DIT4TraM

Distributed Intelligence & Technology
for Traffic & Mobility Management

Benefits of a distributed approach to mobility management, and preconditions for effectiveness

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	D1.1
Deliverable Title	Benefits of a distributed approach to mobility management, and preconditions for effectiveness
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, Dirk Helbing, Marco Rinaldi, Shlomo Havlin, Efrat Blumenfeld, Damian Dailisan, Dimitrios Tsitsokas, Mingru Wang, Caio Vitor Beojone, Manon Seppecher, Jiajie Yu, Mina Khalesian

Release Approval

Name	Role	Date
Serge Hoogendoorn	Project reviewer	February 27th 2023
Erik-Sander Smits	Project reviewer	February 27th 2023
Nikolas Geroliminis	Task package leader	February 28th 2023
Dirk Helbing	Work package leader	February 28th 2023

Document History

Version	Description	Name
1	Versions for internal review among authors	DIT4tramWP1_1_deliverable_Ver1
2	Versions for internal review among project members	DIT4tramWP1_1_deliverable_Ver2
3	Versions ready for release	DIT4tramWP1_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 1.1 of the DIT4Tram Project, comprised of research and development work carried out in Task 1.1: *Benefits of a Distributed Approach*, as part of Work Package 1: *Next-generation multimodal traffic and transportation management paradigm* during the period M1–M18.

This report details theoretical and algorithmic approaches toward the study of distributed control paradigms in transportation. As modern transportation systems increase in complexity, high computational costs and resilience concerns pose barriers to the current centralized control approach. Together with growing concerns about data-security and privacy, decentralized, distributed systems offer a promising alternative to centralized systems.

After a brief introduction to the different kinds of non-centralized control architectures in chapter 1, chapter 2 presents the benefits of decentralization for transportation systems, and examines the price of anarchy in decentralized systems. We find in a stylized model that, it is possible that distributed systems with suitable information exchange and feedback/incentive systems can actually perform optimally or close to optimal with the methodologies developed in DIT4TraM. We point out that, neglecting the effect of subsystem interactions and uncertainties due to humans in the loop, can result in an undesirable evolution of the system state.

The core of this report lies in presenting various designs and implementations of distributed transportation systems in Chapter 3. These approaches range from i) control of traffic signals at intersections, ii) using dynamic and flexible lane use in multimodal transport settings, iii) repositioning and redistribution of on-demand transport vehicles according to time-varying demand distribution, iv) auctioning schemes for multimodal mobility services, and v) cooperative schemes for local bottleneck control.

We start with envisioning various passenger-oriented control architectures for large-scale urban systems, which will benefit Task 3.1. First, we look into the traffic signal control problem and propose three different methods.

- To exemplify the potential of distributed control in traffic signal control, we proposed a two-layer hierarchical adaptive signal control framework for a network-wide application, combining centralized perimeter control (PC) with distributed Max Pressure control.
- We also introduce reinforcement learning (RL) as a compatible approach, which learns from experience and data to adapt to changing traffic patterns. Compared to the centralized RL algorithm, the state-action space of Multi-Agent Reinforcement Learning agents is reduced significantly, leading to better training performance.

- A jam-tree approach is developed to identify traffic bottlenecks and facilitate better signal control.

We also examine the effectiveness of dynamic Bus Lanes, which regulate the number of vehicles allowed in the bus lane based on real-time traffic information to optimize road space use, reduce congestion, and improve the performance of both buses and cars. Dynamic feedback regulators adjust the number of cars allowed in the bus lane in response to traffic conditions.

Second, distributed approaches are also expected to be applied to the repositioning of on-demand vehicles. We proceed to emphasize the problem of repositioning on-demand transport vehicles, which will benefit Task 3.3. In particular, we design different decentralized decisional architecture consisting of a mesh of controllers that divide the urban network into as many service areas. Three different approaches to re-balance the temporal and spatial distribution of on-demand transport vehicles, resorting to auction, multi-layer control and incentization,respectively.

Third, dynamic bidding auction strategies are promising venues for decentralizing traffic management. The report also discusses auctioning schemes for multimodal mobility services, which will benefit Task 4.2. We propose to reshape traffic signal priority by auctioning strategies so as to take into account e.g. vehicle occupancy, public transport services, ride-sharing, etc, contributing to sustainability objectives.

Last, we also illustrate the connection between Task 1.1 with the cooperative schemes for local bottleneck control, which will be developed in Task 2.4. By introducing information sources from connected vehicles, we develop appropriate prediction models in an MPC framework for both non-connected and connected traffic participants, reflecting the different information quality and granularity. Thus, cooperation between the different users (classes) can be achieved under certain conditions.

