to improve car traffic and bus performance through traffic signal control. In the centralized method, the traffic flow environment is built with Cell Transmission Method (CTM), a macroscopic and continuous platform. Since the traffic flow environment in this research is microscopic and discrete, Edie's definition is applied to the trajectory data to ensure a proper estimation of the density and outflow in the centralized RL method [15], [16].

Figure 13 compares the queue length of each control method. According to the results in Table 2 and Figure 13, the performance of the centralized RL method is acceptable but still worse than the decentralized methods. In the centralized algorithm, the reward definition is more suitable for a continuous traffic model, which does not encounter large fluctuations in the traffic and bus state over a short time interval. When applied to the microscopic and discrete traffic model in this research, lots of noise is added when estimating the traffic and bus state, which may explain why the model underperforms compared to the initial results.

Compared to fixed control and longest queue first methods, the proposed distributed RL algorithm decreases the standard deviation of space headway by 46.76% and 27.16%, respectively. For average queue length, the improvements are 9.30% and 13.55%, respectively.

Table 2: Test results of each control method in scenario 2

Control method	Space headway Average (m)	Space headway Standard deviation	Time headway Average (s)	Time headway Standard deviation	Average queue length (vehs)
Distributed RL	1637.78	392.98	283.35	64.00	124.50
Centralized RL		923.68	353.32	190.55	211.89
Fixed		738.18	273.88	118.28	137.26
Longest queue first		539.51	301.47	85.44	144.02
Max pressure		569.34	342.01	112.26	153.94

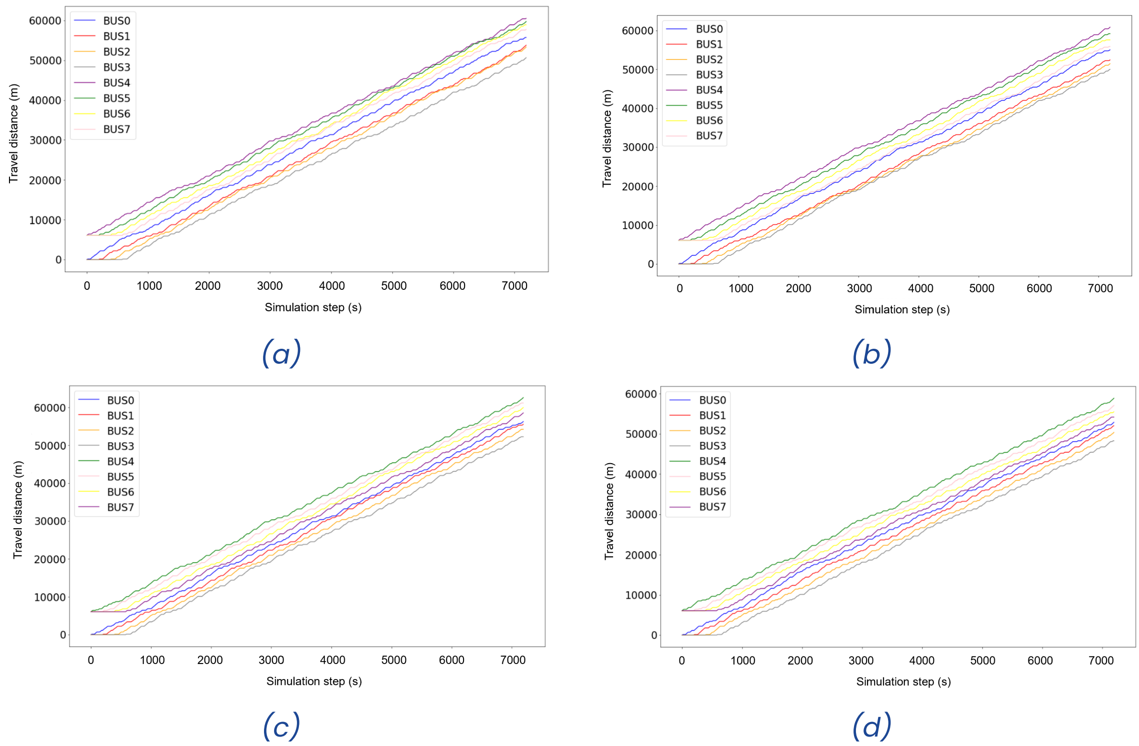


Figure 12: Travel distance of buses along the simulation step in scenario 1. (a) Longest queue first, (b) max pressure, (c) RL – 3, (d) RL – bus only.

3.6 Concluding remarks

This research proposes a MARL framework for traffic signal control in a multi-modal corridor-level network consisting of private car traffic and bus transit. Agents are designed in a decentralized framework with a continuous state space, leading to the implementation of deep reinforcement learning methods. The agents are trained to reduce the vehicle queue length and balance bus headways simultaneously. The tradeoff between car traffic-related and bus transit-related rewards is discussed via numerical experiments. When bus transit has the same weight as car traffic in negative reward and a greater weight in positive reward, the best performances are achieved. The proposed model is tested in various corridor scenarios, including different bus lane layouts (dedicated bus lanes or mixed traffic lanes) and road configurations. Both the scalability and portability are demonstrated with several stochastic-demand tests and by transferring locally learned strategies to similar intersection configurations. Compared to the best performance among the centralized RL method and model-based adaptive control methods, the proposed distributed RL method decreases the average queue length and the standard deviation of bus headway by 13.55% and 27.16%, respectively, in the corridor network with mixed traffic lanes.

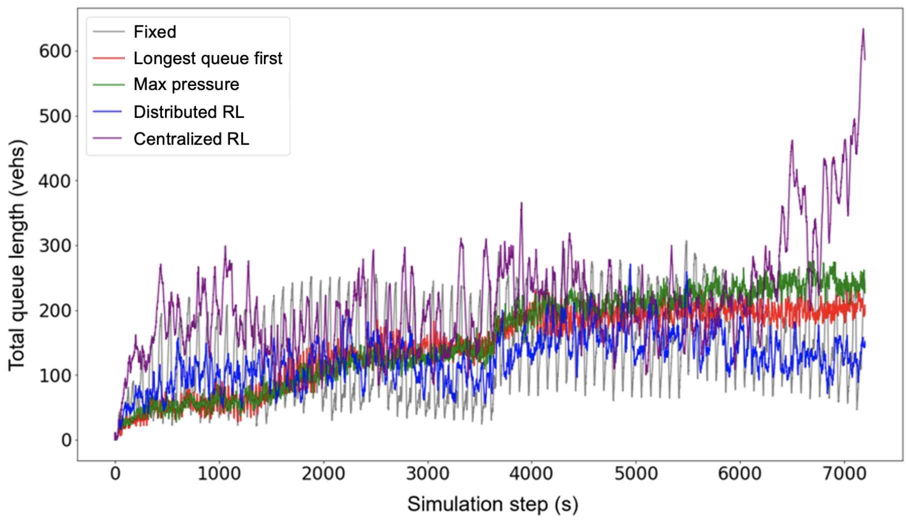


Figure 13: Total queue length during the simulation in scenario 2.

4 Network-level control

Traffic-responsive signal control is a cost-effective and easy-to-implement network management strategy, bearing high potential to improve performance in heavily congested networks with time-dependent congestion. Max Pressure (MP) distributed controller gained significant popularity due to its theoretically proven ability of queue stabilization and throughput maximization under specific assumptions. Perimeter control (PC) based on the concept of Macroscopic Fundamental Diagram (MFD) is a state-of-the-art aggregated control strategy that regulates exchange flows between homogeneously congested regions, with the objective of maintaining maximum regional travel production and prevent over-saturation. In this work, the effectiveness of network-wide parallel application of PC and MP strategies embedded in a two-layer control framework is assessed in a link-based macroscopic simulation environment. With the aim of reducing implementation cost of network-wide MP without significantly sacrificing performance gains, we evaluate partial MP deployment to subsets of nodes, indicated as critical by a node classification algorithm that we develop, based on node traffic characteristics.

The proposed two-layer controller is structured as follows. In the upper layer, perimeter control is applied in an aggregated scale between a set of homogeneously congested regions. At the end of every control cycle, the controller, based on inputs of aggregated regional vehicle accumulation, specifies the target inter-regional exchange flows for the next cycle, which are translated into the respective inter-regional green times between every pair of adjacent regions. The controller-specified inter-regional green times are then translated to exact green times per approach, for all PC controlled intersections, located on the boundaries between regions, by taking into account the actual boundary queues. In the lower layer, distributed control based on Max Pressure regulator is applied to a set of eligible intersections, in the interior of the regions. This set can contain all or a fraction of signalized intersections of the region, with the exception of those used for PC (if PC is applied in parallel). MP controllers do not communicate with each other or with any central control unit, but operate independently based on queue measurements directly upstream and downstream the controlled intersections, by adjusting green times of the approaches accordingly, at the end of every control cycle. The control layers do not exchange information, however their combined effect is indirectly considered by both controllers through the real-time traffic measurements that they receive as inputs. The mathematical formulations of both controllers are described in the following subsections, followed by a brief description of the utilized traffic model and traffic simulation process.

4.1 Max pressure control

4.1.1 Algorithm description

The traffic network is represented as a directed graph (N, Z) consisting of a set links $z \in Z$ and a set of nodes $n \in N$. At any signalized intersection n, I_n and O_n denote the set of incoming and outgoing links, respectively. The cycle time C_n and offset are pre-defined and not modified by MP. Intersection n is controlled on the basis of a pre-timed signal plan (including the fixed total lost time L_n), which defines the sequence, configuration and initial timing of a fixed number of phases that belong to set F_n. During activation of each phase $j \in F_n$, a set of non-conflicting approaches v_j get right-of-way simultaneously. The saturation flow of every link z is denoted as S_z. The turning ratio of an approach between links $i - w$, where $i \in I_n, w \in O_n$ is denoted as $\beta_{i,w}$ and refers to the fraction of the outflow of upstream link i that will move to downstream link w. The present version of MP assumes that turning ratios are known to the controller. Also, it assumes that phase sequencing is given as input and does not change during the control. Consequently, during every cycle, all phases will be activated for a minimum time in the same ordered sequence.

The control variables of this problem, denoted as $g_{n,j}(k_c)$, represent the duration of green of every stage $j \in F_n$ of all controlled intersections $n \in N$. Assuming that real-time measurements or estimates of the queue lengths and turning ratios are available, the pressure $p_z(k_c)$ of every incoming link $z \in I_n$ of node n, at the end of control cycle k_c, is computed as

$$p_z(k_c) = \left[\frac{x_z(k_c)}{c_z} - \sum_{w \in O_n} \frac{\beta_{z,w} x_w(k_c)}{c_w} \right] S_z, \quad z \in I_n \tag{19}$$

In Equation (19), $x_z(k_c)$ denotes the average number of vehicles that are present in link z during control cycle k_c and c_z denotes the storage capacity of link z. Queue normalization aims at considering the link size, so that the pressure of a smaller link is higher than that of a larger one with the same number of vehicles. In other words, pressure takes into account the likelihood of link queues - upstream and downstream - to spill-back in the following cycle. Pressures of all incoming links are calculated at the end of every cycle based on the latest queue measurements, which constitute the state feedback variables and are collected through proper instrumentation. Higher pressure indicates higher potential in traffic production, i.e. significant volume waiting to be served and enough available space in downstream links to receive it. Low or close to zero pressure indicates lower need for right-of-way time, due either to small queue upstream, or to lack of space downstream (links close to capacity). We should note that negative pressures are meaningless, so constraint $p_z(k) \geq 0$ must always hold.

Pressure is calculated for all incoming links $z \in I_n$ of node n by Equation (19). Then, the pressure corresponding to every stage j at control cycle k_c is defined as the sum of the pressures of all incoming links that receive right-of-way in stage j, as follows.

$$P_{n,j}(k_c) = \max\left\{0, \sum_{z \in v_j} p_z(k_c)\right\}, \quad j \in F_n \tag{20}$$

This metric is then used as weight for the distribution of the total available green time between the competing stages of the intersection.

After pressure values $P_{n,j}$ are available for every phase $j \in F_n$, the total amount of effective green time G_n, calculated as

$$G_n = C_n - L_n = \sum_{j \in F_n} g^{\star}_{n,j}, \quad n \in N \tag{21}$$

is distributed to the phases of node n in proportion to pressure values. In equation (21), $g^{\star}_{n,j}$ denotes the green time assigned to phase j by fixed-time analysis, using any of the standard algorithms. It holds that $g^{\star}_{n,j} \geq g_{n,j,\min}, \forall j \in F_n$. Since phases are activated in a strictly defined and non-changing order with a guaranteed minimum green time, green duration, $\tilde{g}_{n,j}(k)$, is assigned to phases proportionately to the computed pressures, as follows

$$\tilde{g}_{n,j}(k_c) = \frac{P_j(k_c)}{\sum_{i \in F_n} P_i(k_c)} G_n, \quad j \in F_n \tag{22}$$

Equation (22) provides the raw green times calculated by MP controller. However, these values cannot be applied directly because it must first be guaranteed that they comply with a set of necessary constraints. Therefore, an additional step is added in the signal update process, whose objective is to translate MP outputs of Equation (22) to practically applicable green times $G_{i,j}$. This is done by solving online, for every control cycle k_c, the following optimization problem:

$$\begin{aligned} \underset{G_{n,j}}{\text{minimize}} \quad & \sum_{j \in F_n} \left(\tilde{g}_{n,j} - G_{n,j}\right)^2 \\ \text{subject to} \quad & \sum_{j \in F_n} G_{n,j} + L_n = C_n \\ & G_{n,j} \geq g_{n,j,\min}, \; j \in F_n \\ & \left|G_{n,j} - G^{p}_{n,j}\right| \leq g^{R}_{n,j} \\ & G_{n,j} \in \mathbb{Z}^{+} \\ & \forall j \in F_n \end{aligned} \tag{23}$$

According to the above formulation, the applicable green times for every phase $G_{n,j}$, $j \in F_n$, should be as close to the non-feasible regulator-defined greens $\tilde{g}_{n,j}$

as possible, while satisfying a set of constraints. The first one is about maintaining constant cycle duration C_n. The second one ensures that all phases get a predefined minimum green duration $g_{n,j,\text{min}}$. In order to avoid potential instability of the system due to large changes in the signal timing happening too fast, we impose a threshold to maximum absolute change of green between consecutive cycles in the third constraint, where $G^p_{n,j}$ denotes the applied green times of the previous cycle and $g^R_{n,j}$ is the maximum allowed change between consecutive cycles. Finally, feasible green times must belong to the positive integers set. This type of integer quadratic-programming problem can easily be solved by any commercial solver fast enough to allow online solution for every control cycle. The solution of this optimization problem, i.e. variables $G_{n,j}, \forall j \in F_n$, are the new feasible phase green for node n which will be applied in the next control cycle. The above process is repeated at the end of the cycle for every controlled intersection, regardless of what is happening to the rest of the network. The controller only requires real-time queue information of the adjacent intersections and respective turning ratios and the algorithm is executed once per cycle.

4.1.2 Critical node selection method

Given the high requirements in monitoring equipment that increase proportionately to the number of controlled intersections, it is interesting to investigate how the impact of the MP control scheme is affected in the following cases: if only a fraction of the eligible intersections are controlled; whether some nodes are more critical than others in the sense of MP control for the same fraction; and what are the characteristics that would allow us to identify them. In an effort to reply to these research questions, we develop a node selection methodology based on current network traffic characteristics, by using principles of traffic engineering combined with an optimization approach. A set of three node assessment criteria is defined, which we linearly combine into a kind of node (dis)utility function. The coefficients need to be properly calibrated. Given traffic information of the current network situation (e.g. FTC case), a peak-period P is defined, based on the observed network state, as a set of time steps T_P. The selection process can be described, step by step, as follows:

- For every node n that is eligible to receive MP controller, the following three quantities are estimated: The first, denoted as m^n_1, represents the average node congestion level, as the mean over time of the mean occupancy (queue normalized over link capacity) of all incoming links $z \in I_n$ of node n, during peak period P. It is equal to

$$m^n_1 = \frac{1}{\|T_P\|} \frac{1}{\|I_n\|} \sum_{i \in T_P} \sum_{z \in I_n} \frac{x_z(i)}{c_z} \tag{24}$$

where i is the simulation time-step index, T_P is the set of time-step indices corresponding to the peak period P and $\|T_P\|$ is the size of set T_P. The second, denoted as m_2^n, represents the mean over time of the variance of link occupancy of all incoming links $z \in I_n$ of node n during peak-period P, computed by

$$m_2^n = \frac{1}{\|T_P\|} \sum_{i \in T_P} \text{var}(X_z^n(i)) \tag{25}$$

where $X_z^n(i) = \{x_z(i)/c_z | \forall z \in I_n\}$ is the set of normalized queues of all incoming links z of node n at time step i. The third quantity, denoted as N_c^n, represents the fraction of the peak period P, during which node n is considered 'congested'. In this analysis, we assume that a node is 'congested' during control cycle k if the average queue of at least one incoming link $z \in I_n$ of node n during k is higher than a preset threshold percentage p of its storage capacity, as shown by binary function $C_n(k)$ below.

$$C_n(k) = \begin{cases} 1, & \text{if } \frac{1}{t_c^n} \sum_{i=(k-1)t_c^n+1}^{kt_c^n} x_z(i) \geq p\, c_z, \quad \text{for any } z \in I_n \\ 0, & \text{else} \end{cases} \tag{26}$$

$$N_c^n = \frac{t_c^n}{\|T_P\|} \sum_{\forall k \in P} C_n(k), \tag{27}$$

In Equation (26), t_c^n denotes the control cycle size in number of simulation time-steps, i.e. $t_c^n = C_n/t$, where C_n denotes the control cycle duration of node n and t the simulation time-step duration. In Equation (27), the ratio $\|T_P\|/t_c^n$ is the number of control cycles that constitute peak period P. In other words, N_c^n represents (in the scale of 0 to 1) the fraction of the peak period P during which, at least one incoming link is congested and causes queue spill-back. In the current analysis, we set $p = 80\%$, since this is shown to significantly increase the probability for spill-back occurrence (see [17]).

- Then, the level of importance of each node n regarding MP control is estimated as a linear combination of the the above variables, denoted as R^n, as follows:

$$R^n = \alpha m_1^n + \beta m_2^n + \gamma N_c^n \tag{28}$$

Quantity R^n is then used as a base to rank nodes and drive the selection of the most critical ones. The coefficients in Equation (28) act as weights for the importance of every criterion and their values can be calibrated based on a trial and evaluation grid test, as described below.

- Finally, based on a target network penetration rate for MP control (i.e. percentage of eligible network nodes to receive MP controller), nodes are selected in sequence of increasing R^n, until the target number is reached.

4.2 Perimeter Control

4.2.1 Proportional-Integral regulator for MFD-based gating

The concept of gating in perimeter control strategies for multi-region systems consists of controlling transfer flows in the perimeter or the boundaries of the protected regions, in order to prevent vehicle accumulation from rising excessively and leading to congestion phenomena. Flow control can be applied by means of adaptive traffic signals on the boundaries between regions, where green time of the controlled approaches is periodically adjusted, based on the actual traffic state of the region. State information can be provided in real-time by loop detectors installed properly in the interior of the region, or by other types of traffic measuring equipment. The concept of the Macroscopic Fundamental Diagram (MFD) enables the development of reliable feedback control strategies that assess the network state based only on measured accumulation, which is associated with travel production. Driven by the characteristics of the MFD curve, during peak-hours PC strategies manipulate perimeter inflows so that accumulation in high-demand regions remains close to critical - for which travel production is maximum - and does not reach higher values belonging to the congested regime of the MFD if possible.

In this work no traffic states prediction is considered, as we follow a less computationally expensive approach, where the system is controlled in real-time through a classical multi-variable Proportional-Integral (PI) feedback regulator (see [18]), as follows:

$$\mathbf{u}(k_c) = \mathbf{u}(k_c - 1) - \mathbf{K}_P \left[\mathbf{n}(k_c) - \mathbf{n}(k_c - 1)\right] - \mathbf{K}_I \left[\mathbf{n}(k_c) - \hat{\mathbf{n}}\right] \tag{29}$$

In the above, $\mathbf{u}(k_c)$ denotes the vector of control variables u_{ij} for control interval k_c, which represent the average green times corresponding to the controlled approaches between adjacent regions i and j (heading from i to j), as well as to the external perimeter approaches of every region (if external gates exist), denoted as u_{ii}; $\mathbf{n}$ is the state vector of aggregated regional accumulations n_i; $\hat{\mathbf{n}}$ is the vector of regional accumulation set-points $\hat{n_i}$; and $\mathbf{K}_P$, $\mathbf{K}_I$ are the proportional and integral gain matrices, respectively. If Equation (29) is written in analytical form instead of matrix form, a system of equations will be produced. Every equation specifies the average green time, for the next control interval, for all nodes in specific direction between pairs of adjacent regions (e.g. u_{ij} and u_{ji} for adjacent regions $i, j \in \mathcal{N}$), while the last $\|\mathcal{N}\|$ equations refer to the average green time of all external approaches of each region. The set-points are decided based on the regional MFDs. The number of controlled regions and the respective MFD shapes depend on the network partitioning to a set of homogeneously congested regions $\mathcal{N}$, which can involve real or simulated traffic data. In order to be functional, the PI regulator requires as inputs the real-time regional accumulations, the set-points, as well as the proportional and integral gain matrices. Regional accumulations are supposed to be provided by loop detectors or other measuring equipment, properly distributed in the network. In this work we assume perfect knowledge of regional accumulations, that are averaged over the control interval.