1 Introduction

Modern transportation systems are becoming increasingly complex due to the introduction of new travel modes, the expansion of networks, and the introduction of novel transportation policies. Different components of transportation systems interact with each other, which results in a series of high-complexity decision problems. The constant interaction between different components often hinders the efficiency of centralized control methods due to the inherent computational efficiency, resilience, and information-exchanging capacity. Thus, developing novel traffic control architecture in a distributed way has become increasingly essential in modern transportation systems.

In the past decades, an enormous number of methods for urban traffic control and management systems have been developed and implemented all over the world to alleviate congestion. However, many practical limitations, including accuracy and computational burden, have hindered the possible success of these optimization and control methods. Local adaptive strategies that are widely used around the world are based on heuristic optimization techniques but might compromise efficiency under complex interactions among different subsystems, such as congestion propagation phenomena and queue spillbacks. Other traffic control strategies use sophisticated global optimization methods, which make their online application to large-scale urban networks difficult due to expensive computational costs. Therefore, it remains a significant challenge to design efficient control strategies for heterogeneous large-scale transportation networks that deal with complex traffic conditions.

1.1 Major control architectures

In this section, we briefly summarize the major non-centralized control architectures developed in many complex systems, including decentralized control, distributed control, and hierarchical control. Similar to urban transportation systems, many systems with a complex multi-scale or so-called hierarchical feature requires some sophisticated modelling and control framework to optimize system performance. Despite many advantages of controlling such systems with centralized control architectures, such as the optimality, the computational burden and scalability issues become less tractable as the complexity grows. There are increasing applications of applying multi-scale modelling/control in various domains such as the electrical grid (e.g.,[1]–[4]), biology systems (e.g.,[5]), chemical industry (e.g.,[6]), and meteorology (e.g., [7], [8]). Despite some recent moderate developments including model-based approaches such as singular perturbation theory (e.g.,[9]–[11]), and data-drive approaches such as dynamic model decomposition (e.g., [12], [13]), systematic theories for analysis or computation for general system remains limited. Readers are referred to [14] for a comprehensive review.

1.1.1 Decentralized control

Many complex transportation systems are controlled under decentralized architectures, in which the input (control variable) and the output (controlled variable) are assigned to disjoint subsystems, as shown in Figure 1. When the interaction among subsystems is weak, the design of local regulators for subsystems is straightforward once the decentralized control architecture is defined. However, when strong interaction is presented among subsystems, the stability and performance of the decentralized controller would be significantly affected [15].

In view of the great importance of stability and performance under decentralized control, many efforts have been made in the existing studies. Some of them rely on the vector Lyapunov functions (e.g., [14]), which is a mathematical tool used to analyze the stability of dynamical systems. In contrast, others consider sequential design (e.g., [16]), optimization (e.g., [17]), overlapping decompositions (e.g., [18]).

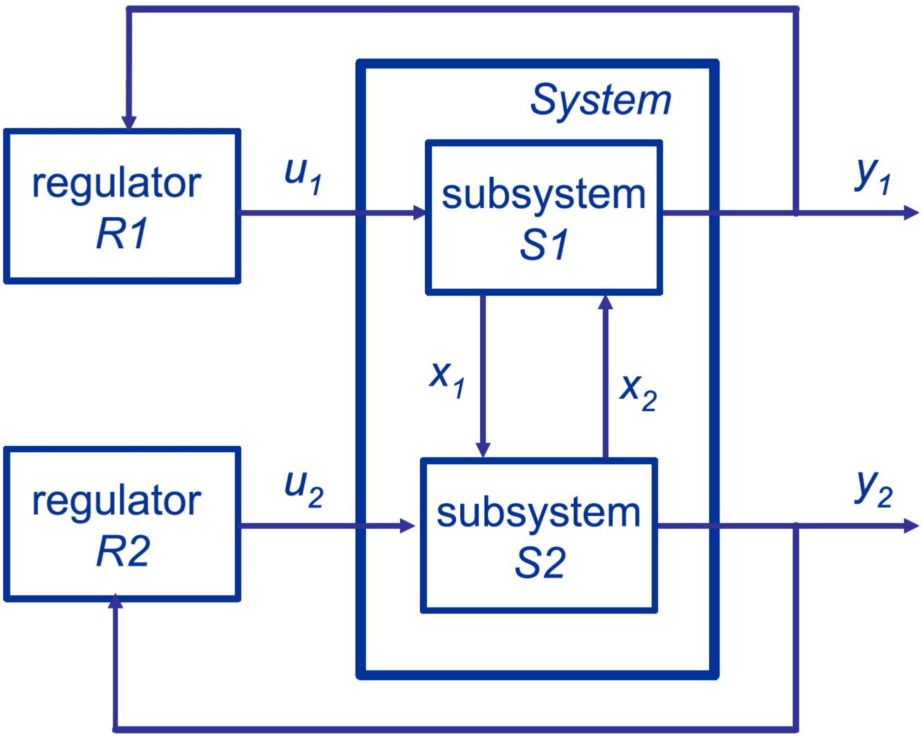


Figure 1: Decentralized control of a two input (u_1, u_2)–two output (y_1, y_2) system (adapted from [14]), where x_1 and x_2 are the states of subsystems S_1 and S_2

1.1.2 Distributed control

Distributed control architectures feature information exchange among local regulators. The information for exchanging is often concerned with the future predicted control or locally computed state variables over the considered prediction horizon. Then any local regulator can predict the interaction effects over the considered prediction horizon. An example is indicated in Figure 1, in which local regulators (R1 and R2) communicate with each other about their unilateral decisions.

The performance of a distributed control is determined by the protocols of information transmission and synchronization. Generally, two major types of algorithms exist (1) fully connected algorithms with full information transmission between all local regulators and (2) partially connected algorithms with information transmission. Ref. [19] have demonstrated that restricting the information exchange among directly interacting subsystems produces a negligible performance deterioration. Meanwhile, in each sample time, the information exchange can happen once or multiple times. The former approach is referred to as a noniterative algorithm, and the latter is often referred to as an iterative algorithm. Furthermore, the objective of each local regulator can be either the minimization of a local performance index or a global cost function. The former approach is referred to as independent algorithms (e.g., [20], [21]), and the latter is often referred to as cooperating algorithms (e.g., [22]). Ref. [22] and [23] have shown that iterative and independent algorithms drive the system towards a Nash equilibrium, while iterative and cooperating methods approximate some Pareto optimal solutions under an ideal centralized control structure.

It is worth noting that distributed control has also been used for the coordination of totally independent systems in order to achieve a common target and to deal with joint constraints (e.g., [24]).

1.1.3 Hierarchical control

The use of hierarchical control structures, sometimes based on the model predictive control (MPC), is an efficient alternative to the aforementioned decentralized and distributed control architecture. Under hierarchical control schemes, algorithms at the higher level coordinate the actions of local regulators placed at a lower level, as illustrated in Figure 3. Roughly speaking, the higher-level coordinator computes the "prices" (Lagrange multipliers) of some "coherent" constraints in the global optimization problem given the state, input, and output variables defined by the local regulators, and the local regulators recompute the optimal trajectories of the state, input, and output variables iteratively [25]. Ref. [26] discussed coordination schemes for discrete-time systems in the context of MPC, and applied to transportation networks in [27].

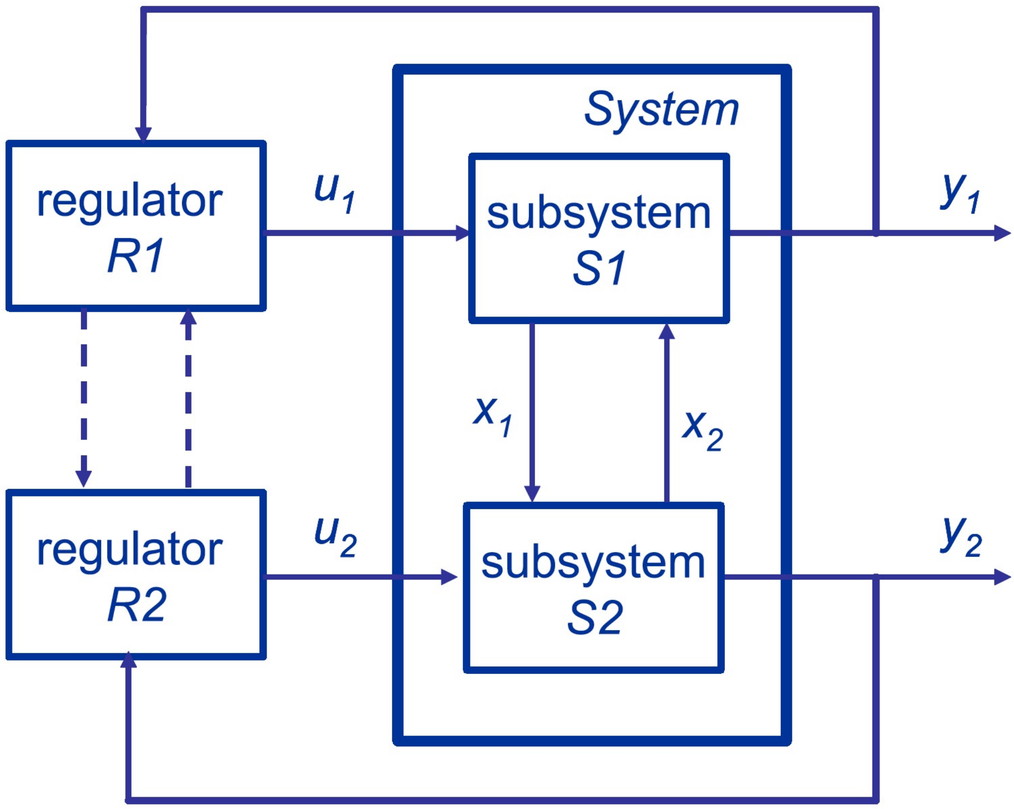


Figure 2: Distributed control of a two input (u_1, u_2)–two output (y_1, y_2) system (adapted from [14]).