The PI controller is activated at the end of every control interval and only when real-time regional accumulations are within specific intervals, in the proximity of the specified set-point, i.e. activated when $n_i \geq n_{i,\text{start}}$ and deactivated when $n_i \leq n_{i,\text{stop}}$, for $i \in \mathcal{N}$ and typically $n_{i,\text{stop}} < n_{i,\text{start}}$. This is important as early activation of the PI regulator (i.e. for low accumulation) can lead to signal settings aiming at increasing congestion in the controlled areas, so that production gets closer to critical, which is the target. However, such a policy can accelerate congestion and compromise the system performance. From Equation (29), the average green time $u_{ij}(k_c)$ of all controlled approaches on the border between regions i and j with direction from i to j is calculated. Based on this average value, the exact green time for every specific intersection is calculated according to the process described in the following subsection. After deactivation of the controller, the FTC signal plan for all PC intersections is gradually restored.

4.2.2 Green time calculation for PC intersections

After average green time u_{ij} for all PC controlled approaches between adjacent regions i and j is defined from Equation (29), it is used as base to define the exact new green duration for the respective phases containing the approaches leading from region i to j. However, since not all intersections serve the same demand, green time assignment is more efficient if decision takes into account current queue lengths of the respective approaches. Moreover, new greens are subject to a set of constraints, similar to the ones imposed in the case of MP signal update process (see Equation (23)).

Hence, for every u_{ij} that is specified by the PI controller, an optimization problem is solved for determining the exact green duration of the primary and secondary phases, denoted as p and s, respectively, of all controlled intersections with direction from i to j. Primary phase p includes approaches with direction from i to j and secondary phase(s) s include approaches in the vertical (parallel to the boundary between regions) or the opposite direction (from j to i). The sum of available green of primary and secondary phase remains constant. Assuming that the set of controlled nodes in the direction from i to j is denoted as $\mathcal{M}_{ij}$, m is the node index, $G_{m,p}$ is the final green of the primary phase, $G_{m,s}$ is the final green of the secondary phase, $G_{m,t}$ is the sum of available green time for primary and secondary phases,

$Q_{m,p}$ and $Q_{m,s}$ are the sum of the observed queues of all incoming links belonging to the primary and secondary phases during the last control cycle, respectively, and $S_{m,p}$ and $S_{m,s}$ denote the sum of saturation flows of all approaches of node m belonging to primary phase p and secondary phase s, respectively, the following optimization problem is formulated:

$$\begin{aligned}
\underset{G_{m,p},G_{m,s}}{\text{minimize}} \quad & \theta_1 \left(\sum_{m \in \mathcal{M}_{ij}} G_{m,p} - u_{ij} \|\mathcal{M}_{ij}\| \right)^2 + \\
& \theta_2 \sum_{r \in \{p,s\}} \sum_{m \in \mathcal{M}_{ij}} Q_{m,r} \left(1 - \frac{G_{m,r} S_{m,r}}{Q_{m,r} + 1} \right)^2 \\
\text{subject to} \quad & \\
& G_{m,p} + G_{m,s} = G_{m,t} \\
& G_{m,r} \geq g_{m,r,\text{min}}, \\
& \left| G_{m,r} - G^{pr}_{m,r} \right| \leq g^{R}_{m,r} \\
& G_{m,r} \in \mathbb{Z}^{+} \\
& \forall r \in \{p, s\}, \forall m \in \mathcal{M}_{ij}
\end{aligned} \tag{30}$$

In the above, we seek to minimize the sum of two terms, the importance of which is weighted by parameters θ_1 and θ_2, respectively. The first term aims at minimizing the difference between the finally assigned total green of the primary phase of all PC intersections $m \in \mathcal{M}_{ij}$ of the approach $i - j$ and the total green indicated by the controller for the same approach. The second term aims at achieving green time distribution proportionally to the observed queues of the primary and secondary phases of the controlled intersections. This is done by minimizing the sum of the differences between the outflow that will be achieved in the primary and secondary phases of the PC nodes based on the finally assigned green time and the queue that was observed in the respective phases during the last control cycle.

The constraints of the problem are the following: the first one is about maintaining cycle duration, i.e. ensuring that the sum of primary and secondary phases remains constant; the second one dictates that minimum green $g^{R}_{m,r}$ is assigned to all phases, where r is an index that indicates the type of phase (primary p or secondary s); the third one ensures that maximum absolute change between new green $G_{m,r}$ and green of the previous control interval $G^{pr}_{m,r}$ is below the preset threshold of $g^{R}_{m,r}$; and the forth one dictates that green time intervals are integer. The new control plans take effect, for every intersection, after the end of their ongoing cycle.

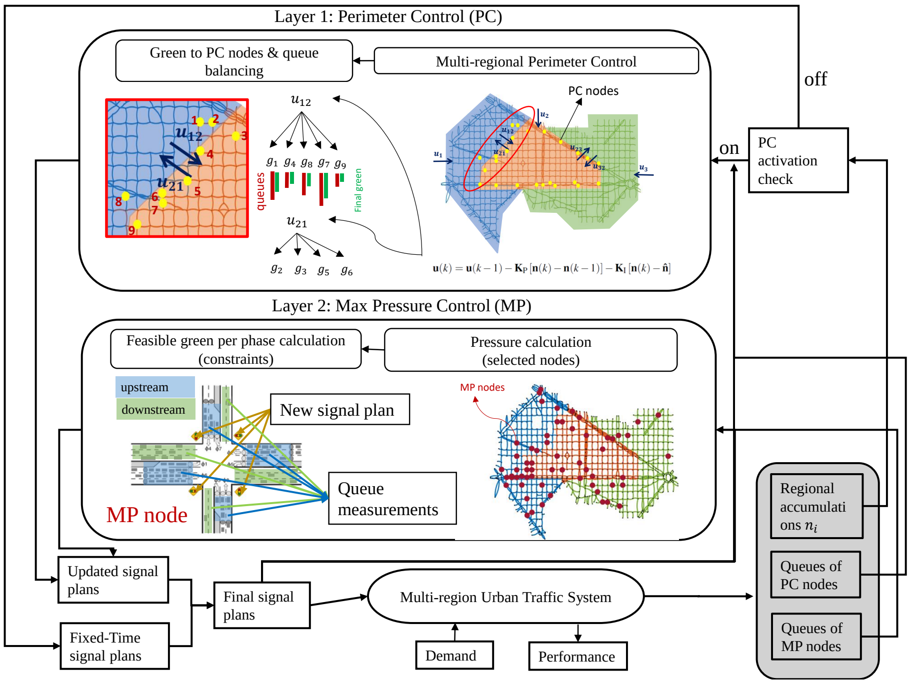


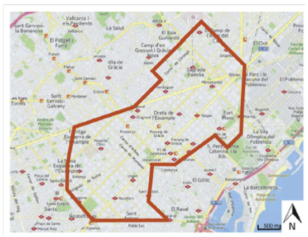
Figure 14: Detailed structure of the hierarchical control scheme combining perimeter control and Max Pressure control.

4.3 Max pressure and perimeter control

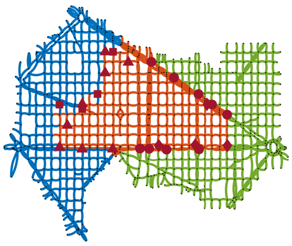
A detailed schematic depiction of the proposed two-layer control framework is shown in Figure 14. Based on the most recent collection of traffic information during the last control cycle, (aggregated regional accumulations $n_i, i \in \mathcal{N}$, where i is the region index), queues of primary and secondary phases of PC nodes ($Q_{m,p}, Q_{m,s}$, $\forall m \in \mathcal{M}_{ij}$, $\forall \{i, j\} \in \mathcal{N}$ with i adjacent to j), and queues of upstream and downstream links of MP controlled nodes ($x_z, \forall z \in I_n \cup O_n$, for all n belonging to the set of MP controlled nodes), the controllers decide on the updated signal plans of the respective intersections, according to the processes described in the previous sections. Perimeter control is activated/deactivated under specific conditions (when regional accumulations are above/below predefined thresholds). When PC is activated, first a PI controller defines the average new green of the controlled nodes for every approach $i - j$ between adjacent regions, as well as for the external perimeter of each region. Then this average green time is distributed to the specific controlled nodes proportionally to the recorded queues of the primary and secondary phases, in order to prevent queues from growing excessively and impeding traffic upstream. In the second layer, Max Pressure calculates the pressure of each phase of every controlled node, based on the recorded queues upstream and downstream. Afterwards, the pressures are translated to new green time for every phase while satisfying a set of constraints. Finally, new signal settings are applied to all controlled intersections for the next cycle, before the process is repeated.

4.4 Implementation to a large-scale network via simulation

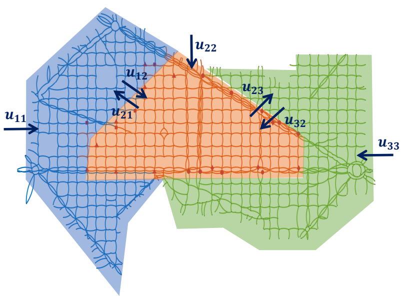
The proposed adaptive signal control schemes are evaluated using the urban traffic model described in [19], which was coded from scratch and executed in Matlab R2020a, while optimization problems 23 and 30 are solved by Gurobi 9.1.2 solver. Real-life large-scale signalized traffic network of Barcelona city center is used as case study and Fixed-Time Control (FTC) settings with no adaptive element, are used as benchmark case. Both MP and PC schemes are applied separately, as well as in combination, for two different demand scenarios that create moderate and high levels of congestion in the FTC case, respectively. All MP cases are tested in full-network implementation and in node subsets for different penetration rates, selected by the proposed algorithm as well as randomly, for comparison. Detailed description of the network, the simulation settings and the performed experiments are provided in this section.



(a)



(b)



(c)

Figure 15: (a) Map of the studied network of Barcelona city center; (b) Model of Barcelona network clustered to 3 homogeneous regions, with annotation of nodes used for PC; (c) schematic representation of controlled approaches for perimeter and boundary flow control, with green time per approach as control variable.

4.4.1 Case study

The traffic network utilized in this study is a replica of Barcelona city center, in Spain, as shown in Figure 15a. It consists of 1570 links and 933 nodes, out of which 565 represent signalized intersections with fixed-length signal control cycles ranging from 90 to 100 sec. Links have from 2 to 5 lanes. All control schemes are tested in two different demand scenarios, one leading to moderate and one to high congestion in FTC settings. For the purpose of PC implementation, the network is partitioned into three regions of similar traffic distribution, according to the clustering method described in [20]. The resulting regions are displayed in different colors in Figure 15b. In the same figure, the controlled intersections used for gating through PC are displayed. Figure 15c schematically represents the perimeter and boundary flow control variables u_{ij}, which denote the average green time of all approaches in the direction from i to j, while u_{ii} denotes the equivalent time for all external approaches of region i. All nodes on the external perimeter of all three regions are controlled. This partitioning leads to 4 state and 7 control variables, which are also depicted in Figure 15c.

The dynamic profile of total generated demand, for both demand scenarios, consist of a 15-minute warm-up period followed by a 2-hour constant peak demand, for a total simulation period of 6 hours, which is representative of the morning-peak. Medium demand includes 251k trips generating at 88 origin links and heading towards 104 destination links, whereas high demand scenario includes 316k trips, from 123 origins to 130 destinations. Figure 16 graphically describes the spatial dis-

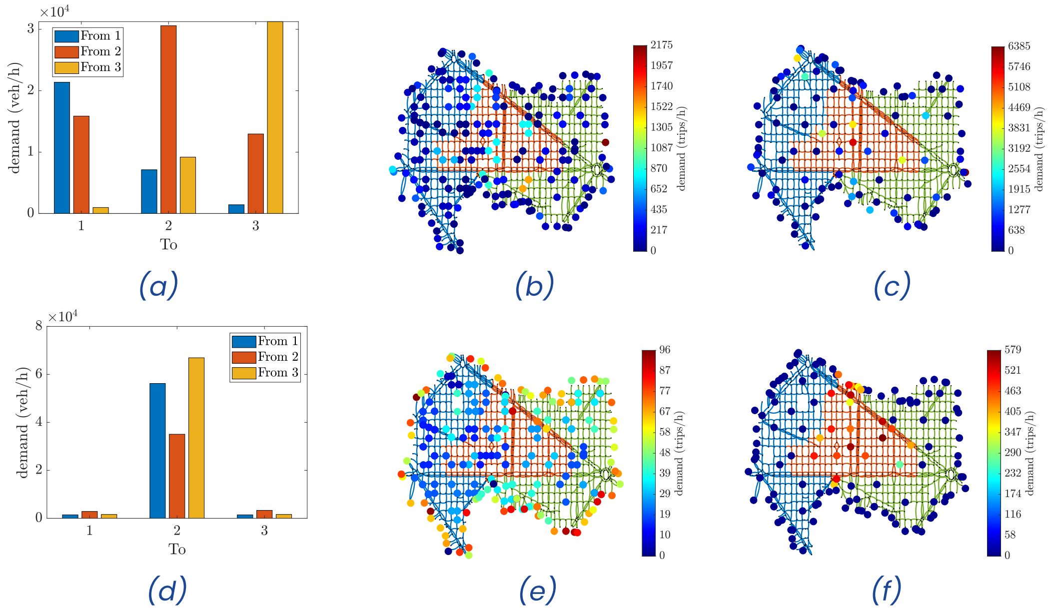


Figure 16: Description of the two demand scenarios in peak period: medium (a)-(c) and high (d)-(f). (a) and (d): aggregated trip distribution between regions; (b) and (e): trip origin density; (c) and (f): trip destination density.

tribution of demand for both scenarios: (a) to (c) refer to medium demand while (d) to (f) refer to high demand. The distribution of trips between regions, resulting from suitable clustering process as described below, is presented in the first graph of each row (a and d). Each bar corresponds to the total demand originating from each origin region, each represented by different color, while horizontal axis indicates destination region. Second graph of each row (b and c) presents the spatial distribution and density of origin points, represented by dots on the network map. Dot color indicates demand volume per origin point, as described by the respective colorbar. Similarly, third graph of each row (c and f) represents the same information for destination points. Apart from difference in total number of trips, the two demand scenarios lead to different traffic patterns, since high demand scenario has a clear directional profile, with trips mainly originating from the periphery (regions 1 and 3) and heading towards the center (region 2), whereas medium demand shows more diverse trip distribution between regions, with more intra-regional trips in all regions, plus a less intense directional pattern towards the city center (region 2). The objective of using different demand scenarios is to test the efficiency of the proposed control schemes under different traffic conditions. It should be noted, that no demand information is required by the controllers, which only receive real-time queue measurements and/or turn ratios as inputs.

4.5 Control scenarios

The case of fixed-time, static signal control, labeled as 'FTC', is used as benchmark for all tested control schemes. Firstly, MP is evaluated as single control strategy (no PC applied simultaneously). With the aim of investigating the performance of MP in relation to the number and location of the controlled nodes, we test the following scenarios:

- MP control of all eligible network nodes. All signalized nodes receive MP controller.
- MP control of fraction of network nodes, selected randomly. For each penetration rate of $5\%, 10\%, 15\%, 20\%, 25\%$, 10 randomly created MP node sets are evaluated through simulation.
- MP control of fraction of network nodes, selected by the proposed algorithm. For the same penetration rates as above, MP node sets are created according to decreasing values of R, after suitable parameter calibration. FTC simulation results are used for calculating variables m_1^n, m_2^n and N_c^n, and thus quantity R.

Afterwards, MFD-based PC based on the PI controller, is applied first as a single control scheme and then in combination with distributed MP control, integrated in the two-layer framework. Similar to the case of single MP, the MP layer of the combined scheme is tested for several controlled node layouts, in various penetration rates, as well as in full network implementation (100 % eligible nodes). The following scenarios are evaluated:

- Single PC for 3-region system
- PC for 3-region system combined with MP control to all eligible network nodes
- PC for 3-region system combined with MP control to fractions of eligible network nodes, selected randomly. For each penetration rate of 5%, 10%, 15%, 20%, 25%, 10 sets of randomly selected eligible nodes are formed (same as in single MP scenarios). MP control is evaluated in parallel with PC scheme.
- PC for 3-region system combined with MP control to fractions of eligible network nodes, selected by the proposed algorithm. For the same penetration rates as above, MP node sets are created according to decreasing values of R, after suitable parameter calibration. FTC simulation results are used for calculating variables m_1^n, m_2^n and N_c^n, and thus quantity R.

4.6 Experiment settings

The control scenarios under evaluation are simulated for a 6-hour time period for medium demand, and for a 8-hour period for high demand, representative of typical morning peak. The time window for turn ratio update is 15 minutes. MP regulator is active for all controlled intersections during the entire simulation time and signal plans are updated at the end of every cycle. Only phases lasting longer than 7 seconds in the pre-timed scheduling are eligible for change by the regulator and minimum green time allowed per phase $g_{n,j,\text{min}}$ is also 7 seconds. Maximum allowed fluctuation of green time between consecutive cycles, $g_{n,j}^R$ is set to 5 seconds, to avoid instabilities. Inputs for MP regulator are link queues of incoming and outgoing links, as well as estimated turn ratios for all approaches of controlled nodes.

Regarding PC implementation and based on the network partitioning in 3 regions described above, the PI controller of Equation (29) regulates 7 control variables u_{ij}, which represent the average green time of all controlled intersections in the approaches between adjacent regions 1-2, 3-2, 2-1, 3-1, as well as those of the external perimeter of each regions 1, 2 and 3 (see Figure 15c), using 3 state variables, i.e. the regional accumulations $n_i, i = 1, 2, 3$. Therefore, $\mathbf{u}$ is a 7x1 vector, $\mathbf{n}$ and $\hat{\mathbf{n}}$ are 3x1 vectors, while proportional and internal gain matrices $\mathbf{K}_P$ and $\mathbf{K}_I$ are of dimensions 7x3. Four first rows refer to boundary approaches in the order they are listed above and three last rows refer to external perimeter approaches of regions 1 to 3. Due to different directional patterns of the two demand scenarios, PI parameters differ slightly. For the medium demand scenario, setpoint accumulation for the three regions are $\hat{n_1} = 10000, \hat{n_2} = 12000, \hat{n_3} = 6800$ veh, proportional gain matrix is $\mathbf{K}_P = [15, -10, 0; 0, -5, 10; -15, 10, 0; 0, 5, -10; -20, 0, 0; 0, -20, 0; 0, 0, -20]$ and integral gain matrix is $\mathbf{K}_I = \mathbf{K}_P \times 10$, $n_{i,\text{start}} = \hat{n_i}$, $n_{i,\text{stop}} = 0.85\, \hat{n_i}$, for every region $i = 1, 2, 3$. For the high demand scenario, accumulation setpoint is $\hat{n_1} = 4500, \hat{n_2} = 9200, \hat{n_3} = 8000$ veh, proportional gain matrix is $\mathbf{K}_P = [18.5, -2.1, 0; 0, -3.3, 6.8; -13.3, 5.6, 0; 0, 4.6, -3.5; -16.4, 0, 0; 0, -9.8, 0; 0, 0, -10.5]$ and integral gain matrix is $\mathbf{K}_I = [18, -69, 0; 0, -69, 62; -44, 24, 0; 0, 1, -40; -54, 0, 0; 0, -30, 0; 0, 0, -51]$, $n_{i,start} = 0.99\, \hat{n_i}$ and $n_{i,stop} = 0.93\, \hat{n_i}$, for every region $i = 1, 2, 3$. In all cases, activation of the PI controller happens when at $n_i \geq n_{i,\text{start}}$ for at least 2 regions $i = 1, 2, 3$, while deactivation happens when $n_i < n_{i,\text{stop}}$ for all 3 regions. PC application requires inputs of aggregated regional accumulations n_i for all regions $i = 1, 2, 3$ for the PI controller, while for the phase of applicable green time calculation (optimization problem 30), queue measurements for all approaches of primary and secondary phases of PC intersections, $Q_{m,p}$ and $Q_{m,s}$ respectively, are required, together with the latest applied signal plan. After performing grid search optimization, we found that the best performing values are $\theta_1 = 0.4$ and $\theta_2 = 0.9$, which we used in all cases. The process is repeated every 90 seconds.

4.7 Results

4.7.1 Single Max Pressure

Results of simulation experiments concerning single Max Pressure schemes are presented in this section. The case of full-network MP implementation is compared to the FTC case and to different scenarios of partial MP implementation, in different fractions of network nodes, selected both randomly and by the proposed methodology. For the process of node selection, the considered peak-period is 2-hour long and consists of $T_P = 80$ control cycles of 90 seconds. FTC case is simulated and results are used for the calculation of m_1, m_2 and N_c. After performing a trial-evaluation grid search for the medium demand scenario, the best performing values are $\alpha = 0.6$, $\beta = -1.8$ and $\gamma = -1$, and selection is done starting from nodes with lowest R. In this way, the algorithm prioritizes selection of nodes with relatively high queue length variance and spill-back occurrence during peak time but with moderate mean queue lengths. In Figure 17 the node selection process for the case of medium demand is pictured for penetration rates 5%, 10% and 15%, where dots represent all network signalized nodes. First column graphs (a, d, and g) show the relation between m_1 and m_2, while second column graphs (b, e and h) show the relation between N_c and m_2, all calculated based on simulation results of FTC case. Blue dots represent nodes that are selected to receive MP controller according to the proposed algorithm. Third column graphs (c, f and i) visualize the spatial distribution of the selected MP nodes, depicted as red dots.

The performance of single MP network control for the medium demand scenario,

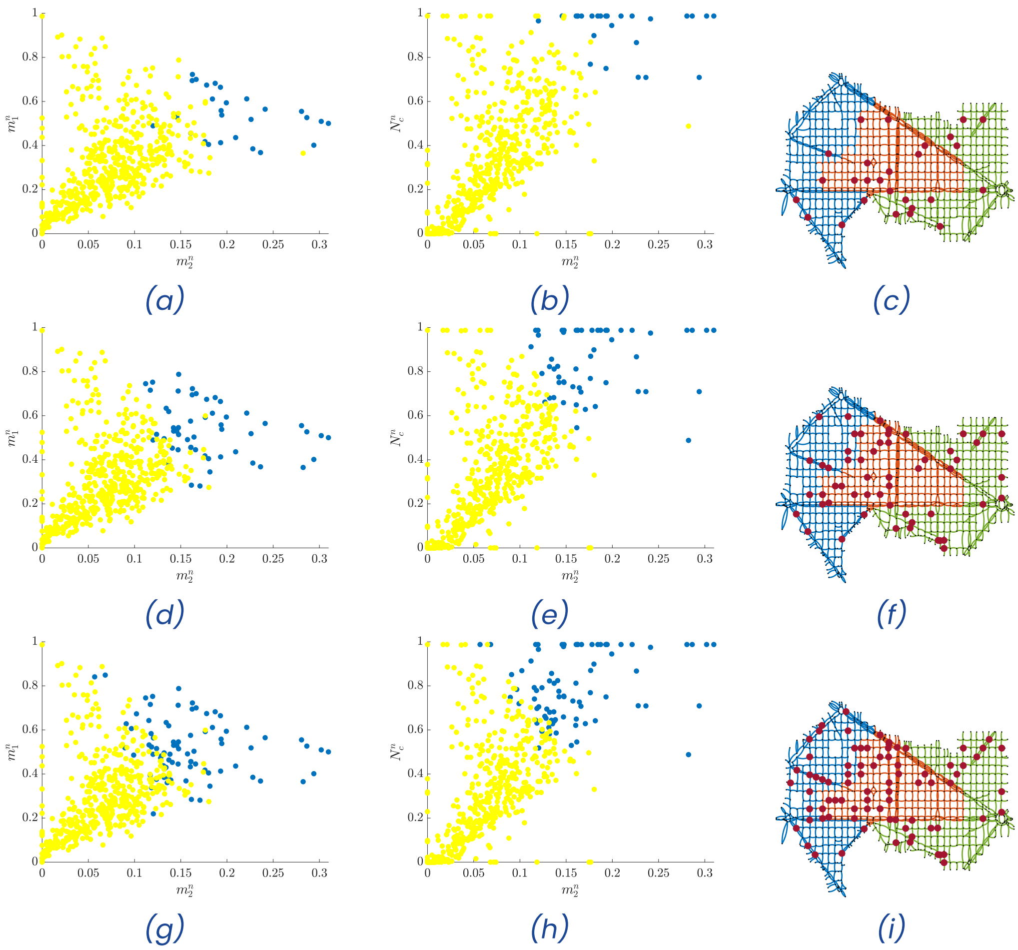


Figure 17: Visualization of the MP node selection process for the case of medium demand, according to the proposed method. Each row refers to different penetration rate (5%, 10% and 15%, from top to bottom). First and second column graphs show relations between selection variables m_2 - m_1 and m_2 - N_c, respectively, for all network nodes, for the benchmark case of FTC. Blue dots represent the selected nodes for MP control. Third column figures show the plan of the studied network, partitioned in 3 regions, with the spatial distribution of the selected MP intersections shown as red dots.

for different fractions of MP controlled nodes, as well as for the standard full network implementation (penetration rate of 100%), is shown in Figure 18. On the vertical axis the percentile improvement of total travel time (vehicle-hours traveled or VHT) with respect to FTC scenario is shown. Each boxplot refers to 10 cases of random selection of MP controlled intersections, corresponding to the respective penetration rate. The case of 100% rate refers to full MP network implementation. Red triangles represent scenarios where controlled nodes are selected according to the method described in section 4.1.2. Firstly, we observe that for the case of medium demand, almost all MP scenarios lead to improved total travel time, even those with randomly selected MP nodes. However, most cases of node selection made by the proposed method significantly outperform random assignment. In

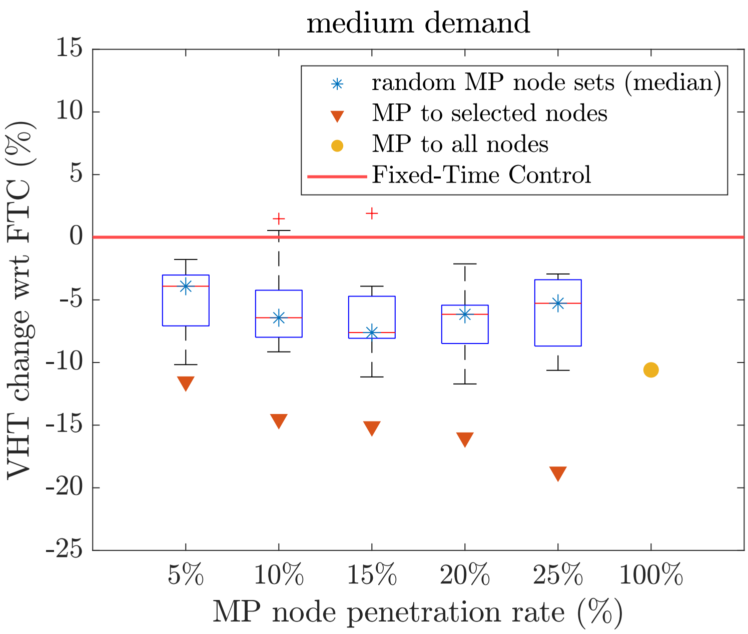


Figure 18: Comparison of Total Travel Time improvement, with respect to FTC scenario, of single Max Pressure application, for medium demand scenario. Boxplots refer to 10 randomly created MP node sets for every penetration rate, red triangles refer to node selection based on the proposed method and yellow dot represents the full-network implementation.

fact, we notice that the higher the number of controlled nodes, the larger the difference between random and targeted selection performance. These observations indicate that the proposed selection process is successful in identifying critical intersections for MP control. Interestingly, the case of installing MP regulator to all network nodes leads to smaller improvement than those including only a fraction of controlled nodes according to the proposed algorithm. More specifically, with only 10% of critical nodes the system travel time improves by 14.5% and with 25% of critical nodes, it improves by 18.8%, while in the case of controlling all nodes, it improves only by 10.6%. This remark indicates not only significant cost reduction can be achieved by reducing the number of controlled nodes through the proposed selection process, but also system performance can be increased. However, this behavior is observed for moderate demand, where the network does not reach highly congested states in the FTC case.

A different behavior is observed in the case of high travel demand, both in the node selection pattern and in the corresponding network performance. Firstly, we observed that using the same values for parameters α, β, γ as in moderate demand case and following the same selection pattern leads to poor system performance, very similar to the one of random selection. Therefore, parameter optimization is done again, by performing a new trial-evaluation test, specifically for the high demand scenario. The new values found are $\alpha = -0.72$, $\beta = -0.4$, $\gamma = -0.2$. In this case, we observe a change of sign in α, which directs the selection towards nodes

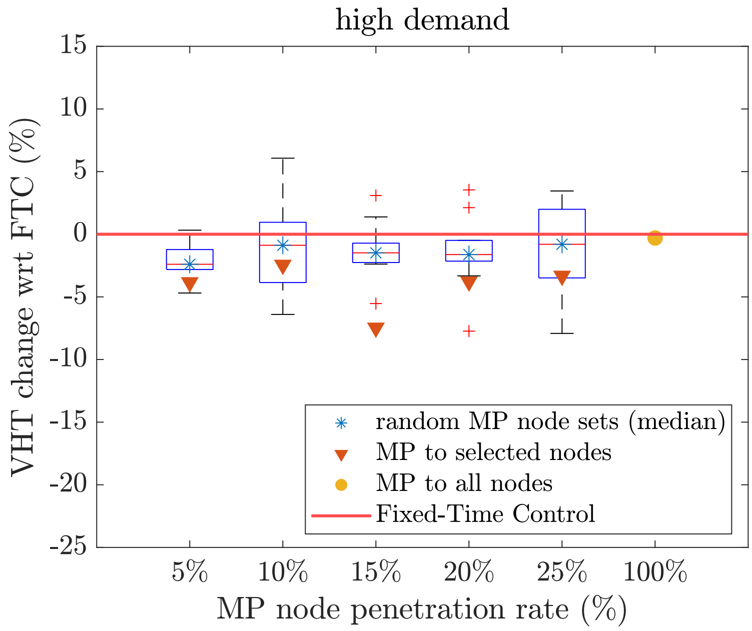
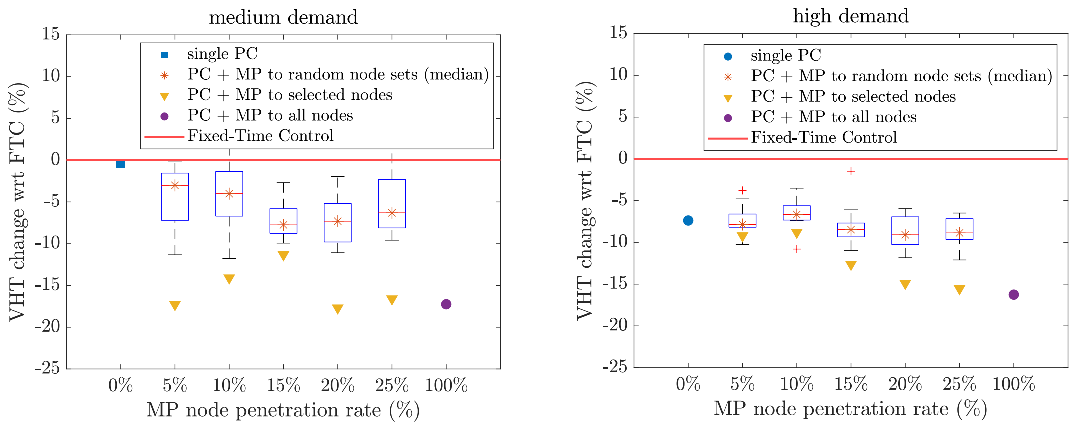


Figure 19: Comparison of Total Travel Time improvement, with respect to FTC scenario, for single Max Pressure application for high demand scenario. Boxplots refer to 10 randomly created MP node sets for every penetration rate, red triangles refer to node selection based on the proposed method and yellow dot represents the full-network implementation.

with high mean queues (m_1) while high queue variance (m_2) and spill-back duration (N_v) are given lower weights. In other words, in this case, the best found selection pattern prioritizes nodes with high mean queues.

Regarding system performance of single MP in high demand scenario, results show relatively smaller improvement with respect to FTC than in the case of medium demand. This can be seen in Figure 19, Again, boxplots refer to 10 cases of randomly selected node sets for MP control per penetration rate. Firstly, it is interesting that the case of full-network MP control (yellow dot) leads to practically zero improvement compared to FTC. However, smaller MP penetration rates in best case result in improvement between 3% and 7%. The effectiveness of node selection method seems to drop, even with re-optimized parameters of function R, since we observe a few random sets performing better than the ones of the proposed method. Performance of targeted selection is always better than the median of the random set though, and in the case of 15%, for which parameter optimization was performed, performance is significantly better than random cases. Based on these remarks, we can infer that partial MP implementation, except for being less costly, can also lead to improved performance, especially in highly congested networks with single MP control, which implies that significant spatial correlation exist between performance of MP controlled nodes that can act detrimentally to system performance.



(a) (b)

Figure 20: Comparison of Total Travel Time improvement, with respect to FTC scenario, for single PC and combined PC plus MP implementation, for (a) medium, and (b) high demand. Graphs show the performance for different MP node penetration rates (0% is single PC) for the two-layer PC+MP framework. Boxplots refer to 10 random MP node selections per rate while yellow triangles refer to node selection based on the proposed method.

4.7.2 Two-layer framework: Perimeter control combined with Max Pressure

Results of the two-layer framework combining PC with MP schemes are discussed in this section. Simulation results in terms of performance improvement with respect to FTC case are shown in Figure 20, where (a) refers to medium and (b) to high demand. The case of 0% penetration rate (square) corresponds to typical PC application without any MP control (for comparison), while the remaining cases refer to combined control of PC and MP in different node penetration rates. Again, boxplots aggregate results of 10 scenarios of random MP node selection of the corresponding penetration rate, combined with the same PC scheme. Triangle annotation refers to the combined scheme where MP nodes are selected according to the proposed methodology, while the case of 100% (dot) refers to the combined scheme with MP installed in all nodes. Selection of MP nodes is based on FTC results and is done with the same parameter values α, β, γ as in single MP, for every demand scenario respectively, while PI controller parameters are the same for all cases of the same demand scenario.

For medium demand, we observe that single PC does not lead to considerable improvement with respect to FTC (only 0.5%). This is not surprising, since there is not enough demand to drive the network to heavily congested states, where PC would get activated for longer and would have a higher impact, and production MFD does not drop significantly for FTC case, as we will see also in figure 21 below. However, in all combined scheme cases, we observe significant travel time improvement with respect to the FTC case. Especially in the cases of 5% and 20%

Table 3: Total travel time for all control scenarios, for medium and high demand, in vehicle hours traveled (VHT). Under ΔVHT, the percentile change of VHT with respect to FTC case is shown. For random MP node selection, the median VHT of 10 random node set replications is reported.

	Medium demand				High demand			
MP node selection	Targeted		Random		Targeted		Random	
Control scenario	VHT	ΔVHT(%)	VHT	ΔVHT(%)	VHT	ΔVHT(%)	VHT	ΔVHT(%)
FTC	221400	-	-	-	484000	-	-	-
MP 5%	195580	-11.7	212743	-3.9	464390	-4.1	472394	-2.4
MP 10%	189250	-14.5	207173	-6.4	471680	-2.5	479732	-0.9
MP 15%	187860	-15.1	204558	-7.6	447360	-7.6	476813	-1.5
MP 20%	185940	-16.0	207777	-6.2	465280	-3.9	476145	-1.6
MP 25%	179850	-18.8	209726	-5.3	467480	-3.4	480113	-0.8
MP 100%	197960	-10.6	-	-	483000	-0.2	-	-
PC	220380	-0.5	-	-	447000	-7.6	-	-
PC + MP 5%	183000	-17.3	214725	-3.0	438540	-9.4	445948	-7.9
PC + MP 10%	190070	-14.2	212544	-4.0	441060	-8.9	451741	-6.7
PC + MP 15%	196190	-11.4	204242	-7.8	422550	-12.7	442860	-8.5
PC + MP 20%	182230	-17.7	205216	-7.3	411510	-15.0	440004	-9.1
PC + MP 25%	184650	-16.6	207452	-6.3	408430	-15.6	441069	-8.9
PC + MP 100%	183220	-17.2	-	-	405300	-16.3	-	-

MP nodes selected by the proposed method, performance is slightly higher than the best performing single MP scheme of the respective penetration rate. Moreover, similar to the single MP cases, we observe that the proposed node selection algorithm leads to higher performance gains compared to random selection for most cases. The highest improvement for the two-layer controller, which is about 17.7% , is recorded for the case of 20% MP nodes, while both 100% and 5% penetration rates achieve about 17.2%. Therefore, in multiple cases, as in 5%, 20% and 100%, adding PC on top of MP increases network performance from 3% to 7% with respect to single MP.

For high demand, results of the combined scheme are more promising than single MP, as we can see in Figure 20(b). While single MP application, as well as single PC, only improve traffic performance by around 8% compared to FTC in the best case, the two-layer framework manages to improve up to 15%, in the case of 25% MP nodes selected by the proposed algorithm, and up to 17% in the case of full-network MP implementation. Therefore, in high demand scenario, PC strategy can be significantly enhanced by the additional distributed MP layer, even with only a fraction of properly selected, controlled MP nodes, which leads to performance very similar to the one of 100% controlled nodes, but with only 25% of the respective cost. Detailed performance of all evaluated controlled schemes can be found in Table 3.

Figure 21 depicts simulation results of the benchmark case of FTC, single MP case for all eligible nodes ('MP 100%'), single MP controlling 20% of nodes selected according to the proposed method ('MP 20%'), and the combined PC with MP to 20% of selected nodes ('PC+MP 20%'), all for the case of medium demand. In 21(a)

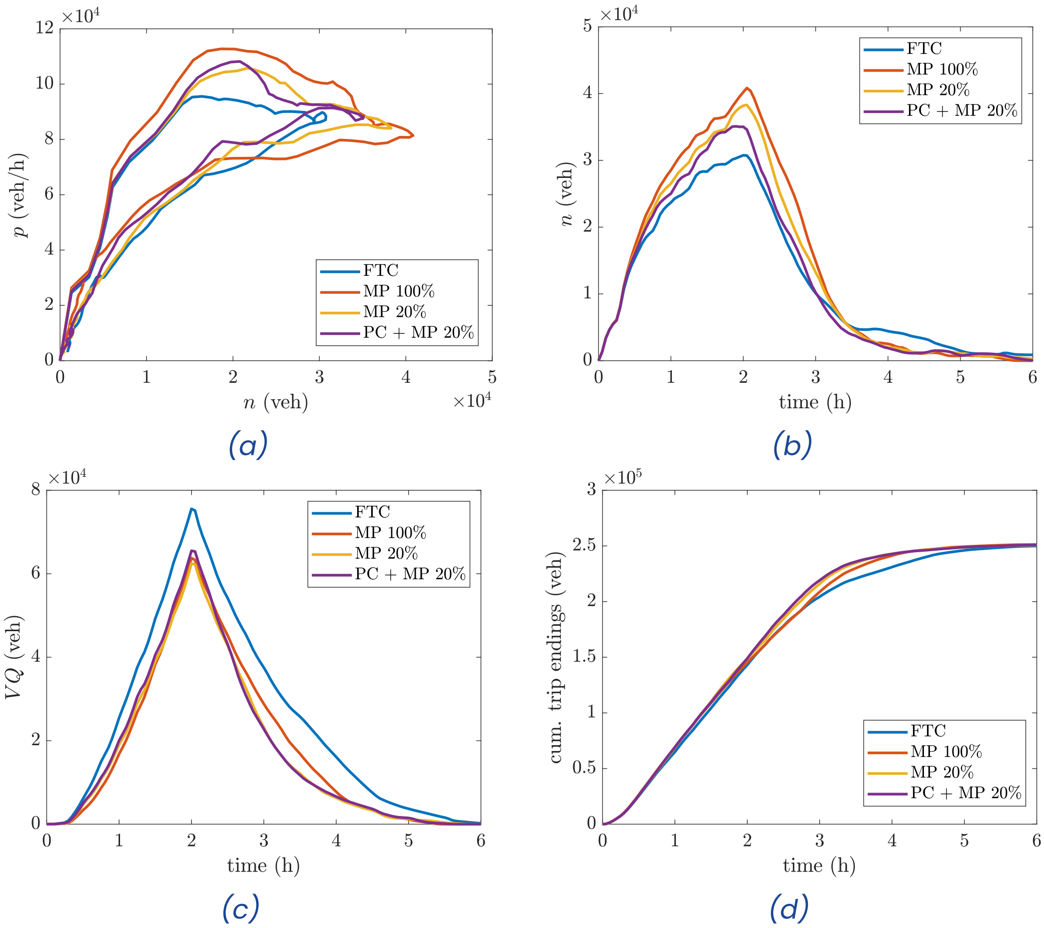


Figure 21: Simulation results for medium demand scenario. Comparison between FTC, MP to all network nodes (100%), MP to only 20% nodes selected by the proposed method, and combined PC with MP to 20% selected nodes. Figures refer to the entire network. (a) MFD of accumulation vs. production; (b) time-series of accumulation; (c) time-series of total virtual queue; (d) time-series of cumulative trip endings.