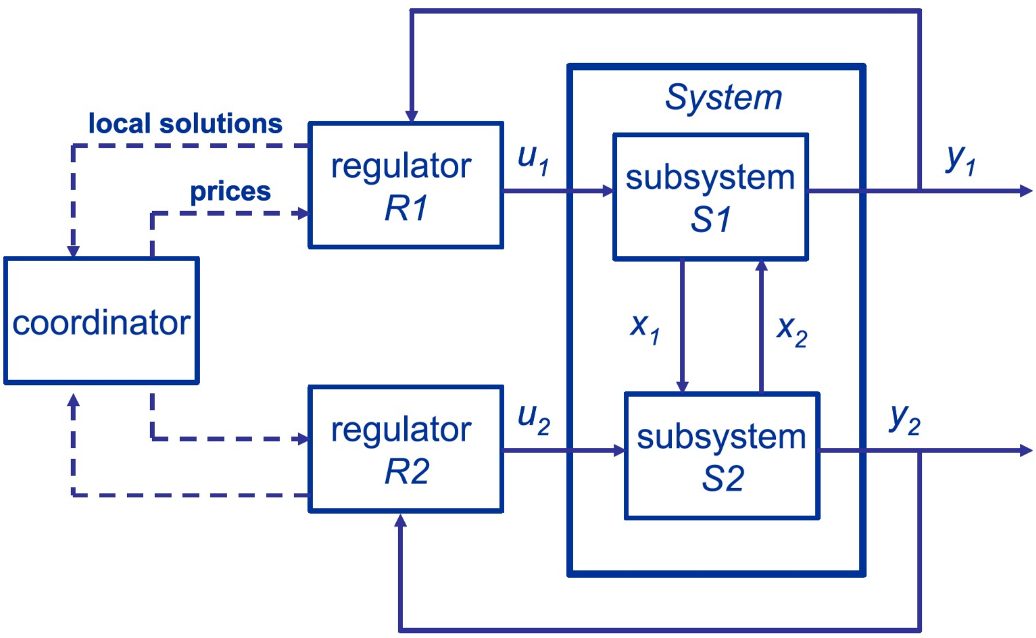


Figure 3: Hierarchy control of a two input (u_1, u_2)–two output (y_1, y_2) system. Figure Adapted from [14].

Other types of hierarchical control schemes appear in hierarchical multilayer systems, in which multiple controllers work at different time scales. Such controllers can be used in two major cases.

In the first case, the overall process under control is characterized by different dynamic behaviors, i.e., by slow and fast dynamics. As depicted in Figure 4, one regulator acts at a lower frequency, computing the control action with a long-term effect on the system, while the other regulator computes the control variables for the tracking problem at a higher frequency. In other words, the regulator at a higher layer computes its desired control inputs, which are the reference signals of the immediately lower layer. For example, [28] described the systems at any layer with a linear model, where information is passed bottom-up to relax the requirements of the higher layer when infeasibility occurs at the lower layer.

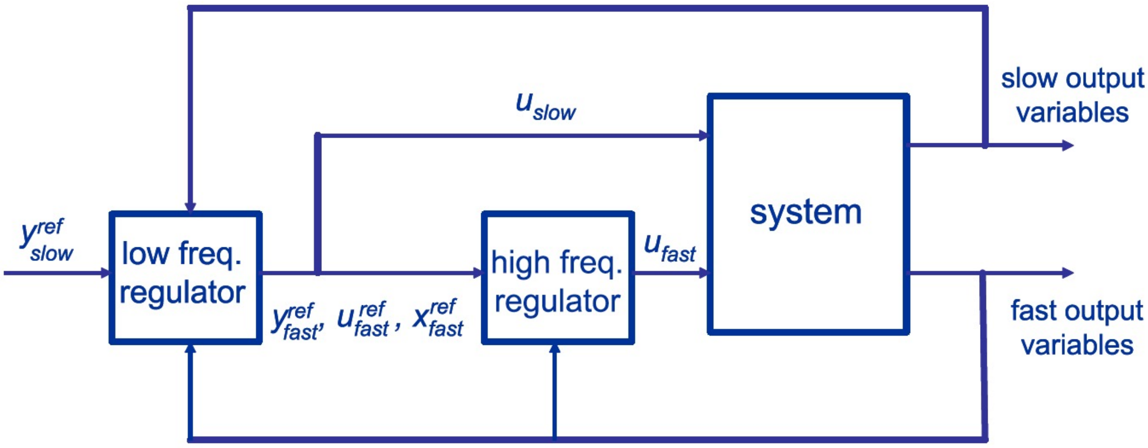


Figure 4: Control of a system with slow and fast dynamics. Figure Adapted from [14].

In the second case, optimization and control algorithms working at a different rate compute both the optimal targets and the effective control actions to be applied. At the higher level of a detailed and accurate system model, real-time optimization (RTO) is performed to compute operating conditions regarding a performance index of economic criterion. At the lower level, a simplified dynamic model of the system would be adopted (see e.g., [29]). In contrast, [30] considered a simpler and more abstract model at the higher level and a more accurate model at the lower level. To guarantee that the input and output steady-state references computed by RTO are feasible, accurate steady-state target optimization must be done. For example, [31] proposed an RTO procedure based on a dynamic model of the process. Ref. [32] mixed the two layers to integrate nonlinear steady-state optimization and linear MPC control.

1.2 Scope of the deliverable

This reports presents the potential benefits of distributed systems over centralized ones while designing the proper distributed architectures and information exchange requirements for different DiT4Tram subsystems. The work is the first part of WP1, which aims to develop an inclusive, generic, scalable, and adaptable paradigm for distributed management of transport systems and services for enhancing mobility in urban traffic. The paradigm is centered on cyber-physical systems using the IoT and other data sources to measure and represent the state of the system in real time. Smart components are assumed to respond to information in an adaptive way and are capable of learning.

This report investigates when decentralized systems tend to perform poorly compared to the performance under optimal centralized control, which is often termed the Price of Anarchy. In a stylized model, we show that it is possible that distributed systems with suitable information exchange and feedback/incentive systems can actually perform optimally or close to optimal with the methodologies developed in DIT4TraM. And neglecting the effect of subsystem interactions and uncertainties due to humans in the loop, can result in an undesirable evolution of the system state.

1.3 Links with other work packages

Task 1.1 includes concepts and architectures of passenger-oriented distributed control (Task 3.1), auctioning schemes for multimodal mobility services (Task 4.2), cooperative schemes for local bottleneck control (Task 2.4) and finally the traffic management platforms (WP6). A summary of the connected tasks between this deliverable and other work packages is provided as follows.

1.4 Structure of the deliverable

In this report, Chapter 2 presents the general conceptualization of decentralization in transportation and discusses the price of anarchy. Chapter 3 introduces some distributed control architectures for different applications and modes in transportation systems. Finally, Chapter 4 includes the conclusions.

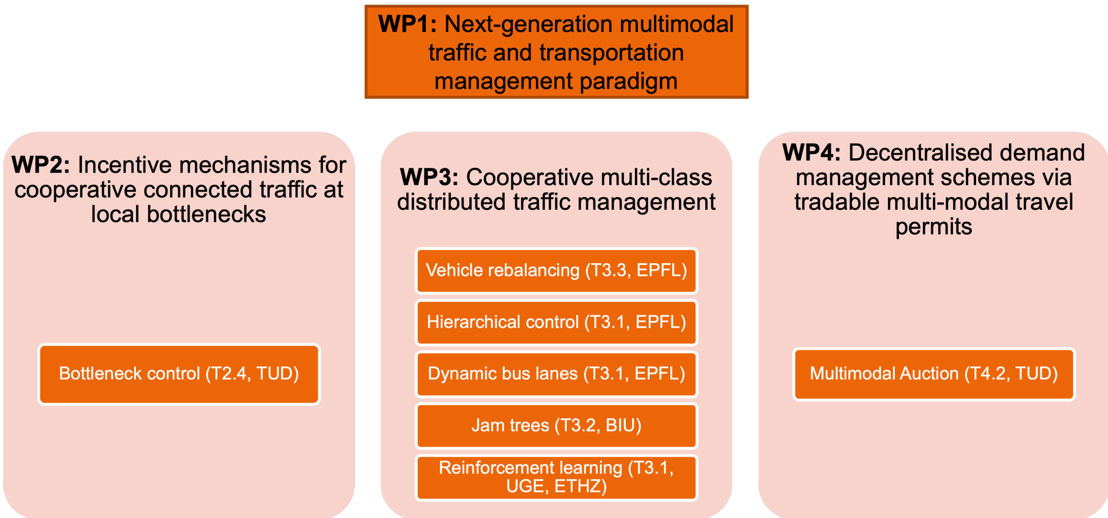


Figure 5: Summary of the connected tasks between this deliverable and other work packages.

2 Decentralization in transportation

In this chapter, we discuss the benefits of using decentralized, distributed systems in transportation (Section 2.1), and the price of anarchy under decentralized systems (Section 2.2). The results in the section are contributed by ETHZ.