MFDs of accumulation versus production are shown for the four cases. We notice that all scenarios involving MP significantly increase the maximum production, compared to the FTC case, and therefore, increase both critical and maximum observed vehicle accumulation. This remark indicates that MP strategy can increase system serving capacity in conditions of moderate congestion, and by balancing queues around controlled nodes, it leads to better road space utilization. As a result it allows a higher number of vehicles to be in the system at the same time, which was not possible in FTC due to local gridlocks that were forcing excess demand to stay in virtual queues. This is evident in (b), where total network accumulation of all scenarios is higher than in FTC, as well as in (c), where total virtual queues are remarkably lower. Moreover, by comparing 'MP 20%' and 'PC+MP 20%' MFD curves in (a), we see that the latter leads to slightly lower maximum accumulation and, thus, smaller capacity drop and hysteresis loop in the unloading part. In this case, combined PC+MP performs slightly better than MP by approximately 2%. However, the opposite is observed in the respective cases of 25% MP nodes, which is probably due to traffic correlation among additional MP controlled nodes. We should note here that some change of MFD curve in presence of MP is to be expected, especially with respect to critical accumulation, and this should be considered in the process of parameter tuning for PC. Another interesting remark is that, between the two single MP scenarios, 'MP 100%' results in higher increase in system serving capacity, but on the other side, it introduces more vehicles in the network and thus, reaches higher congestion levels and capacity drop in peak time than FTC. Or, production rises significantly, but also drops during peak, causing some delays and heterogeneity non-existent in FTC, as shown by the unloading part of the MFD curves. This effect can explain why MP installed in all nodes performs worse than partial installation to 20% of nodes, and is closely related to the shape of MFD and how fast production drops when network enters the congested regime. However, in this case, minimum recorded production is not much lower than the one in FTC, while capacity increase is significant, thus, despite the importance of the production drop in MP 100% case, VHT savings compared to FTC are still significant. This effect is less intense in the 'MP 20%' and 'PC+MP 20%', where a slightly lower maximum production is recorded, but lower maximum congestion and capacity drop are observed as well. For medium demand, the highest delay savings are achieved for the case of PC + MP 25%.

Similarly, figure 22 shows simulation results of four, best-performing control scenarios, for the high demand scenario. FTC case is compared to the case of single MP with full-network control ('MP 100%'), the case of single PC, and the case of combined PC with distributed MP in subset of 25% of eligible nodes, selected according to the proposed method. Regarding single MP scheme, a behavior similar to medium demand case is also observed for the high demand, although in the latter, capacity increase is relatively smaller compared to FTC (about 6.5%), while production drop in peak time is significantly higher (around 27%), which can be due to reaching more congested state by allowing more vehicles inside the network at the same time, as we can see in 22b. Overall, MP 100% case performs almost similar to FTC in terms of VHT but significant differences are observed between MFD curves. However, smaller hysteresis is recorder during network unloading in the case of MP 100%, thus reducing the damage made by the production drop. Interestingly this effect is eliminated in the case of the two-layer framework, where PC plays an important role in prohibiting the system from reaching highly congested states. Therefore, in the combined case of 'PC + MP 25%', the network reaches slightly higher production in peak period compared to single PC case, which drops with a smaller rate as accumulation increases above critical, due to MP control. Also, PC impedes the excessive increase of vehicle accumulation in the system and prevents highly hysteretic behavior due to heterogeneity. Among the four cases shown, the combined framework leads to shorter total travel time, reduced by almost 15% with respect to FTC, when single PC achieves a decrease of around 7.5%. In short, adding

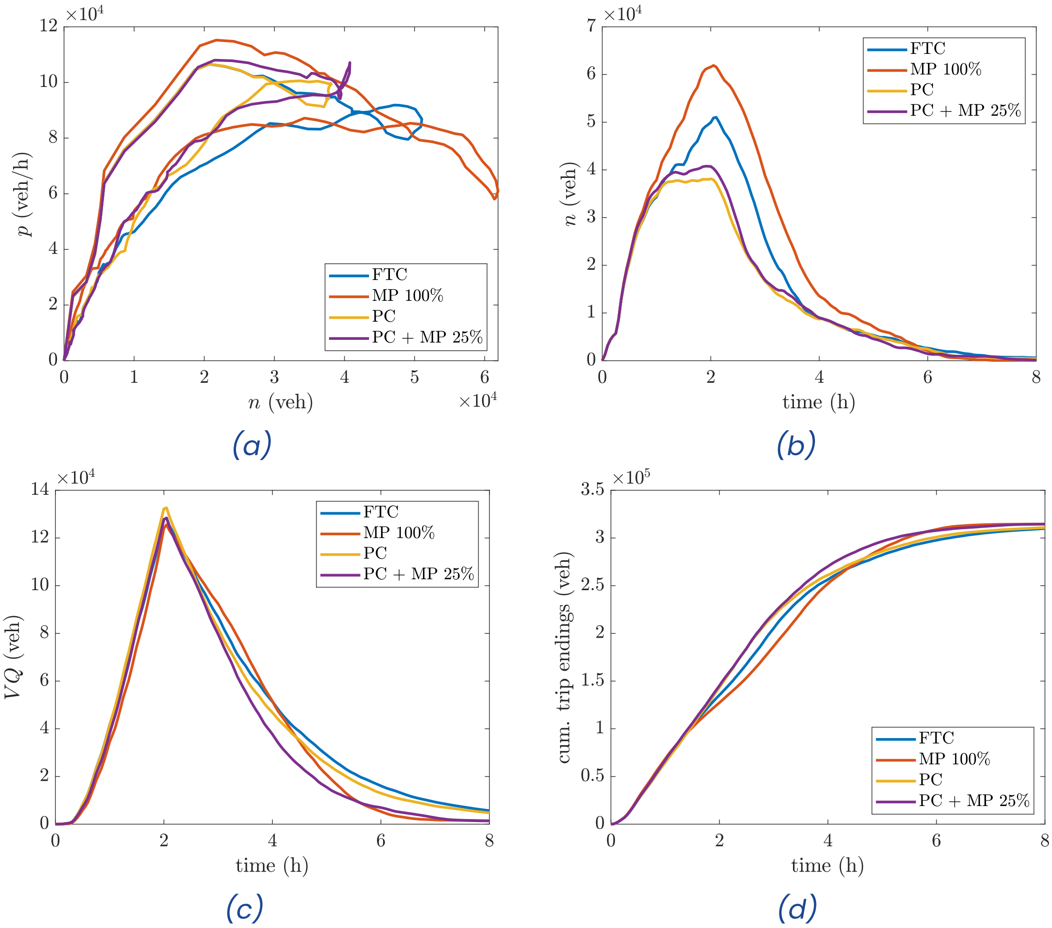


Figure 22: Simulation results for high demand scenario. Comparison between FTC, single PC, single MP in 100% of nodes and combined PC with MP in 25% of nodes selected by the proposed method. Figures refer to the entire network: (a) MFD of accumulation vs. production; (b) time-series of accumulation; (c) time-series of total virtual queue; (d) time-series of cumulative trip endings

a MP layer significantly improves single PC performance, while properly selected MP nodes allow for a smaller network penetration rate that leads to comparable performance as in full-network MP implementation.

4.8 Findings

This section refers to traffic-responsive signal control for urban networks and proposes a two-layer hierarchical control framework, which combines perimeter control, implemented after partitioning of the network in homogeneously congested regions, and distributed Max Pressure control, implemented to isolated network intersections. The key points of this research are the following:

- Combined implementation of multi-region PC with distributed MP in a two-layer hierarchical control framework is proposed for large-scale network control.

- Partial implementation of MP control in subsets of network nodes is tested.
- An algorithm to help identify critical nodes for MP control according to queue-related metrics around the node (mean, variance and spill-backs of queues during peak hour) is developed and tested.
- Several control layouts for MP involving different penetration rates of controlled nodes, the selection of which was both targeted and random (for comparison) are tested by an enhanced version of link-based dynamic macroscopic SaF model, which integrates traffic signals, queue capacities and spill-backs.

More specifically, the proposed node selection method proves effective in identifying critical node sets for MP control, since it outperforms random selection for all network penetration rates in the medium demand scenario. Even though its effectiveness seems to drop in the high demand scenario for the single MP scheme, it remains effective in all combined schemes of PC and MP. The proposed selection variables (m_1, m_2 and N_c) seem to play an important role as indicators of node importance with respect to MP, while further research can help unravel the mechanism that relates selection variable importance to demand patterns, and thus determine optimal parameter values in a universal way by dropping parameter optimization requirement. Overall, it seems that significant correlation exists between controlled nodes, which affect each other in a way not necessarily beneficial for the system, since performance gains can decrease for penetration rates above 25% and can even drop to zero in 100% in highly congested scenarios. This phenomenon highlights the importance of partial MP implementation, especially in increased congestion, and the role of spatial distribution of the controlled nodes, which the proposed selection method tries to unravel.

Regarding the two-layer combined scheme of PC and partial MP, results are promising in most tested cases, especially in the high demand scenario, where in our case study, adding MP in only 25% of properly selected network nodes leads to doubling the performance gains of single PC compared to FTC case, from 7.5% to more than 15%. Moreover, almost the same performance gain is achieved in the case of full-network MP implementation, proving the proposed selection method effective and, as a result, reducing implementation cost to one fourth, compared to full-network scheme. Furthermore, PC protects high-demand regions from reaching saturated states, and therefore from capacity drop, which seems to also increase MP efficiency, given that single MP shows zero improvement for full network implementation in high demand scenario, while combined MP with PC achieves twice the gain of single PC.

5 Multi-modal control

Public transport is a vital component of modern society and plays a crucial role by providing a convenient and affordable mode of transportation. It helps in reducing traffic congestion and air pollution in cities, promoting sustainable living, and improving mobility. Although a lot of effort has been placed on the design problem by transportation engineers over the years, the smooth and reliable operation of public transport is a great challenge especially in congested urban road networks. Thus, transit prioritisation and control schemes are of utmost importance and in many cases essential in order to ensure the reliability and smooth operation of public transport. Such schemes are usually focused on reducing bus travel time between stops through prioritising buses over vehicles [21]–[23]. However, this is often done at the expense of the overall passenger throughput of intersections and corridors. These approaches are effective only if supported by major modal shifts towards public transport. When that is not the case, these strategies can deteriorate the performance of the vehicular traffic and reduce the overall level of service of the infrastructure [24], [25].

5.1 AI assisted Dynamic Bus Lane Control in Connected Urban Environments

5.1.1 Introduction

The advent of the connected era and by extension the advent of connected vehicles create new opportunities in terms of transit prioritisation strategies, as accurate information can be extracted on bus delays and traffic network conditions. Moreover, through advanced communication protocols (V2X) connected vehicles can be instructed to alter their behaviour (e.g. lane change) in a desired manner.

In this section, two learnable, highly scalable and transferable bus prioritisation strategies are developed by NTUA, using reinforcement learning under a connected environment in urban corridors whose aim is to enhance the reliability and efficiency of bus operation. The first strategy is called "Intermittent Dynamic Bus Lanes" (IDBL) and extends in a dynamic and adaptive manner the concept of Dynamic Bus Lanes as proposed by [25], [26]. The second strategy, inspired by [27], is called "Adaptive Bus Lane Density Control" (BLDC). It is essentially a control strategy that refines the design of dynamic bus lanes, compared to the first one, by trying to balance between efficient bus operation and unaffected accommodation of vehicle flows by implementing a mechanism for controlling the bus lane density.

In both cases, it is assumed that the demand for public transport at any given time is reflected on the bus line frequencies given by the operator's timetables and, there-

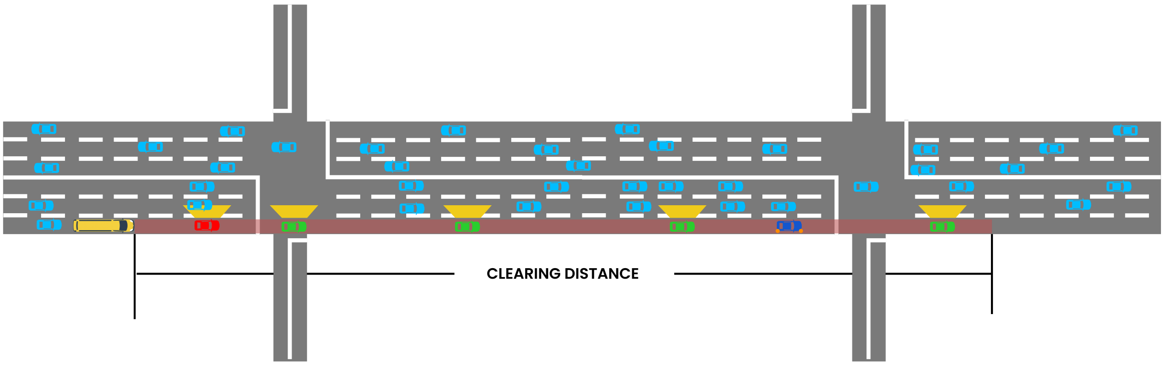


Figure 23: Lane evacuation strategy on a typical corridor. The dynamic bus lane segment is shown in red colour.

fore, the main focus is on making buses more punctual and reliable in the least invasive way rather than always making buses faster assuming greater public transport demand. This approach can be beneficial in facilitating gradual modal shifts and gives a more flexible foundation for applying modal shift policies.

5.1.2 Intermittent Dynamic Bus Lanes

Problem setup

In the first proposed bus prioritisation strategy, called "Intermittent Dynamic Bus Lane" (IDBL), buses are agents that are able to control the activation or deactivation of bus lanes in the front area of the bus in order to mitigate transit delays. At each bus stop, the agent essentially decides whether the downstream lane portion of a given clearing distance (various values are tested) should be vacated by all vehicles following an efficient evacuation strategy based on the reported deviation from the bus schedule and the observed traffic conditions until the next bus stop. The IDBL strategy can be described in detail as follows. When a bus-agent arrives at a bus stop, he observes the arrival delay (or early arrival) at the current bus stop and information about the prevailing traffic conditions in the corridor segment between the current bus stop and the next bus stop. Based on this observation, the bus-agent decides if the bus lane will be activated or not. If the agent decides to activate the bus lane, a process similar to that of 2.2.1 is being followed. All relevant vehicles are informed continuously that the bus lane has been activated to ensure that vehicles in the adjacent lanes will not enter the activated bus lane within the given clearing distance. This ensures that vehicles in the adjacent lanes will not enter the activated bus lane within the given clearing distance. Vehicles within the dynamic lane segment check their adjacent lane for neighbouring vehicles. If a gap of sufficient length between the leading and following vehicles is observed in the adjacent lane, they get instructed to change lane and exit the activated bus lane (for example the green vehicles of Figure 23). Otherwise, vehicles continue straight until a sufficient gap is found.

It should be noted that drivers are assumed to be fully compliant to the given instructions. This means that drivers outside the activated bus lane will never enter and drivers inside the bus lane will always seek to change lanes when requested, but conduct a lane change only when conditions are suitable for a safe lane change, i.e. they operate as they would normally do. Right turning vehicles within a specified distance (80 metres) from their relevant junction are excluded from the lane changing instruction (Figure 24). Moreover, if the lane area upstream of the bus-agent is not activated as a bus lane by another following bus, all vehicles are allowed to enter behind the bus.

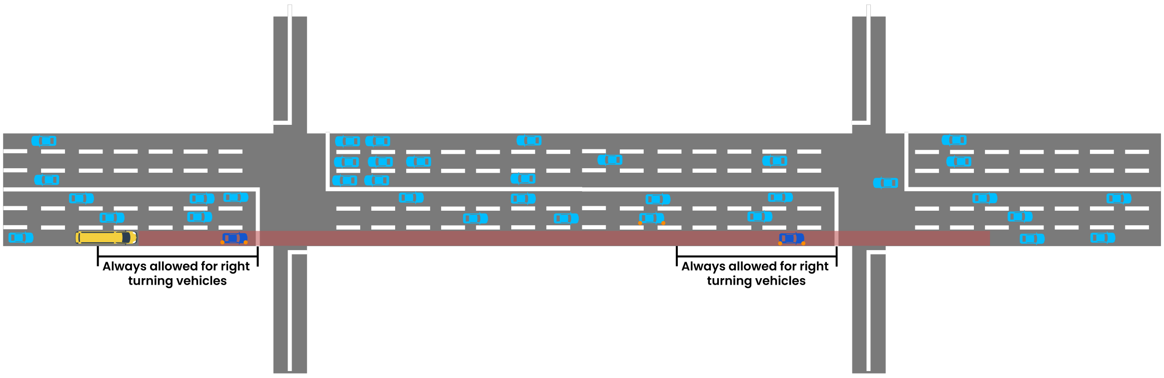


Figure 24: Configuration of Intermittent Dynamic Bus Lane (red stripe). Right turning vehicles are allowed always on the denoted section within the dynamic lane segment.

Finally, when the dynamic bus lane is deactivated, vehicles in the adjacent lanes and vehicles on the (possibly) previously activated bus lane are informed that they are allowed again to enter or stay on the bus lane, respectively, and no other restrictions take place.

A Deep Reinforcement Learning Bus Lane Controller

As mentioned, in the proposed approach a bus-agent is considered to move along a corridor with multiple bus stops. A learnable controller is developed to activate or deactivate the dynamic bus lane whenever the bus reaches a bus stop based on the existing traffic conditions and the reported deviation from a predefined timetable. For the control problem, the principles of Deep Reinforcement Learning are leveraged. More specifically, a Proximal Policy Optimization algorithm (PPO) [5] is trained to control the activation or deactivation of the dynamic bus lane everytime a bus reaches a bus stop. Proximal Policy Optimization is a state-of-the-art on-policy model-free reinforcement learning algorithm that belongs in the policy-gradient methods. It aims to find a balance between exploration and exploitation in order to improve the training stability by updating the policy in a local, proximal manner, avoiding "too severe" policy updates.

The structural components of the agent, namely the state, action and reward, are defined as follows: The bus-agent observes a state s everytime it reaches a bus stop. The state s is defined by a vector comprising of the instantaneous lane occupancy (the total length of vehicles including minimum gap distance on a lane segment divided by the segment's length) downstream of the bus stop until the next bus stop o_{bl}, the instantaneous lane occupancy of the corresponding adjacent lane segment o_{adj}, the mean travel speed ($5\ min$ aggregation period) of vehicles on the same lane segment $speed_{bl}$, the mean travel speed ($5\ min$ aggregation period) of vehicles on the corresponding adjacent lane segment $speed_{adj}$, the arrival delay on the current bus stop of the bus $delay_{bs}$ and the previously taken action a_{prev}. Additionally, for each traffic signal (2 in total) between the current and the next bus stop the agent also observes regarding traffic signal's upstream rightmost lane, the number of waiting vehicles $n_{bl}^{tl_i}$, $i = 1, 2$, the total waiting time of vehicles $W_{bl}^{tl_i}$ $i = 1, 2$, and the instantaneous lane occupancy $o_{bl}^{tl_i}$, $i = 1, 2$. The same values are computed for all corresponding adjacent lanes, i.e., the number of waiting vehicles $n_{adj}^{tl_i}$, $i = 1, 2$, the total waiting time of vehicles $W_{adj}^{tl_i}$, $i = 1, 2$, and the instantaneous lane occupancy $o_{adj}^{tl_i}$, $i = 1, 2$.

When the bus-agent reaches a bus stop based on the observed state s , it takes on a binary action $a \in \{0, 1\}$. If $a = 0$ the dynamic bus lane is activated, otherwise ($a = 1$) is deactivated.

The reward is the key component that drives the agent into learning how to take the optimal action since the agent seeks to obtain a policy that maximises the sum of future (discounted) rewards. The goal of the bus-agent is to minimise the bus stop arrival time deviation from a given timetable. To that end, the reward r is defined as:

$$r = -|t_{ar} - t_s| \tag{31}$$

where t_{ar} is the arrival time at the next bus stop that the agent experienced, t_s is the scheduled arrival time at the next bus stop given by the predefined timetable.

5.1.3 Adaptive Bus Lane Density control

Adaptive Bus Lane Density Control (BLDC) strategy follows the same general rationale as IDBL but with three key differences (Figure 25). First, the bus-agent has control over the number of vehicles allowed within the dynamic bus lane. Instead of simply allowing or banning all vehicles, the agent is more flexible and is able to adjust the bus lane density to find the optimal balance between ensuring reliable transit schedules and accommodating vehicular flows. Second, the agent decides and acts when reaching a certain distance (100 metres) from the bus stop. This provides the agent with crucial additional time to regulate the density of the dynamic

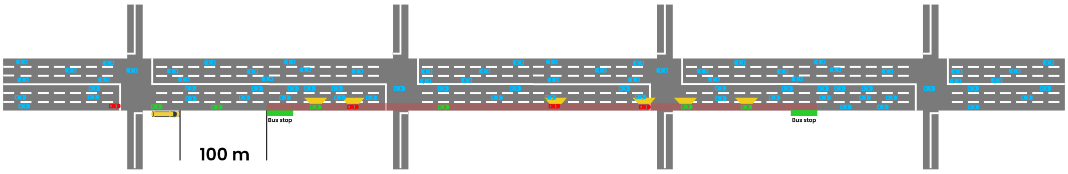


Figure 25: BLDC's configuration

lane segment starting in front of the upcoming bus stop. Third, the length of the bus lane changes dynamically with its initial length being equal to the distance between two consecutive bus stops. This means that as the bus passes one bus stop towards the next one, the bus lane length is limited from the position of the bus on the bus lane until the next bus stop. Similar to IDBL, vehicles are allowed to enter the upstream lane of the bus if it is not activated by another following bus. Based on the above configuration the BLDC method may lead - at least conceptually - to a more precise and stable control of the selected density.

The problem of which vehicles will be instructed to exit the bus lane arises, since BLDC doesn't evacuate all vehicles in the dynamic bus lane segment. Moreover, it is crucial for the effectiveness of the strategy that the density of the bus lane stabilises to the desired one as soon as possible. It is therefore important that the instruction to exit is given to vehicles for which the possibility of successful lane change is maximised. The same holds for vehicles on the adjacent lane that will be informed that are allowed to enter the bus lane when the prevailing density is less to the one decided by the bus agent. Finally, it would be also desirable that the instructed to exit vehicles as well as the informed ones that are allowed to enter will be those that cause the least possible disturbance to the rest of the traffic. To that end, a hierarchy of vehicles in terms of the lane change ease was created using the lane change priority p_{lc} score based on Time-To-Collision (TTC) metric [28] defined as follows:

$$p_{lc} = TTC_f + TTC_l \tag{32}$$

Where TTC_f is the longitudinal TTC of the vehicle with the closest following vehicle in its relevant adjacent lane, TTC_l is the longitudinal TTC of the vehicle with the closest leading vehicle in its relevant adjacent lane.

The bus lane density is monitored every 1 second. Depending on whether the observed density is less or greater than the one decided by the bus-agent, the BLDC method makes use of the TTC-based score and gives exit-instructions or enter-notifications, respectively, in descending order (Figure 26 and 27).

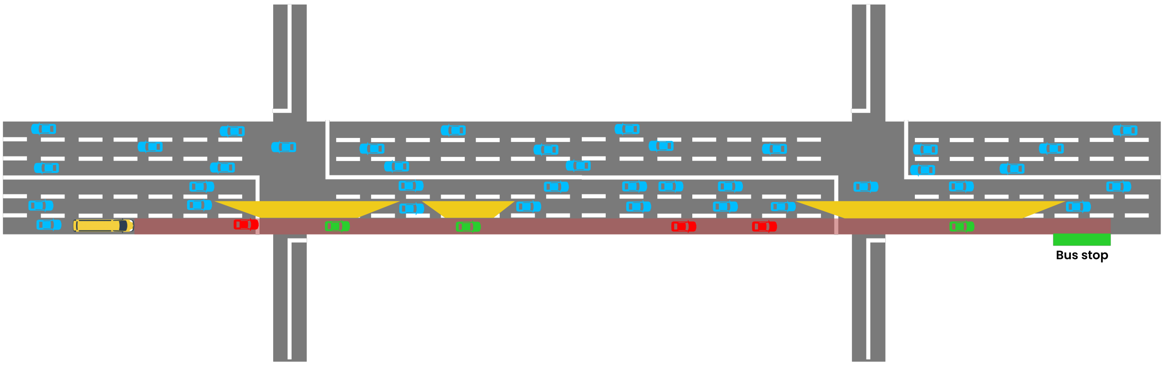


Figure 26: Example of a BLDC's exit-instruction strategy for 3 exiting vehicles. In green are shown the vehicles with the highest priority score that receive an exit-instruction. Vehicles with lower priority scores are shown in red. Yellow areas illustrate the space dimension of the priority score.

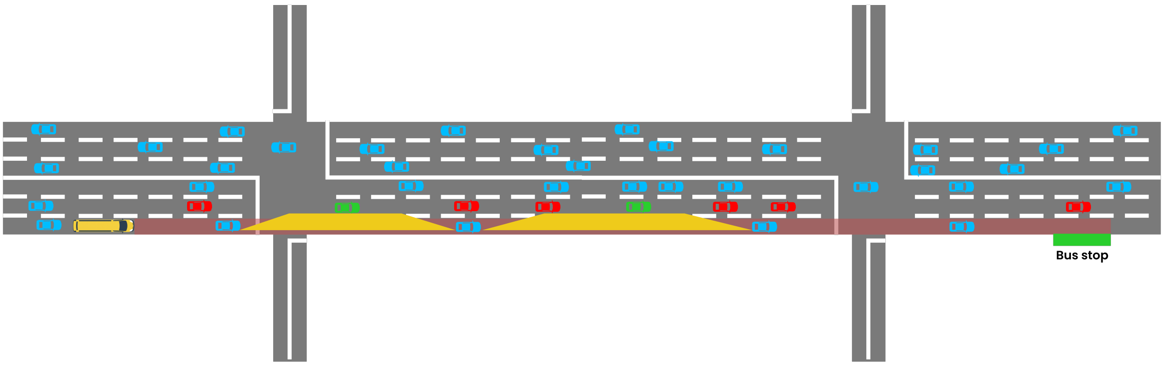


Figure 27: BLDC's enter-notification strategy. Again in green, vehicles with the highest priority score that receive an enter-notification are shown. Yellow areas represent the space dimension of the priority scores.

Drivers are assumed to be fully compliant to the given instructions, as in IDBL, in a way that always seek to change lane when instructed while vehicles that don't receive an enter-notification will never enter the bus lane. Finally, it should be noted that the enter-notification is not mandatory, meaning that drivers are only expected to enter the bus lane if it's for their own benefit.

Deep Reinforcement Density controller

Similar to IDBL, the bus is a reinforcement learning agent which decides the desired bus lane density between two consecutive bus stops based on the prevailing traffic conditions and an estimated deviation from a predefined timetable. The

Soft Actor-Critic (SAC) [29] algorithm is selected for the learning process. SAC is a model-free, offline reinforcement learning algorithm that uses a combination of an actor network, which is responsible for selecting actions, and a critic network, which is responsible for evaluating the quality of the actions selected by the actor. The PPO algorithm (used in IDBL) was also tested however it produced less favourable results during the training phase and is therefore disregarded.

The structural components of the agent, namely the state, action and reward, are defined as follows. Everytime the bus-agent is within 100 metres from the upcoming bus stop observes a state s. The state s consists of the downstream instantaneous lane occupancy (as was defined in IDBL method) from the current bus stop until the next bus stop o_{bl}, the instantaneous lane occupancy on the corresponding adjacent lane segment o_{adj}, the mean travel speed ($5\ min$ aggregation period) of vehicles on the dynamic bus lane $speed_{bl}$ the mean travel speed ($5\ min$ aggregation period) of vehicles on the corresponding adjacent lane segment $speed_{adj}$, the estimated arrival delay to the upcoming bus stop of the bus $delay_{bs}^{est}$, the previously taken action a_{prev} and the successful lane changes, lc_{suc}^{prev}, as a result of the given exit-instructions based on the previous action of the agent. Moreover, for each traffic signal (2 in total) between the current and the next bus stop the agent also observes regarding traffic signal's upstream rightmost lane, the number of waiting vehicles $n_{bl}^{tl_i}$, $i = 1, 2$, the total waiting time of vehicles $W_{bl}^{tl_i}$ $i = 1, 2$, and the instantaneous lane occupancy $o_{bl}^{tl_i}$, $i = 1, 2$. The same values are computed for all corresponding adjacent lanes, i.e., the number of waiting vehicles $n_{adj}^{tl_i}$, $i = 1, 2$, the total waiting time of vehicles $W_{adj}^{tl_i}$, $i = 1, 2$, and the instantaneous lane occupancy $o_{adj}^{tl_i}$, $i = 1, 2$.

When the bus-agent is within 100 metres from the upcoming bus stop based on the observed state s takes action a. The action $a \in [0, 1]$ and corresponds to the desired percentage of bus lane coverage by vehicles, therefore, the action space A is continuous, $A = [0, 1]$. The generated value is transformed to the desired density of the bus lane and then the action is applied.

As described above, BLDC method objective is to find the optimal balance between ensuring reliable public transportation and accommodating vehicular flows. To that end, the goal of the bus-agent is to minimise the bus stop arrival time deviation from a given timetable while simultaneously minimise the disturbance to vehicular traffic due to the emerging lane changing behaviour. Thus, the reward r is defined as:

$$r = -\alpha|t_{ar}^{est} - t_s| - \beta|lc_{suc}| \tag{33}$$

Where t_{ar}^{est} is bus' estimated arrival time at the next bus stop, t_s is the scheduled arrival time at the next bus stop given by the predefined timetable, lc_{suc} is the number of successful lane changes as a result of the given exit-instructions based on the current action of the agent and finally α, β are the weights for the timetable-related and disturbance-related part of the reward, respectively.

The α and β weights are hyperparameters and the best values were found to be 1.0 and 2.0, respectively.

5.1.4 Implementation Details

Description of the testbed network
SUMO (Simulation of Urban MObility) [6] micro-simulation framework is used to create an artificial 3 km corridor (Figure 28) consisting of 8 4-leg consecutive intersections. The main artery's edges have a length of 350 metres and consist of 3 lanes with the upstream edges having one extra reserved lane for left turn with an 80 metres reservation distance. The secondary roads' edges have a length of 250 metres and consist of 2 lanes. 5 bus stops were positioned along the corridor with 700m distance between them. The traffic signals have a 4-phase program with protected left turns. Moreover, the traffic signal programs were optimised with a fixed 90 seconds cycle in terms of green time allocation and synchronisation (green waves) as this is usually the case in real-life urban corridors.

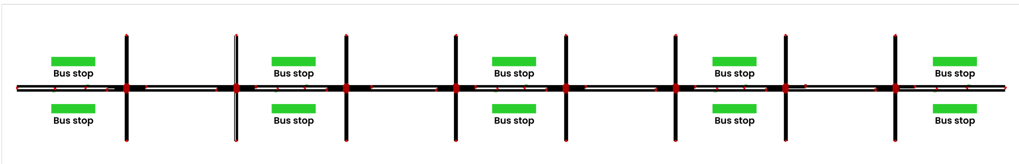


Figure 28: The testbed network in SUMO environment

Traffic Demand Scenarios and Bus Timetable Extraction
Initially, a one-hour baseline traffic demand scenario is created that resembles normal traffic conditions and corresponds to approximately 600 veh/hour per lane for the main artery and 150 veh/hour per lane for the secondary roads. To ensure uniform conditions on the main artery, a constrained convex optimization framework was developed. The constraints concern mainly right and left turns. Furthermore, the uninterrupted flow in the entrances of the testbed network results in unrealistic queue accumulations and for that reason the flow was slightly constrained also to avoid such phenomena. The derivation of the bus timetable was based on the baseline scenario. In more detail, a one hour simulation using the baseline scenario's hourly demand with a warm-up period of 15 minutes was run for 100 times with random seeds. The bus timetable resulted as the mean measured arrival time for each bus stop.

Training Setting
Both described reinforcement learning agents were trained under the same setting. First the task at hand was designed to be episodic and the environment was single-agent. In each episode one bus-agent interacts with the environment, starting from the first bus stop (initial state) and continuing until reaching the final bus stop (terminal state). For increased generalisation three more key steps were added to the training setting. First, the bus in each episode is inserted with a random delay ranging from -2 minutes to +3 minutes. Second, the bus to enter the simulation was chosen randomly from a schedule with 6 buses per hour (per direction). The simulation was run (including warm up period) until the arrival of the bus at the first bus stop for IDBL (or within a distance of 100m for BLDC) and then the episode was initialised. This ensures that the agent experiences diverse traffic conditions and queue accumulations as the traffic evolves. Third, the demand of each episode is generated as a random scale of the baseline scenario demand ranging from 0.5-1.5.

Moreover, a synchronous environment parallelization is performed. Parallel environments in reinforcement learning training can greatly enhance the training process by allowing for more samples to be collected in a shorter amount of time. By training multiple instances of an agent in parallel, each instance can run in its own environment and collect data independently. This can lead to more diverse and comprehensive training data, leading to better generalisation and improved performance of the trained agent. Finally, parallel training can lead to faster convergence and a reduction in training time compared to a single-threaded training approach. This resulted to a training time below 2 hours until convergence.

Evaluation Procedure
Although the training was conducted in a single-agent environment both for IDBL and BLDC method, the evaluation was carried out in a multi-agent one with bus flow continuously entering the simulation environment. More precisely, for each of the developed methods, 2 sets of 11 1-hour simulation scenarios are executed 10 times each (with random seeds) and averaged, that vary with respect to bus frequency (6 and 12 buses per hour per direction on the main artery) and demand scale (0.5-1.5 with step 0.1). The proposed strategies are compared with the most frequent practice regarding bus lane control, i.e. dedicated bus lanes and with mixed traffic conditions (no control).

To compare the aforementioned strategies regarding the compliance with the derived bus timetables, the metric of Mean Absolute Arrival Time Deviation (MAATD) is computed which is defined as:

$$MAATD = \frac{\sum_{i=1}^{n} \sum_{j=1}^{k_i} \left| t_o^{i,j} - t_s^{i,j} \right|}{n \sum_{i=1}^{n} k_i} \tag{34}$$

$$\forall\ bus\ stop\ i,\ \forall\ bus\ j\ that\ reached\ bus\ stop\ i$$

where $t_o^{i,j}$ is the observed arrival time at bus stop i of bus j after the implementation of a strategy and $t_s^{i,j}$ is the arrival time at bus stop i of bus j given by the derived bus timetable. This metric essentially is an aggregation of the observed deviations from the timetable (either ahead or behind schedule) of all the buses that have entered the simulation within the one hour evaluation period. Moreover, the mean time headways between consecutive buses are measured and the mean time headway is extracted. For a fair comparison of the overall impact of each strategy to vehicular traffic, the Total Travel Time and Delay (TTTD) metric is measured and is defined as:

$$TTTDD = TotalTravelTime + TotalDepartDelay \tag{35}$$

Where $TotalTravelTime$ denotes the total travel time of all vehicles inserted in the simulation with the vehicles generated within the warm up period excluded and $TotalDepartDelay$ denotes the total departure delay of the inserted vehicles as well as of the vehicles waiting to be inserted into the simulation. Again, the vehicles generated within the warm up period are excluded.

5.1.5 Findings

Regarding the IDBL method the clearing distance of the dynamic bus lane is a hyperparameter. For that reason, four different clearing distances (100m, 200m, 300m and 400m) were trained and tested.

In Figure 29 the training results for all clearing distances are depicted. All methods have converged and it is obvious that clearing distance equal to 300 metres has performed better concerning training. As is depicted in Figure 30 in terms of Mean Absolute Arrival Time Deviation the best clearing distance on average is 300 metres and this model is used in the comparison with the other methods.

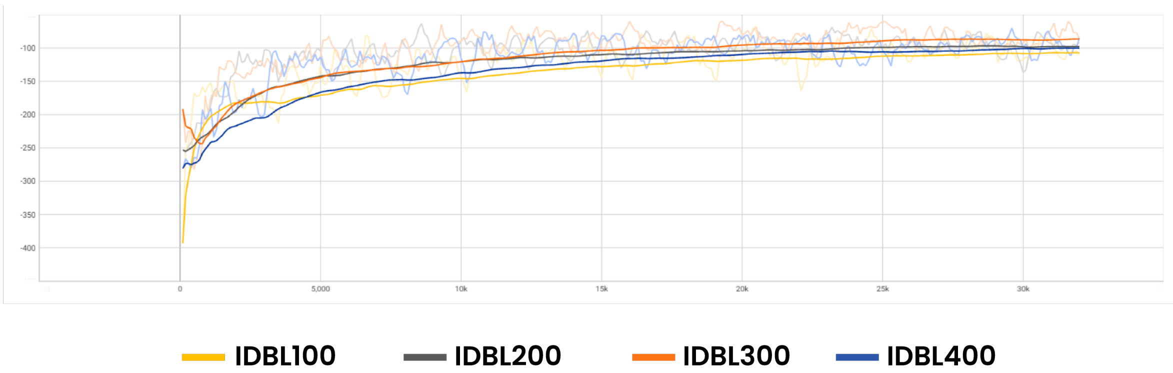


Figure 29: Training results for IDBL for all clearing distances.

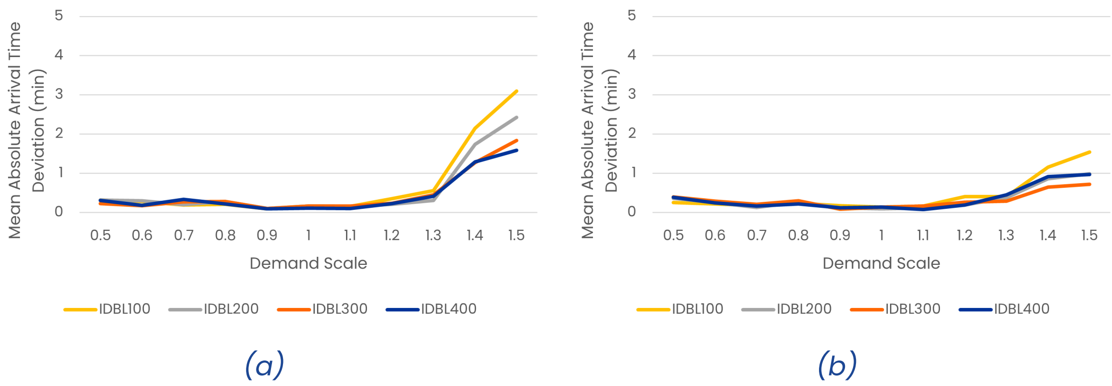
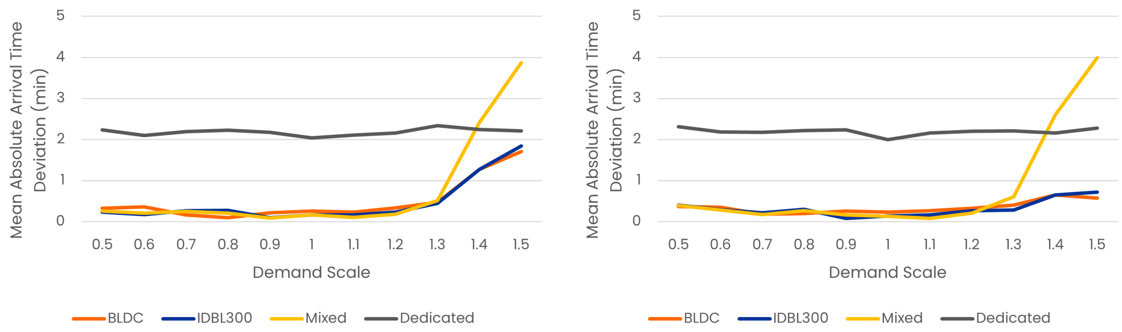


Figure 30: Mean Absolute Arrival Time Deviation of IDBL per clearing distance for 6 buses per hour (a) and 12 buses per hour (b)

To evaluate the compliance with the derived bus timetable, the Mean Absolute Arrival Time Deviation as well as the Average Time Headways are depicted in Figure 31 and 32, respectively. As is expected the dedicated bus lane for demand scale less than 1.4 is the worst method regarding Mean Absolute Arrival Time Deviation for both bus demands. This highlights that this method is suitable only under certain traffic conditions at least for bus prioritisation. The other 3 methods are close with Mean Absolute Arrival Time Deviation below 30s until the demand scale is equal to 1.3. Above that value, mixed traffic exhibits a more severe divergence reaching a four minute Mean Absolute Arrival Time Deviation for 6 bph and 12 bph with the proposed methods staying below 2 minutes for 6 bph and below one minute for 12 bph. Moreover, the IDBL method seems to perform slightly better for higher demand scales compared to the BLDC method. An interesting result is that both methods (BLDC and IDBL) show a better performance for 12 bph compared to 6 bph. This can be explained by the more intense influence between consecutive buses' actions when the time headway is smaller.