2.1 Benefit of distributed systems in transportation

While a centralized system should, in theory, converge to the optimal solution, for much larger networks a centralized optimization technique is an NP-hard problem [33]. Using decentralized systems in transportation can offer benefits in terms of scalability, resource efficiency, and privacy preservation over a centralized system. This approach is also inherently suited for managing and modeling transportation systems [34], and lends itself well to data- and simulation-driven methods such as machine learning.

In the context of traffic signal control, distributed systems allow for a scalable design of systems by using traffic control systems that focus on smaller areas. This also drastically reduces the necessary state variable representation as opposed to simultaneous (centralized) control of multiple intersections, allowing for the reuse of these systems with minor tweaks and to be deployed to other locations.

Distributed systems also spread the workload of control across multiple subsystems, compared to a centralized system wherein a single entity is responsible for controlling and managing the vast systems involved in smart city systems. Distributed systems can help to ensure that transportation systems continue to operate, even in the event of a failure or outage. However, there remains the need for some level of coordination and information sharing between local systems [35] to ensure that each controller's actions do not conflict or interfere with other neighboring intersections [36].

2.1.1 Resilience under disruptions, accidents, or disasters

Resilience, or a system's ability to withstand, respond to, and recover from disruptions [37], is often evaluated through resilience curves. These curves show the evolution of a performance metric that maps system states to a scalar value throughout a scenario [38]. This provides insight into how a system responds to disruptions, such as the rate of performance degradation or recovery, the depth of impact, whether the system can restore its performance once the disruption is removed, and the speed at which this occurs.

In transportation systems, disruptions can take the form of sudden changes in traffic volume (demand-side) or changes in network topology (supply-side). In urban traffic systems, these two types of disruptions are strongly interlinked [39]. Transportation systems are vulnerable to various disruptions, such as natural disasters (e.g., earthquakes, floods [40]) or road works or accidents. Such disruptions hamper the performance of traffic control systems by either changing the traffic conditions to ones that the control algorithms are unsuited for, or in the case of centralized systems, having a single point of failure for an entire traffic network.

To achieve resilience, a decentralized system replaces the centralized controller with independent controllers distributed throughout the system. As individual controllers focus on their own local regions, this can limit the disruptive effects of failures on a global scale.

In recent years, machine learning approaches have gained popularity in the research of traffic signal control. Deep Reinforcement Learning approaches in particular have found their way into traffic control literature [41]–[45], with a common approach of employing multi-agent reinforcement learning (MARL) as a decentralized, distributed control paradigm. Oftentimes, these research works report improvements on the order of 10–15% improvement over non-reinforcement learning methods. We found however that a thorough treatment of such algorithms under disruptions was lacking; furthermore, there is a lack of a common performance baseline that can be used to compare results by different research groups. Whereas popular commercial traffic software (e.g., AIMSUN [46], VISSIM [47]) are also used by cities to develop and calibrate models, these are seldom used in most of the deep reinforcement learning literature[48]. Instead, much work on deep reinforcement learning is done on open source and extensible software such as SUMO [49] and CityFlow [50]. However, this means that researchers are unable to use traditional and established methods in traffic signal control (which are readily available in AIMSUN and VISSIM) as a tradeoff for more computationally efficient training of models.

Thus it is necessary for an extended evaluation framework for such algorithms to determine robustness in varying traffic conditions, such as disruptions in various parts of a traffic network due to accidents and/or disasters. Secondly, a common and easy-to-implement performance baseline must be established to make for fair, comparable, and interpretable reporting of algorithm performance across different groups. These are related to Task 1.3, and in the context of control algorithms, preliminary results show that indeed traffic control algorithms are sensitive to disruptions, which cause significant deviation from the idealized scenarios algorithms were designed/trained for (Figure 6).