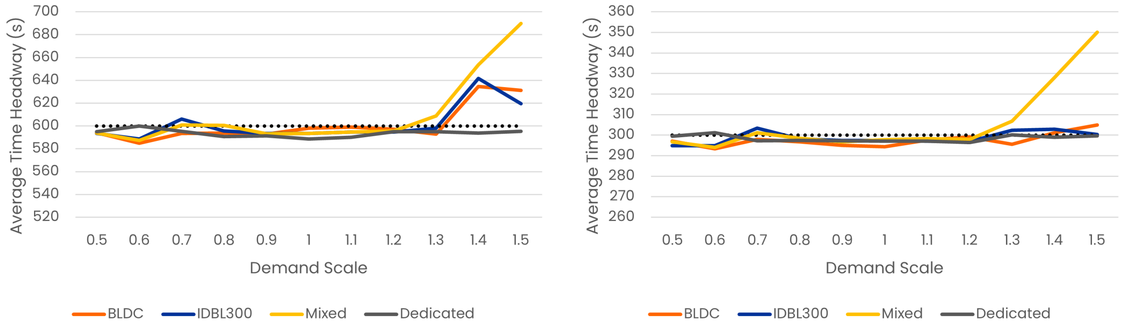
Concerning Average Time Headway the best method is dedicated bus lanes as is expected. Again for mixed traffic and for demand scale above 1.2 the average time headways are steeply increasing (for 6bph and 12 bph). The same holds for the two proposed methods for 6bhp with a milder deviation from the desired time headway for demand scale above 1.3. However, for 12 bhp they exhibit an impressive low deviation from the desired time headways for all tested demand scales.

To evaluate the impact and disturbance of each strategy to vehicular traffic the TTTDD metric of vehicles is depicted in Figure 10. Both the proposed methods are close (slightly worse) to mixed traffic where the vehicular traffic is completely uninterrupted for all demand scales and for both bus demands. Moreover, the impact of BLDC is less compared to IDBL for larger demand scales (above 1.2) and on average is of the order of 6% less. The low vehicular impact of IDBL is due to the small clearing distance although it is less flexible when it comes to bus lane evacuation.



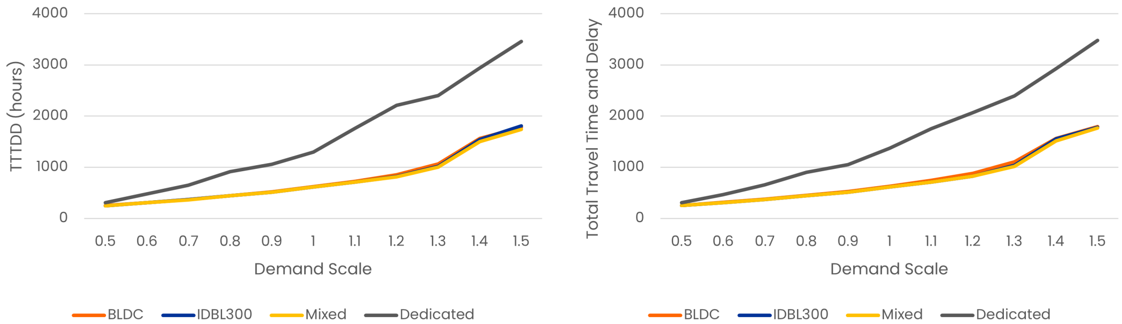
(a) (b)

Figure 31: Mean Absolute Arrival Time Deviation for all methods for 6 buses per hour (a) and 12 buses per hour (b)



(a) (b)

Figure 32: Average Time Headway for all methods for 6 buses per hour (a) and 12 buses per hour (b).



(a) (b)

Figure 33: Total travel time and delay of vehicles for all methods for 6 buses per hour (a) and 12 buses per hour (b).

5.1.6 Concluding remarks

This part introduced two novel approaches for dynamic bus lane control based on Reinforcement Learning, IDBL and BLDC, whose aim is to enhance the reliability and efficiency of bus operation while minimising the impact on vehicular traffic.

Reinforcement Learning gives the ability to bus-agents to learn the correlation between prevailing traffic conditions and expected transit delays. The results' analysis showed that both methods manage to minimise transit delays even in higher demand scenarios while maintaining the disturbance of vehicular traffic close to mixed traffic conditions. The fully connected environment and drivers' full compliance are some strong assumptions that were made. The former was necessary in order to provide both methods with crucial information about traffic conditions on the network but creates some difficulties for the implementation of the proposed methods in the field as is demanding from an infrastructure point of view. The latter is expected to have enhanced the results to some extent as drivers in real life may not follow the given instructions and maybe in some situations exhibit delinquent behaviour. Some next steps for further research, is to conduct a sensitivity analysis of the proposed methods on connected vehicles penetration and drivers' compliance rates with some extensions needed to be made in order to substitute the missing information due to reduced coverage of connected vehicles by using for example induction loops. Another natural direction is to combine both worlds so as to take advantage of each method's benefits and test it in real urban corridors.

5.2 Dynamic Bus Lane control and Perimeter Control

5.2.1 Introduction

This section continues to discuss the concept of Dynamic Bus Lanes (DyBL), as introduced by [27] in the framework of distributed network control, in combination with perimeter control, as proposed by EPFL. A DyBL can be defined as a controlled lane destined primarily for buses, which also allows certain number of private vehicles to use it when traffic conditions allow for it, in the sense that bus priority is guaranteed in cases of high congestion. The idea is to investigate the potential benefits of a dynamically controlled car entry rate in the bus priority lane based on real-time traffic input in network-scale signal control, in parallel with perimeter control, which would ensure public transport priority while making a better use of the road space than conventional Dedicated Bus Lanes (DBL) in cases of uncongested traffic. A DyBL is similar to a DBL in design, with the difference of requiring special signaling equipment (e.g. dedicated traffic lights, pavement lights or variable message signs) to inform drivers of whether cars are allowed to enter or not in real time. A schematic representation of DBL and DyBL is shown in figure 34.

The objectives of this study are to implement the concept of DyBL in combination with perimeter control and evaluate different control algorithms and system architectures in terms of network performance, aiming at minimizing total passenger travel time. A simple dynamic controller is configured for controlling DyBL car

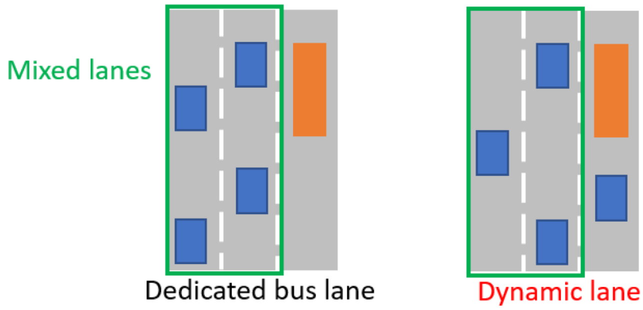


Figure 34: Mixed-use lanes, Dedicated Bus Lane (DBL) and Dynamic Bus Lane (DyBL)

inflow and implemented via microsimulation for a realistic network and demand scenario, in combination with a simple two-region perimeter control.

5.2.2 Methodology

Dynamic control of Bus Lanes

Buses always use the dynamic bus lane while car vehicles can use the mixed-use lane or the dynamic bus lane, on condition that this is allowed. Private vehicles can change lane only at the beginning or the end of each section in the dynamic corridor. While travelling within the corridor, they should continue on the lane they chose when they entered. The dynamic controller for cars shown below, is applied:

$$p_t = p_{t\text{-}1} + K_{\text{mix}}(k_{\text{mix},t} - k_{\text{mix},s}) - K_{DL}(k_{DL,t} - k_{DL,s}) \tag{36}$$

In the above, p_t is the percentage of cars which are allowed to use dynamic lanes at time step t, $k_{\text{mix},t}$ is the average car density over all mixed-use lanes in the corridor at time t, $k_{DL,t}$ is the instantaneous car density on the dynamic lane, $k_{\text{mix},s}$ and $k_{DL,s}$ are the setpoints for the two measures and K_{mix} and K_{DL} are controller coefficients for the two error terms. Note that coefficients K_{mix}, K_{DL} are positive, so when the difference $k_{DL,t} - k_{DL,s}$ is positive, i.e. the density in the dynamic lane is above the setpoint, the second cost term will try to decrease the percentage of cars allowed. Similarly, when the density in mixed-use lanes is above the setpoint, the first cost term will try to increase the percentage of cars in the dynamic lane. The produced car percentage is rounded into 0, 10, 20, ... % before being applied. The renewed percentage is applied only at the end of a control time frame of 200 sec, which is longer than the traffic signal cycle (90 sec). The density setpoints correspond to the critical values as indicated by the fundamental diagram of the studied corridor 3, for the mixed and dynamic lanes.

Perimeter Control

Similar to section 4.2.1, the concept of perimeter control is to adjust traffic signal timing of a set of intersections on the boundaries between protected regions in order to control transfer flows and maintain high flows in the interior of the regions. In this way, highly congested stated are avoided and cars are forced to wait outside the perimeter of the region before being allowed to enter. With the objective of testing DyBL in paraller with PC, we introduce a simple PC regulator with only a proportional term, applied for the green time adjustment of the controlled interregional approaches, as follows.

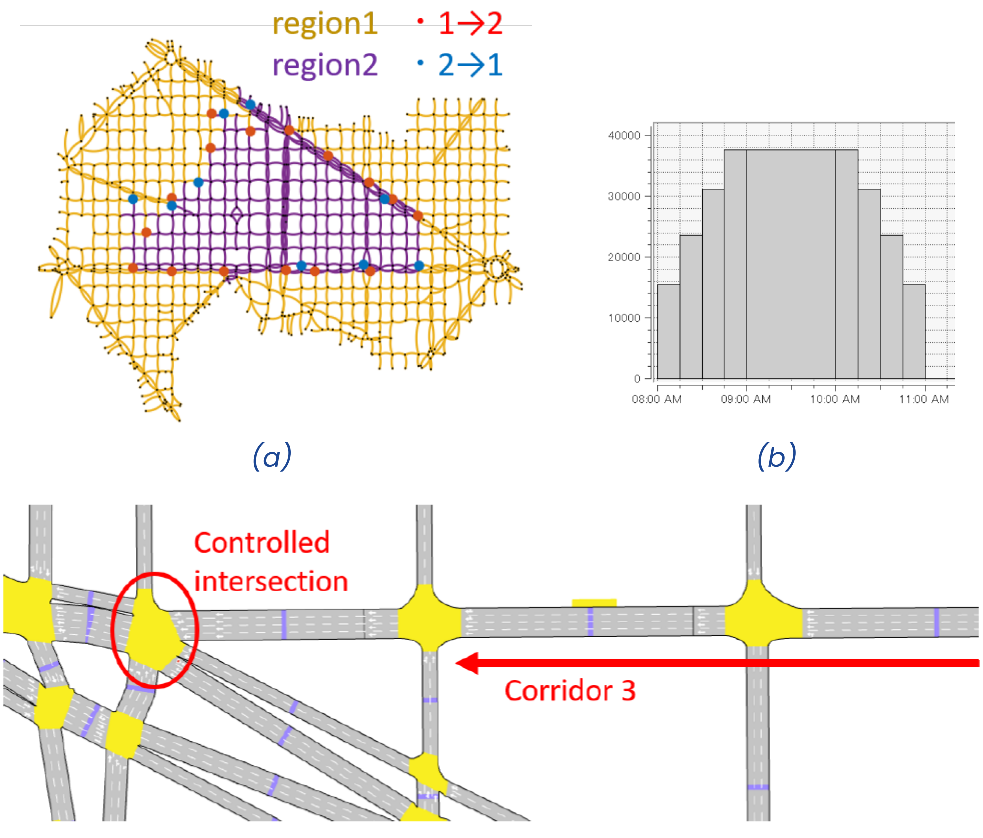
$$\mathbf{G_t} = \mathbf{G_{t-1}} + \mathbf{U}(\mathbf{n}_t - \mathbf{n}_s) \tag{37}$$

where G_t is the vector of green times at time t, $\mathbf{n_t}$ is the vector of regional accumulations, $\mathbf{n}_s$ is the vector of accumulation setpoints and $\mathbf{U}$ is the cost matrix. The setpoints are the critical accumulations according to the observed MFDs. The controller gets activated when accumulation in region 2 (central) overpasses a predefined threshold n_λ. This practice is followed to avoid a control that tries to increase vehicle accumulation in the city center because it is lower than critical in cases of lower congestion. Buses are assumed to be equivalent to 3.9 vehicles when accumulation is calculated. The update values for green time are calculated for each signal group (region 1 to 2 and region 2 to 1) and applied to all traffic signals in the same way.

5.2.3 Case study and implementation

Simulation experiments are performed using microscopic simulator by Aimsun, for the implementation of the proposed dynamic control strategy, combining the control of a dynamic bus lane corridor with neighborhood-scale perimeter control. The utilized network, which makes part of the Barcelona city center as shown in 35(a), consists of 428 links and 158 intersections. There are 112 bus lines and their headway is 8 minutes per line. The free-flow bus speed on all links is set to 15 km/h. The network is partitioned into two homogeneous regions as shown in Figure 35(a) in different colors (1 and 2). The nodes participating in the perimeter control scheme are shown on the network map. The city center is referred to as region 2 and the peripheral area as region 1. The traffic lights which control the flow from region1 to 2 are shown as red dots and the ones from region 2 to 1 as blue dots. The two groups of traffic lights are updated based on the vehicles accumulation in each region and according to an MFD-based perimeter control regulator.

A dynamic bus lane is placed in a central corridor of region 1 of the network with direction towards the city center, which consists of a series of consecutive links. The corridor ends at a perimeter control node where queue is expected to form in peak hour due to PC gating. The simulated car traffic demand is shown in figure 35(b). The car demand consists of 1.5 h of peak traffic, preceded by 45 min of increasing demand and followed by 45 min of decreasing demand. Car occupancy is assumed to be 1 passenger and the total number of car trips is 91640 veh during the entire simulation. Buses follow a fixed frequency timetable with 8 min interval of departure for each line over the 6 hour-period, starting on the same timing as cars. It means 3360 buses run on the network during 3 hours. The mode share is assumed to be 40% bus and 60% car and thus the per bus occupancy is assumed to be 24 pax/bus on average. In terms of implementation within the microscopic simulator, cars that are allowed to use the dynamic bus lanes do indeed use them, in other words the respective amount of car drivers will be "forced" by the simulator to use the dynamic lane. In other words, the indicated by the controller number of cars in the dynamic lane is very close to the simulated one. This approach is adopted in the preliminary phase of our study, while drivers' free choice of whether they want to use the lane can be introduced at a later stage.

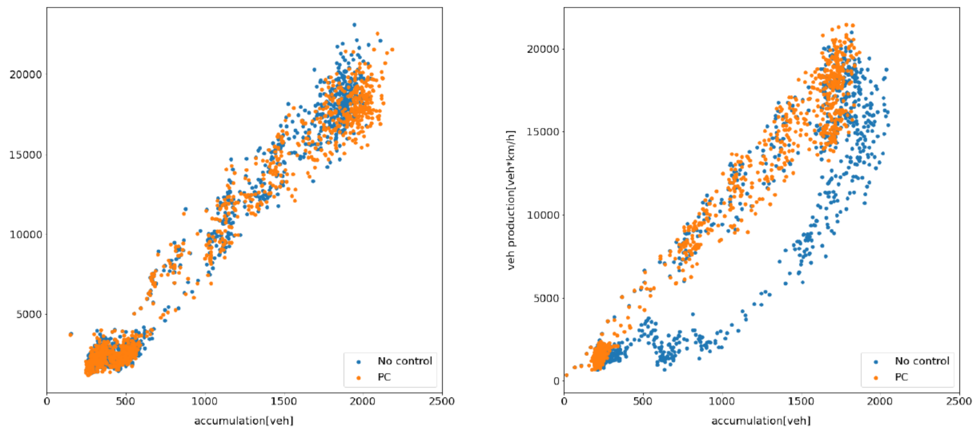


(a) (b) (c)

Figure 35: Case study description. (a) Plan of the studied network of Barcelona city center; (b) Profile of the dynamic demand; (c) Detailed position of the dynamic bus lane corridor, close to perimeter control node.

5.2.4 Results

Figure 36 shows the observed MFDs of the two regions of the studied network, for the utilized demand scenario, in the case of no adaptive signal control (fixed-time control, FTC) and in the case of perimeter control. While region 1 remains uncon-



(a) Region1 (b) Region2

Figure 36: Macroscopic Fundamental Diagrams for regions 1 and 2 in the case of fixed-time control and single perimeter control.

Table 4: PHT difference with respect to the case of no adaptive control (FTC) of all tested scenarios.

	bus (%)	car (%)	total (%)
No Control	0.00 (21203 [pax *h])	0.00 (8157 [pax*h])	0.00 (29282 [pax*h])
single PC	-5.02	-9.23	-5.68
PC + 0% (DBL)	-5.60	-10.2	-6.85
PC + 10%	-5.61	-9.88	-6.81
PC + 20%	-5.57	-9.46	-6.30
PC + (DyBL)	-6.68	-10.3	-6.89

gested in both scenarios, in region 2 a capacity drop and hysteresis loop is observed in FTC which is improved in the PC case.

Figure 37 presents the simulation results of a set of different control scenarios. Boxplots refer to a set of 5 replications with different initial seeds for the stochastic parts of the experiment (e.g. demand generation, path selection etc.) for every case. On the vertical axis, the percentile difference of Passenger Hours Traveled (PHT) for buses, cars and in total are shown with reference to the FTC case (labeled as 'No Control'). Results are presented for the cases of single PC (no dynamic lane), PC with dedicated bus lane (no cars allowed), PC with bus lane with a fixed 10% and 20 % of cars allowed to enter and PC with completely dynamic bus lane, with the controlled described above. We observe that the combined case of PC and DyBL outperforms all other cases, although only slightly for the specific demand scenario. As expected, significant improvement comes as a result of PC, but further improvement seems to relate to the concept of separating cars from buses. More specifically, car travel time seems to decrease when dedicated lanes are implemented, as well as when 10% and 20% of cars are allowed in the lane. The high-

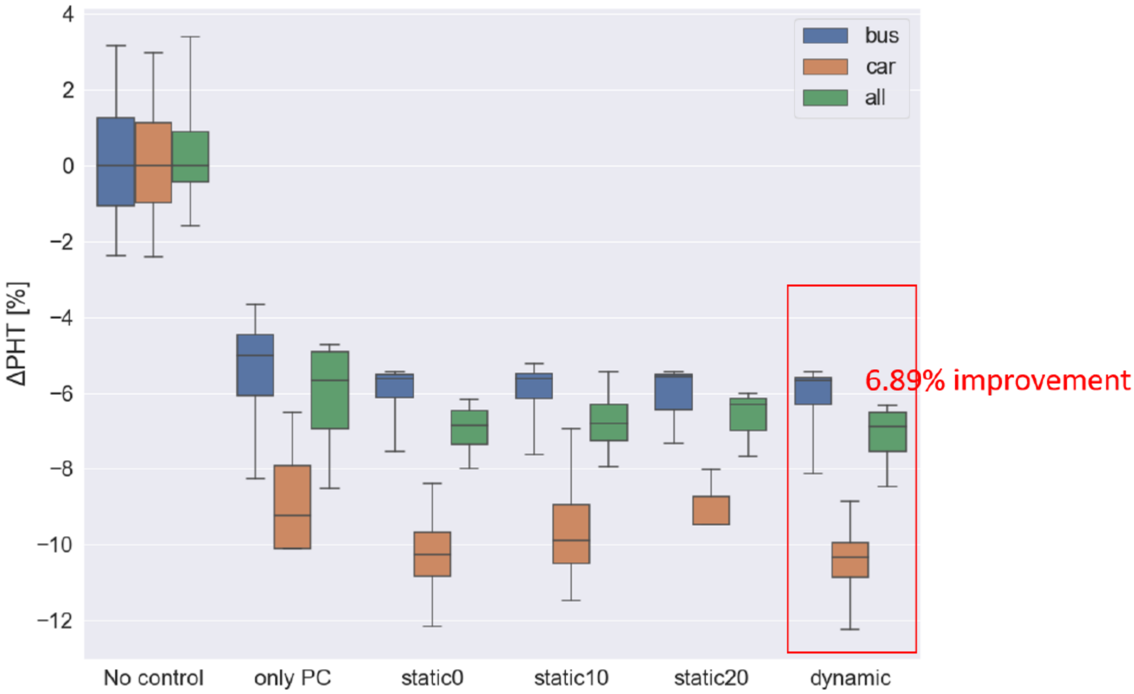
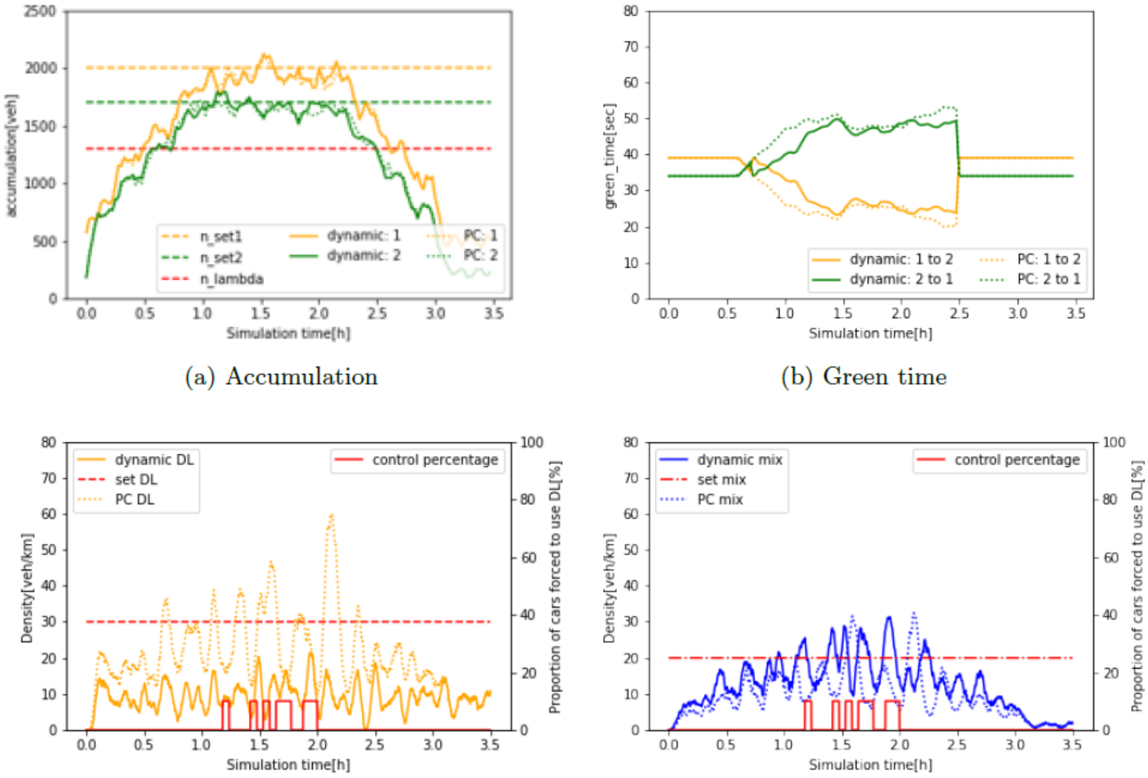


Figure 37: Comparison of Passenger Hour Traveled (PHT) change of all scenarios tested compared to the "no control" case (FTC): Single PC, PC + DBL (0%), PC + bus lane with 10% and 20% fixed allowed car entry rate, and PC + fully dynamic DyBL.

est improvement recorded for the combined case of PC and dynamic control of the bus lane corridor is 6.89% over the total, with a higher improvement for cars and lower for buses, compared to the single PC case. This indicates that the dynamic control of car entry rate in dynamic lanes may be a promising direction in the control of bi-modal networks and further research is required on this topic. Detailed numerical results are listed in Table 4

Figure 38 presents an analysis of one replication of the case of single PC and one of the case of PC combined with dynamic control of bus lane (DyBL). In 38(a) the time-series of regional accumulation in both regions is shown, together with the setpoints used in PC and the activation threshold. We observe practically no difference after dynamic control of bus lane is introduced on top od PC. In 38(b) the implemented changes in green time of the controlled approaches due to PC is shown, where green time is adjusted to reduce inflow and increase outflow of region 2, which is protected from congestion. In 38(c) we see the evolution of density inside the dynamic lane during the simulation, together with the control-defined percentage of allowed cars. In 38(d), the evolution of average density in the mixed-use lanes of the corridor is shown.

For comparison reasons, figure 39 shows simulation results of scenarios where only a fixed percentage of cars are allowed in the bus lane corridor. without PC,



(a) Accumulation

(b) Green time

(c) Density on dynamic lanes

(d) Average density on mixed lanes

Figure 38: Analysis on one replication of the case PC + DyBL. (a) Regional accumulation time-series with PC setpoints and activation threshold; (b) Time-series of interregional green time for the PC intersections; (c) Time-series of density in the dynamic lane for the cases of single PC and PC + DyBL; (d) Time-series of average density in mixed-use lanes of the corridor, for the cases of single PC and PC + DyBL.

for the same demand scenario. Again, the vertical axis represents the percentile improvement of PHT with respect to the FTC case. While further experiments are necessary to validate these findings, we observe a slight improvement in the case where 10% of the cars are using the bus lane compared to the case of DBL (no cars allowed) and a smaller improvement for the case of 20% for the bus travel time, as well as the total. Interestingly, all cases of operational bus lane seem to be improved compared to the FTC with no lane restriction, while 10% allowed car rate seems to be better than 20%, both for the buses and overall.

5.2.5 Findings

In this preliminary analysis, the concept of dynamic bus lanes is implemented in combination with perimeter control in a large-scale network in microscopic simulation environment. Simple dynamic feedback regulators are tested both for the green time adjustment of PC as well as for the control of car entry rate in the dy-

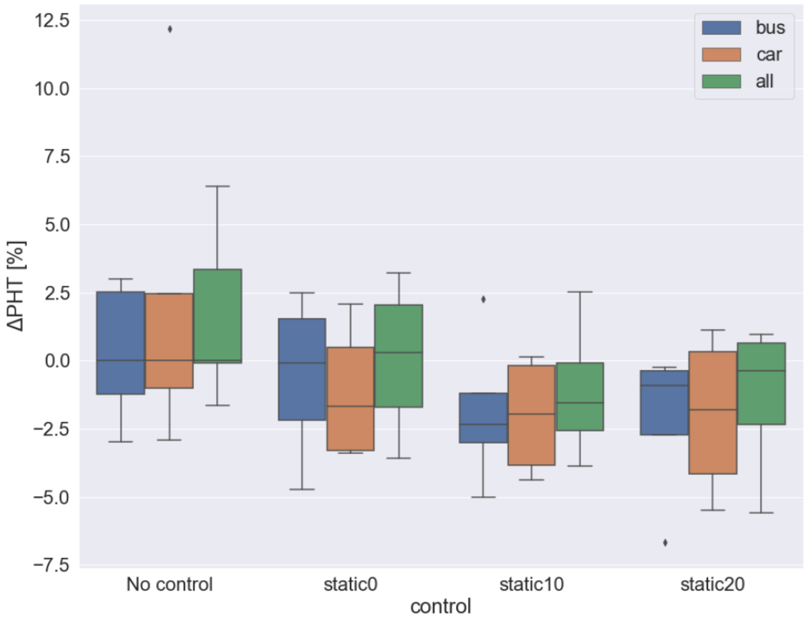


Figure 39: PHT change in cases of fixed allowed car percentage in the dynamic lane compared to the FTC case.

namic lanes. The results of a limited preliminary study for one moderate demand scenario show that the combined PC with the dynamic control of bus lanes tend to lead in better performance in terms of total passenger travel time, being more beneficial for both cars and bus passengers. Further research is required to validate this finding, by testing improved regulators for both controllers and potentially by considering state variables directly related to buses, such as bus speed.

6 Strategy and design of the Utrecht pilot

6.1 Traffic management platform

In this section we discuss the dynamic traffic management platform equipped in the pilot city of Utrecht, the Netherlands, describing its current functioning and detailing the extensions that are planned in relation to this project. The content in the section is summarized by TUD and Arane.

6.1.1 Integrated network management (INM)

The INM system is developed for an urban network where both arterial streets and highways occur in the network (see Figure 40). The system considers the state of congestion and starts reacting at different points in the network, not only locally. The current state is measured, and real-time adaptation of traffic lights is taken. The INM system is therefore classified as adaptive coordinated control. Initially, the system was designed to avoid congestion on the highway while using ramp metering to assure the flow of cars. The main principles followed by this control method are [30]:

1. Use spare capacity while using available buffer space in the network
2. Prevent capacity drop
3. Prevent spill-back and blockades
4. Bottleneck resolution, first locally then network-wide

Measures are taken in two steps; first, the outflow is increased, and then, if necessary, the inflow is limited. The control approach in Utrecht uses the same principles as the highway-developed control scheme on an urban network. A critical corridor is defined based on historical events. It is supervised by the bottleneck investigator (Dutch: Kiemenspeurder), which predicts the moment of a possible breakdown three minutes ahead of its occurrence for a highway bottleneck. In the case of an urban bottleneck, the prediction time is shorter. The process of the system is summarised in Figure 41. With radars in the urban sectors and loop detectors on the highway, the number of cars is measured and given as input to the queue estimator and the bottleneck investigator. Together they build the monitoring unit. The outputs are given to the network supervisor, who builds the strategic control layer. It defines which intersection becomes a Master or a Cleaner. The network supervisor passes the information on to the sub-network supervisor, which oversees the available buffer space around the Masters. It can activate the Slave and Guard functions at intersections. The intersection's role defines how green times of that intersection are calculated, and this information is then given to the traffic signals.

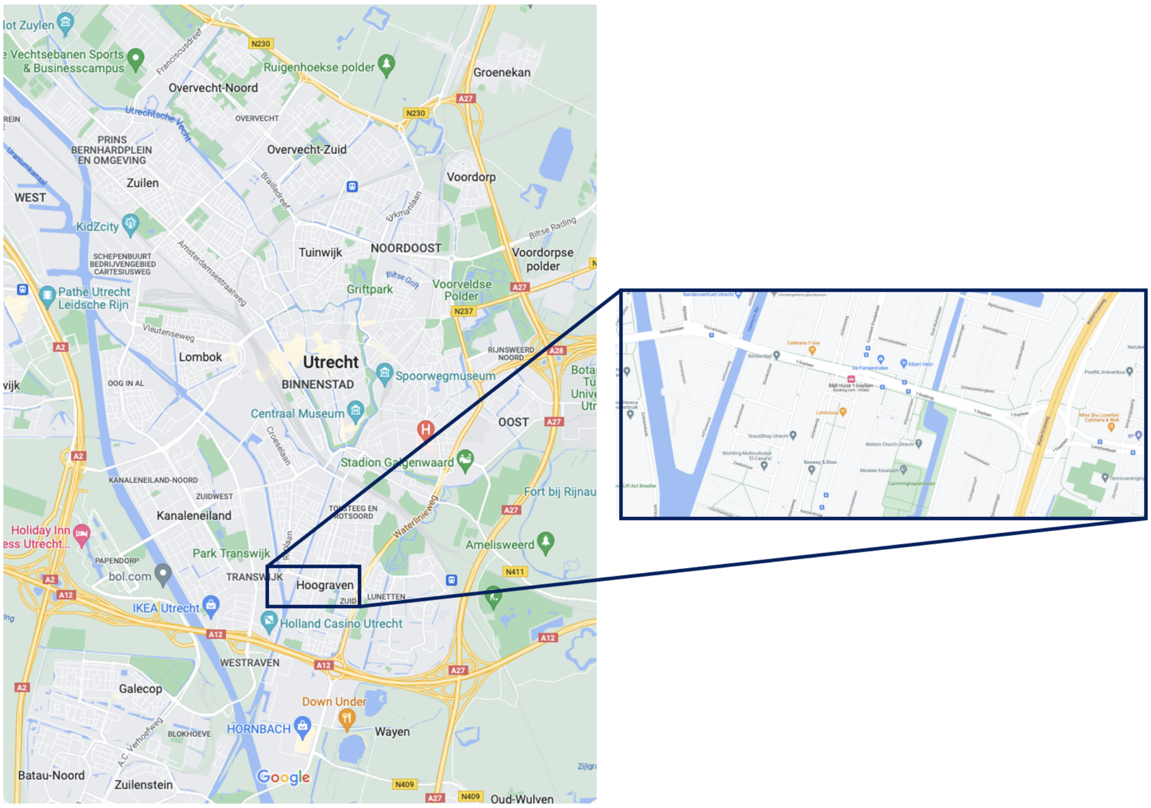
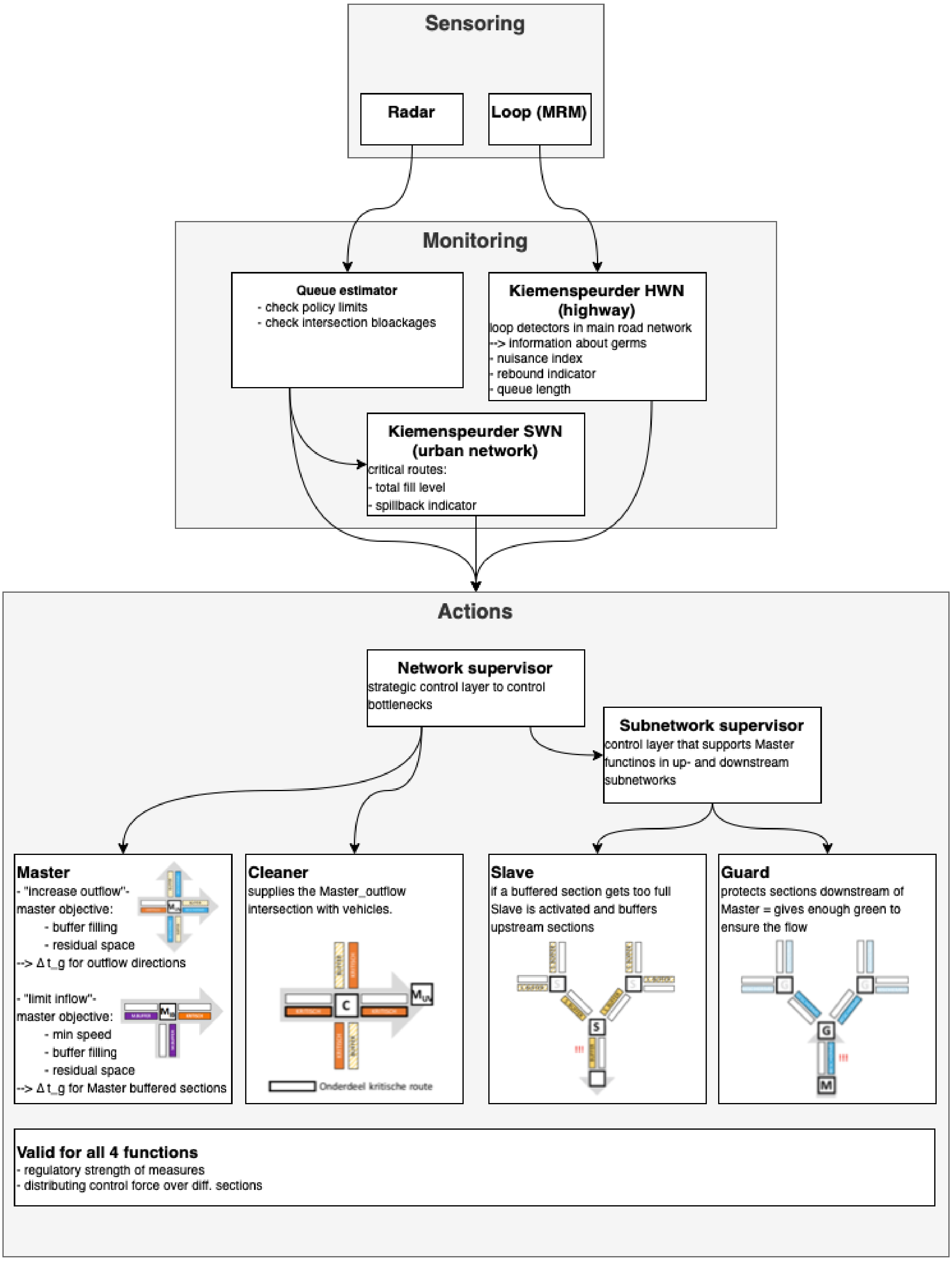


Figure 40: Location of the pilot in Utrecht (Source: Google Maps).

6.1.2 Modules of the INM system

Figure 42 shows the implementation of the INM system in one direction of 't Goylaan, an arterial corridor where the system is currently operational in the city of Utrecht (NL). The intersection downstream of the critical corridor can be entitled with a *Master* function to increase the outflow (1). The first intersection upstream of the Master provides the Master intersection with cars to process through the critical corridor, thus becoming a *Cleaner* (2). The intersection downstream can be activated to guarantee the accommodation of the increased number of cars; this function is called the *Guard* (3). Through the activation of the Guard, some downstream directions are given more green, and competing movements are likely to be held back (4). Finally, the *Slave* function can be activated everywhere where certain movements are held back to the benefit of the traffic flow on the critical corridor (5). The Slave makes sure to distribute queues over the system evenly. If these measures are not enough to stabilize the system, the second step, limiting the inflow, can also be applied to the urban corridor. In that case, the intersection upstream of the critical corridor becomes a "limit inflow"–Master and some vehicles are buffered in the surrounding approaches (6). This calls again for Slave functions at the upstream intersections (7). More details about the system's functions can be found in the module specification.



Basic specification INM rule approach

Figure 41: Scheme of the INM system.

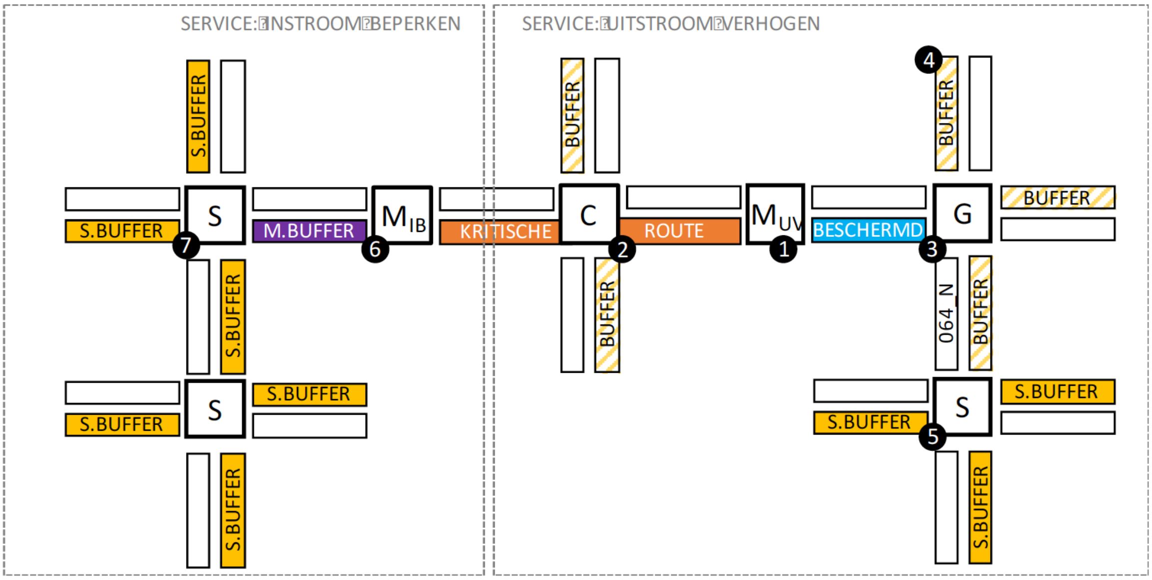


Figure 42: Example of implemented INM system in 't Goylaan [31].

Note that the normalization process for each module is not explicitly explained in this work, but can be found in the basic specification [31]. The idea behind the normalization process is to have more stability when switching the system off. The system is not needed anymore if the number of cars drops below the activation threshold. To prevent fluctuations around the threshold, the system is switched off based on the average sum delta green time and a smoothing parameter.

"Increasing Outflow"–Master

If the given threshold of the filling in the critical road section is reached, the Master is switched on. First, the Master intersection increases the outflow, which means green time in the out-flowing directions is increased. This increase is determined based on real-time measurements which are, in this case, the total queue length x in the critical corridor. The variable is updated through a feedback control mechanism, which brings the advantage of less abrupt changes and considers the value of the previous time step. Equation (38) shows an example for the sum of delta green time Δg^{Σ}.

$$\Delta g^{\Sigma,\text{current}} = \Delta g^{\Sigma,\text{previous}} - K_1 \times \left(x^{\text{current}} - x^*\right) \times \Delta T - K_2 \times \left(x^{\text{current}} - x^{\text{previous}}\right) \tag{38}$$

where x refers to queue length, x^* is the target value, K_1 and K_2 are tuning parameters respectively associated with the deviation of the target value and the variance of the previous time step, and ΔT is the time step.