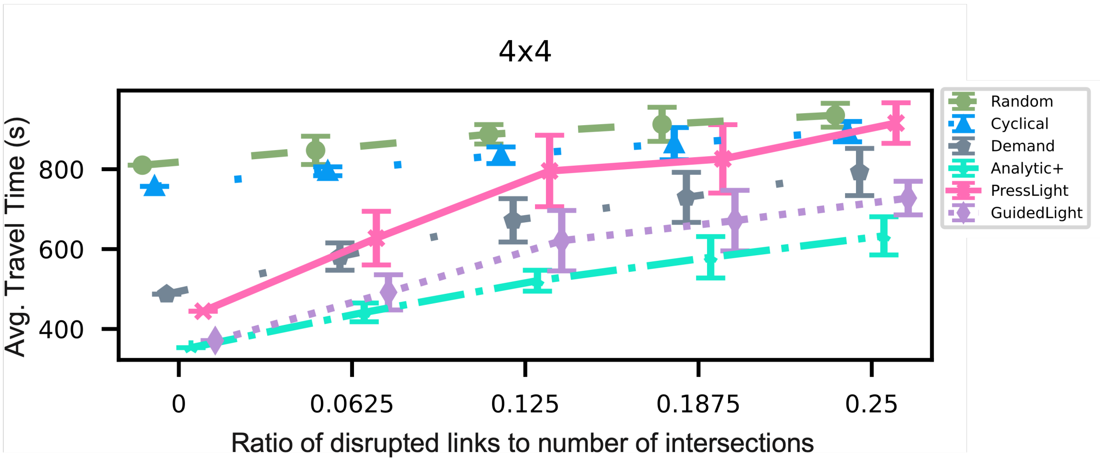


Figure 6: Results for the average travel times in some benchmark simulation experiments. Error bars indicate standard deviations of 100 disrupted scenarios simulated for each disruption level. Random and Cyclical are naive baselines, while Demand is a simple implementation of actuated control. Analytic+, and Presslight are deep reinforcement learning methods, while GuidedLight uses a self-organizing algorithm for traffic light control. Lower values are better. Figure reproduced from Ref. [51].

2.1.2 Privacy protection

The rise in the ubiquity of sensors, detectors, and connected devices raised concerns about the safety of storing data in centralized systems [52]. Intelligent transportation systems (ITS) integrate data from various sensors (e.g., loop detectors, connected autonomous vehicles, traffic cameras, and other smart sensors) to support multi-fold functions such as dynamic information services, traffic control, and vehicle management [53]. While data collected by traditional sensing techniques such as loop detectors use anonymized aggregate measures, other smart sensors may be storing and sharing raw data that may contain some identifiable information [54]. Decentralization can serve as a first step of protection against possible attempts of unauthorized access and misuse (by hackers, organizations, or even governments) of these data.

Decentralization reduces the amount of data controlled by a system at any given time, but this does not necessarily need to come at the cost of model performance. Federated learning [55] emerged in 2016 as a machine-learning setting where many agents collaboratively train models under the guidance of a central server. While training has some degree of centralization, data is kept decentralized, and the agents can still function independently of the central server. Thus, federated Learning places a strong emphasis on building effective models from user data, without compromising on the security and ownership of data from individuals [56].

2.2 The price of anarchy under decentralized control

One form of traffic control is to provide users with individual-level routing guidance, a service that drivers can avail of through apps like Google Maps, Apple Maps, and Waze, among others. As early as 1952, Wardrop suggested optimal ways for a central authority to distribute vehicles using a road network to achieve a *system optimum* (SO) [57]. Wardrop claimed that the lack of coordination led to inefficient transportation systems; this so-called notion of "anarchy" later developed into the concept of *Price of Anarchy* (PoA).

The PoA quantifies the inefficiency in decentralized systems, where agents act independently, compared to a centralized approach [58]. Specifically, it measures the ratio of the worst Nash Equilibrium [59] and the best possible coordinated solution. Networks with high PoAs are assumed to require centralized controls to achieve the best possible coordinated solutions.

While the navigation service providers are to an extent central points for getting guidance on route choices, their approach to route assignment is effectively a black box. This, together with the number of big players in the routing app market, has also resulted in routing apps worsening congestion [60]–[62].

While studies on PoA for simple, theoretical systems (e.g., Braess Paradox [63]) exist, determining the PoA for real networks analytically is difficult. More commonly, studies of the PoA are also restricted to using analytical cost functions (linear or polynomial) to estimate travel times on roads, however, such approaches fail to account for realistic driving behavior and queue spillback effects [64]. Ref. [65] uses a data-driven approach to estimate PoA by assuming real traffic dynamics correspond to the Nash equilibrium (also known as a user equilibrium or Wardrop equilibrium [66]). Other studies tackled reducing the PoA in transportation systems through the removal of roads [67], [68], or delegating routing decisions to a network of cooperating connected automated vehicles [69].

2.3 Braess Paradox with decentralized, learning agents

Routing and congestion games have been a very popular model used to understand network use [70], [71]. These games have important analogies to traffic on roads, and packets traveling over the internet, and have received widespread attention when seeking to optimize the use of such networks.