The calculated sum of delta green times needs to respect the minimum and maximum values (39).

$$\Delta g^{\Sigma,\text{current}} = \min\left\{\max\left\{\Delta g^{\Sigma,\text{current}}, \sum_{i^{\text{dir}} \in J^{\text{dir}}_{i\text{Master}}} \Delta g^{\text{min}}_{i\text{dir}}\right\}, \sum_{i^{\text{dir}} \in J^{\text{dir}}_{i\text{Master}}} \Delta g^{\text{max}}_{i\text{dir}}\right\} \tag{39}$$

where $\Delta g^{\text{min}}_{i\text{dir}}$ and $\Delta g^{\text{max}}_{i\text{dir}}$ are the minimum and maximum green times per direction and $J_{i\text{Master}}$ are the available direction for that Master intersection.

All directions should increase their respective outflow, therefore the sum of delta green times is divided proportionally to the range of green time e.g., maximum green time $\Delta g^{\text{max}}_{i\text{dir}}$ - minimum green time $\Delta g^{\text{min}}_{i\text{dir}}$. This then gives the delta green time per direction i^{dir} in Equation (40).

$$\Delta g^{\text{Master}}_{i\text{dir},i\text{Master}} = \frac{\Delta g^{\Sigma,\text{current}}}{\sum_{j \in J^{\text{dir}}} \Delta g^{\text{max}}_{j} - \Delta g^{\text{min}}_{j}} \left(\Delta g^{\text{max}}_{i\text{dir}} - \Delta g^{\text{min}}_{i\text{dir}}\right) \tag{40}$$

Cleaner

The intersection upstream of a Master intersection is automatically activated to become a Cleaner. This means that it makes sure to supply the Master intersection with incoming cars in the queue while increasing the green times in the direction of the Master. The green times are calculated based on the buffer filling. The principle is the same as for the Master; feedback control is used to update the sum of delta green times (41), and then again, the minimum and maximum values are controlled for (42).

$$\Delta g^{\Sigma,\text{current}} = \Delta g^{\Sigma,\text{previous}} - K_1 \times \left(r^{\text{fill,current}} - r^{\text{fill},*}\right) \times \Delta T - K_2 \times \left(r^{\text{fill,current}} - r^{\text{fill,previous}}\right) \tag{41}$$

where r^{fill} is the relative buffer fill, $r^{\text{fill},*}$ the target value (typically 20%), and the other terms are the same as for the Master in (39).

The minimum and maximum values of delta green times must be respected:

$$\Delta g^{\Sigma,\text{current}} = \min\left\{\max\left\{\Delta g^{\Sigma,\text{current}}, \sum_{i^{\text{dir}} \in J^{\text{dir,Cleaner}}_{i\text{Cleaner}}} \Delta g^{\text{min}}_{i\text{dir}}\right\}, \sum_{i^{\text{dir}} \in J^{\text{dir,Cleaner}}_{i\text{Cleaner}}} \Delta g^{\text{max}}_{i\text{dir}}\right\} \tag{42}$$

Then for each direction, the delta green time is calculated based on the range of delta green time:

$$\Delta g_{i\text{dir}}^{\text{Cleaner}} = \frac{\Delta g^{\Sigma,\text{current}}}{\sum_{j\in J^{\text{dir,Cleaner}}} \Delta g_j^{\max} - \Delta g_j^{\min}} \left(\Delta g_{i\text{dir}}^{\max} - \Delta g_{i\text{dir}}^{\min}\right) \tag{43}$$

Guard

Depending on the number of cars on the road section between the Master and its downstream intersection, that intersection will be activated as a Guard. The Guard ensures that enough buffer space is created for the cars coming from the Master. This means that the green times of that approach are adapted, and competing movements are (potentially) buffered. Green times are calculated following the same principle as for the Cleaner. The only difference is that the activation of the Guard relies on road sections downstream of the Master, whereas the Cleaner depends on the ones upstream.

$$\Delta g^{\Sigma,\text{current}} = \Delta g^{\Sigma,\text{previous}} - K_1 \times \left(r^{\text{fill,current}} - r^{\text{fill},*}\right) \times \Delta T - K_2 \times \left(r^{\text{fill,current}} - r^{\text{fill,previous}}\right) \tag{44}$$

$$\Delta g^{\Sigma,\text{current}} = \min\left\{\max\left\{\Delta g^{\Sigma,\text{current}}, \sum_{i\text{dir}\in J_{i\text{Guard}}^{\text{dir,Guard}}} \Delta g_{i\text{dir}}^{\min}\right\}, \sum_{i\text{dir}\in J_{i\text{Guard}}^{\text{dir,Guard}}} \Delta g_{i\text{dir}}^{\max}\right\} \tag{45}$$

$$\Delta g_{i\text{dir}}^{\text{Guard}} = \frac{\Delta g^{\Sigma,\text{current}}}{\sum_{j\in J^{\text{dir,Guard}}} \Delta g_j^{\max} - \Delta g_j^{\min}} \left(\Delta g_{i\text{dir}}^{\max} - \Delta g_{i\text{dir}}^{\min}\right) \tag{46}$$

Slave

If traffic movements are buffered in adjacent roads, the activation of the Slave function in the upstream intersections takes place when a certain threshold of the buffer space filling in the road is reached. The Slave's main goal is to prevent blockages of intersections due to said buffering. Consequently, the green times are calculated so that the queues in each approach to the intersection are equally distributed.

$$\Delta g^{\Sigma,\text{current}} = \Delta g^{\Sigma,\text{previous}} - K_1 \times \left(r^{\text{fill,current}} - r^{\text{fill},*}\right) \times \Delta T - K_2 \left(r^{\text{fill,current}} - r^{\text{fill,previous}}\right) \tag{47}$$

$$\Delta g^{\Sigma,\text{current}} = \min\left\{\max\left\{\Delta g^{\Sigma,\text{current}}, \sum_{i\text{dir} \in J_{i\text{Slave}}^{\text{dir,Slave}}} \Delta g_{i\text{dir}}^{\min}\right\}, \sum_{i\text{dir} \in J_{i\text{Slave}}^{\text{dir,Slave}}} \Delta g_{i\text{dir}}^{\max}\right\} \tag{48}$$

$$\Delta g_{i\text{dir}}^{\text{Slave}} = \frac{\Delta g^{\Sigma,\text{current}}}{\sum_{j \in I^{\text{dir}}} r_j^{\text{space,upstream}}} r_{i dir}^{\text{space,upstream}} \tag{49}$$

"Limiting Inflow"–Master

If the critical road section filling exceeds a second threshold (higher than for the "Increasing Outflow"–Master), the inflow is limited, and traffic is buffered upstream. In this case, the intersections upstream turn into a "limiting inflow"–Master, and the intersection(s) upstream of the Master can turn into Slave(s) to distribute the queues in the buffers.

In the case of the "limiting inflow"–Master, the sum of delta green times is calculated as in Equations (39) and (40). The delta green times are calculated proportionally to the available buffer space (50). As a result of this, the sum of delta green times is negative since the system reduces the green time to decrease the inflow. The resulting green time can be smaller than the minimum green time because of the fraction of available space. In that case, the green time needs to be at least the minimum value, and the other green times need to be recalculated according to the new value of the sum of delta green times.

$$\Delta g_{i\text{dir},i\text{Master}}^{\text{Master}} = \frac{\Delta g^{\Sigma,\text{current}}}{\sum_{j \in I^{\text{dir}}} r_j^{\text{space,upstream}}} r_{i dir}^{\text{space,upstream}} \tag{50}$$

6.2 DIT4TraM extensions to INM

The functions described in the earlier sections represent a task-wise decentralization of traffic management actions, following expert design principles. In its current form and implementation, the approach is static, that is, the roles of the intersections are defined a-priori and can only be activated or deactivated, based on the aforementioned thresholds. An intersection cannot be co-opted to act as Slave if its role has been predetermined as Guard, or vice-versa.

One of the key outcomes envisioned through the developments of DIT4TraM, and specifically in relation to the pilot study planned in the city of Utrecht, is that of introducing dynamicity to this approach, guiding the selection and assignment of intersection tasks based on the status of the traffic network. To this end, the Jam Tree approach discussed in earlier sessions will be adopted, through appropriate parameter calibration. Once a Jam Tree is detected, by tracking its spatio-temporal evolution the approach will allow identifying which intersections contribute to the target bottleneck, and guide the appropriate distribution of tasks (Master, Guard, Cleaner, Slave) leading to optimal bottleneck resolution. In order to achieve said objective, the INM approach must be further extended in order to be able to take full advantage of additional data sources, chiefly Floating Car Data (FCD), therefore ensuring that the collected network information is of sufficiently high quality. The integration of FCD allows dynamic assignment of the functions Master, Guard, Cleaner and Slave. This prevents suboptimal control performance during a-typical situations when functions are assigned statically.

Finally, extensions to the core functionality of the different modules will be sought, specifically in relation to measurements originating from soft modes. To support this development, partners involved in the Utrecht pilot are setting up appropriate data sources to collect bicycle information (trajectories, approaching the controlled intersection(s)). An objective of the coming developments is to extend the logic of INM to account for the quality of service of (competing) soft modes, thereby extending toward multi-modal management. A preliminary study was conducted considering potential multi-modal extensions for Public Transport along the 't Goylaan corridor, leading to recommendations as to how the logic of key INM components could be extended to ensure better performance [32].

Benefits of integrating Floating Car Data

The use of FCD to identify Jam Trees to estimate the current state of the traffic network has multiple benefits. Besides the possibility to dynamically assign roles to intersection controllers and ramp meters based on Jam Trees, it also allows us to use the FCD as sensor in the monitoring modules of the system. Currently, radar systems are used to estimates queue lengths and available bufferspace. The control modules take actions based on these estimates. If we make these estimates based on FCD, which is less detailed than radar, the spatio-temporal quality will not be the same. However, it allows us to avoid using radar data. At the boundaries of the pilot network, we will test if we can achieve the control goals with the lower level of detail. If this is succesfull, the scalability of INM will increase since FCD is widely available in the Netherlands.

Approach of the pilot

Currently, the FCD is gathered at the pilot site. For each road section between intersections, a 1-minute average speed is delevered. Based on these speed measures, a spatio-temporal analysis is performed. The goal of the analysis is to find up- and downstream interactions between road sections: under which conditions do we identify spillback onto upstream roads? The data analysis leads to a specification of the Jam Tree algorithm, which will be implemented by Technolution.

The involved control modules that will use the Jam Trees are the network supervisor and the subnetwork supervisor. They will be adjusted to allow dynamic assignment of functions within the network. To achieve this, they need a new interface with the Jam Tree identifier. After testing, the DIT4TraM pilot period will be used to evaluate the system. During the pilot the new and adjusted mudules will be switched on-and-off on a weekly basis.

Parallel to the Jam Tree development, the possible use of FCD in INM will be analysed. This is done by analysing historical data from the monitoring modules in the INM-system, and by comparing them with FCD. The different spatio-temporal resolutions (10 sec $\times$ 7 meters versus 1 minute $\times$ $\approx$ 150 meters) are addressed by comparing different aggregates. Based on the results a specification for adjusted control modules will be delivered for implementation by Technolution. The adjusted modules will use FCD instead of radar-data. Also these modules will be tested during the pilot period by switching them on and off on a weekly basis. The evaluation will focus on how successful the INM-system can reach it control targets with FCD.

7 Conclusion

Task 3.1 has developed a variety of distributed algorithms to govern transportation networks, that are geographically distributed and with multiple entities and operators.

At the intersection level, we describe a novel framework for controlling intersections called the Integrated Signal and Bus Lane Control (ISBLC). This framework aims to maximize the passenger throughput at intersections and provide the highest level of service for public transport passengers. The ISBLC adjusts traffic signals and assigns dedicated bus lanes based on observed traffic conditions and information received from Vehicle-to-Infrastructure (V2I) communication regarding vehicle/bus passenger occupancy and bus delays. A state-of-the-art reinforcement learning algorithm is used to reduce total passenger waiting time. The results of the ISBLC are compared to traditional traffic signal and bus prioritization approaches to evaluate its effectiveness and inform future research towards passenger-oriented and multi-modal traffic control schemes.

At the corridor level, the proposed DRL model appears to be a promising solution for the efficient management of traffic signals in multi-modal corridor-level networks. By combining the advantages of reinforcement learning and deep learning, the model can effectively capture the complex interactions between private car traffic and bus transit, and adapt to different road configurations.

At the network level, we have developed a two-layer control framework, which combines the benefits of both Max Pressure and Perimeter Control strategies to improve traffic signal control performance in congested networks. The upper layer is dedicated to PC and operates at an aggregated scale, specifying the target inter-regional exchange flows between congested regions. The lower layer is dedicated to MP and operates at a distributed level, adjusting the green times of intersections based on real-time queue measurements.

In the multi-modal context, we proceed to use reinforcement learning algorithms to optimally control the allocation of road space between buses and vehicles in real-time, taking into account the dynamic traffic conditions and bus delays. The algorithms are trained to minimize bus delay and to ensure that the travel time of the vehicular traffic is not significantly affected. The IDBL strategy creates dynamic bus lanes that adapt to the changing traffic conditions, allowing for a more efficient use of the road space for buses and reducing the likelihood of bus delays. The BLDC strategy takes this one step further by also considering the density of vehicles in the general traffic lanes, thereby ensuring a more balanced allocation of road space.

Finally, Task 3.1 envisions that some dynamicity will be further introduced the INM system, which is currently functioning in the Utrecht pilot. In particular, the selection and assignment of intersection tasks would be based on the status of the traffic network with the development in DIT4TraM. The Jam Tree approach will be adopted such that when a Jam Tree is detected, by tracking its spatio-temporal evolution, the approach will allow identifying which intersections contribute to the target bottleneck, and guide the appropriate distribution of tasks leading to optimal bottleneck resolution.

Publications for D3.1

[1] D. Tsitsokas, A. Kouvelas, and N. Geroliminis, "Two-layer adaptive signal control framework for large-scale networks combining efficient max-pressure and perimeter control," in *Transportation Research Part C: Emerging Technologies (under review)*, Washington, D.C., 2023.

[2] J. Yu, P.-A. Laharotte, and L. Leclercq, "Decentralized traffic signal control with deep reinforcement learning in a multi-modal network: Seeking for a trade-off between bus service and traffic objectives," in *102nd Annual Meeting of the Transportation Research Board*, Washington, D.C., 2023.

[3] D. Tsitsokas, A. Kouvelas, and N. Geroliminis, "Two-layer adaptive signal control framework for large-scale dynamically-congested networks: Combining efficient max-pressure with perimeter control," *Transportation Research Part C: Emerging Technologies (under review)*, 2022.

[4] J. Yu, P.-A. Laharotte, and L. Leclercq, "Distributed signal control for multimodal traffic network: A deep reinforcement learning approach," *Transportation Research Part C: Emerging Technologies (under review)*, 2022.

References

[1] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE internet of things journal*, vol. 1, no. 4, pp. 289–299, 2014.

[2] Q. Guo, L. Li, and X. J. Ban, "Urban traffic signal control with connected and automated vehicles: A survey," *Transportation research part C: emerging technologies*, vol. 101, pp. 313–334, 2019.

[3] F. Rasheed, K.-L. A. Yau, R. M. Noor, C. Wu, and Y.-C. Low, "Deep reinforcement learning for traffic signal control: A review," *IEEE Access*, vol. 8, pp. 208 016–208 044, 2020.

[4] S. S. S. M. Qadri, M. A. Gökçe, and E. Öner, "State-of-art review of traffic signal control methods: Challenges and opportunities," *European transport research review*, vol. 12, pp. 1–23, 2020.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[6] P. A. Lopez, M. Behrisch, L. Bieker-Walz, *et al.*, "Microscopic traffic simulation using sumo," in *2018 21st international conference on intelligent transportation systems (ITSC)*, IEEE, 2018, pp. 2575–2582.

[7] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 3389–3396.

[8] V. Mnih, A. P. Badia, M. Mirza, *et al.*, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, PMLR, 2016, pp. 1928–1937.

[9] F. V. Webster, "Traffic signal settings," Tech. Rep., 1958.

[10] P. Koonce and L. Rodegerdts, "Traffic signal timing manual.," United States. Federal Highway Administration, Tech. Rep., 2008.

[11] G. A. C. (DLR), *SUMO Documentation*, en. [Online]. Available: https://sumo.dlr.de/docs/Simulation/Output/StatisticOutput.html (visited on 02/24/2023).

[12] R. Wunderlich, C. Liu, I. Elhanany, and T. Urbanik, "A novel signal-scheduling algorithm with quality-of-service provisioning for an isolated intersection," *IEEE Transactions on intelligent transportation systems*, vol. 9, no. 3, pp. 536–547, 2008.

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

[14] A. H. Chow, Z. Su, E. Liang, and R. Zhong, "Adaptive signal control for bus service reliability with connected vehicle technology via reinforcement learning," *Transportation Research Part C: Emerging Technologies*, vol. 129, p. 103 264, 2021.

[15] L. C. Edie *et al.*, *Discussion of traffic stream measurements and definitions*. Port of New York Authority New York, 1963.
[16] L. Leclercq, N. Chiabaut, and B. Trinquier, "Macroscopic fundamental diagrams: A cross-comparison of estimation methods," *Transportation Research Part B: Methodological*, vol. 62, pp. 1–12, 2014.
[17] N. Geroliminis and A. Skabardonis, "Identification and analysis of queue spillovers in city street networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1107–1115, 2011.
[18] A. Kouvelas, M. Saeedmanesh, and N. Geroliminis, "Enhancing model-based feedback perimeter control with data-driven online adaptive optimization," *Transportation Research Part B: Methodological*, vol. 96, pp. 26–45, 2017.
[19] D. Tsitsokas, A. Kouvelas, and N. Geroliminis, "Modeling and optimization of dedicated bus lanes space allocation in large networks with dynamic congestion," *Transportation Research Part C: Emerging Technologies*, vol. 127, p. 103 082, 2021.
[20] M. Saeedmanesh and N. Geroliminis, "Clustering of heterogeneous networks with directional flows based on "snake" similarities," *Transportation Research Part B: Methodological*, vol. 91, pp. 250–269, 2016.
[21] M. Eichler and C. F. Daganzo, "Bus lanes with intermittent priority: Strategy formulae and an evaluation," *Transportation Research Part B: Methodological*, vol. 40, no. 9, pp. 731–744, 2006.
[22] J. Viegas and B. Lu, "The intermittent bus lane signals setting within an area," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 6, pp. 453–469, 2004.
[23] Y. Lin, X. Yang, N. Zou, and M. Franz, "Transit signal priority control at signalized intersections: A comprehensive review," *Transportation letters*, vol. 7, no. 3, pp. 168–180, 2015.
[24] K. Othman, A. Shalaby, and B. Abdulhai, "Dynamic bus lanes versus exclusive bus lanes: Comprehensive comparative analysis of urban corridor performance," *Transportation Research Record*, vol. 2677, no. 1, pp. 341–355, 2023.
[25] W. Wu, L. Head, S. Yan, and W. Ma, "Development and evaluation of bus lanes with intermittent and dynamic priority in connected vehicle environment," *Journal of Intelligent Transportation Systems*, vol. 22, no. 4, pp. 301–310, 2018.
[26] M. W. Levin and A. Khani, "Dynamic transit lanes for connected and autonomous vehicles," *Public Transport*, vol. 10, pp. 399–426, 2018.
[27] P. Anderson and N. Geroliminis, "Dynamic lane restrictions on congested arterials," *Transportation research part A: policy and practice*, vol. 135, pp. 224–243, 2020.
[28] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident analysis & prevention*, vol. 35, no. 3, pp. 427–433, 2003.
[29] T. Haarnoja, A. Zhou, K. Hartikainen, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[30] S. Hoogendoorn, R. Landman, J. van Kooten, and M. Schreuder, "Integrated network management amsterdam: Control approach and test results," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, ISSN: 2153-0017, Oct. 2013, pp. 474–479. DOI: 10.1109/ITSC.2013.6728276.
[31] E.-S. Smits, J. Van Kooten, K. Adams, and F. Kessels, *Basisspecificatie GNV-regelaanpak. RVM midden-nederland*, 2020.
[32] P. Vogg, "Intelligent multimodal traffic management: A case study in utrecht, the netherlands," M.S. thesis, Civil Engineering Section, EPFL, Sep. 2022.

DIT4TraM

Distributed Intelligence & Technology
for Traffic & Mobility Management

This project has received funding from the European Union's Horizon 2020
research and innovation programme under Grant Agreement no. 953783.