The Braess Paradox [63] (see Figure 7) is an example of a system with a counter-

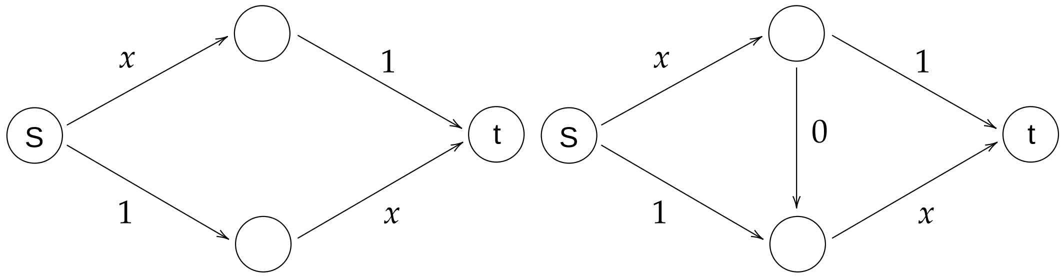


Figure 7: Illustration of the initial network (left), and the augmented network (right) in the Braess Paradox. Agents start in the "S" state and pick a path to reach state "t". The numbers represent the cost of traveling over a link. A cost of x is the ratio of agents that choose that link. Rational and fully-informed agents all pick the crossing link in the augmented network, which leads to high congestion and the worst possible social welfare. (left) Two actions are possible in the original network: up takes the upper edges, and $down$ takes the lower edges. (right) The augmented network allows for an additional action $cross$ is possible, which takes the first upper edge, crosses to the lower section at the middle, and finishes on the second lower edge.

intuitive property in that the addition of a road shifts the NE and worsens the congestion in the network. Conversely, the reverse holds: road removals can reduce congestion in the network. How can this be so? Braess Paradox is not a true paradox, because there is a clear explanation for the phenomenon that does not break any logical rules. Nonetheless, it characterizes a particular situation where independent and self-interested agents suffer without coordination.

This section demonstrates the benefits of a distributed approach by considering noise in the Braess Paradox (Figure 7) while retaining complete decentralization in the system. We achieve this by building on three different sets of agents that learn to play the Braess routing game in three ways: as non-learning and rational agents, irrational agents, and irrational agents that learn.

2.3.1 Non-learning agents

In the Braess paradox, we consider an initial (Figure 7a) and augmented (Figure 7b) network, where agents start from node "S" and travel to node "T". Each of the edges has associated costs (i.e., travel time) which are a function of the ratio of agents choosing an edge. In the initial network, equilibrium is reached when $x_{\text{up}} = x_{\text{down}} = 0.5$, with a total travel time of $T = 1 + x_{\text{up}} = 1.5$ for the upper path, and $T = x_{\text{down}} + 1 = 1.5$ for the lower path.

The addition of the crossing path in the augmented network breaks this equilibrium for the system. An agent originally in the upper path, acting entirely rationally, can decide to cross and have a travel time of $T_{\text{single\_crosser}} \approx x_{\text{up}} + x_{\text{down}} \approx 1$, which is faster than the travel time of 1.5 units in the initial network. This then leads to a domino effect, where all agents on the upper path decide to cross, and all agents on the lower path switch to the upper path and then cross. The Nash equilibrium in this system is the cost of the crossing path: $T_{\text{Nash}} = x_{\text{up}} + x_{\text{down}} = 1 + 1 = 2$. Thus, for the Braess paradox, we can compute the price of anarchy as

$$\text{POA} = \frac{T_{\text{Nash}}}{T_{\text{optimal}}} \tag{1}$$

$$= \frac{2}{1.5}. \tag{2}$$

2.3.2 Irrational ϵ-greedy agents on the Braess network

Braess Paradox is a routing game with a high Price of Anarchy, where decisions of rational agents converge to a Nash Equilibrium with the worst possible outcome. However, the Braess paradox outcome assumes that agents are fully-rational and have perfect information. **Fully-rational** agents have set preferences (e.g., minimizing travel time) and make decisions to satisfy these preferences (e.g., choosing roads that have minimal travel time. **Perfectly-informed** agents know the outcomes of their actions, and additionally of all other agents present in the system. If we relax the fully-rational and perfectly-informed assumptions on agents by adding noise. For agents that use an ϵ-greedy strategy,

$$\epsilon\text{-greedy route choice} = \begin{cases} \text{best believed route}, & \text{w.p. } 1-\epsilon \\ \text{random route}, & \text{w.p. } \epsilon, \end{cases} \tag{3}$$

results in different average travel times, depending on what agents believe to be the best possible routes. We found that for the crossing case, the travel time can be expressed as

$$\bar{T}(\epsilon) = \left(\frac{\epsilon}{3}\right)\left(2 - \frac{\epsilon}{3}\right) + \left(1 - \frac{2\epsilon}{3}\right)\left(2 - \frac{2\epsilon}{3}\right) + \left(\frac{\epsilon}{3}\right)\left(2 - \frac{\epsilon}{3}\right) \tag{4}$$

$$= 2 - 2\frac{\epsilon}{3} + \frac{2\epsilon^2}{9} \tag{5}$$

For values of $\epsilon \in [0, 1]$, the function $\bar{T}(\epsilon)$ is monotonically decreasing. Thus, for irrational agents that do not learn, the best travel time and system optimum is achieved for a value of $\epsilon = 1$, or when agents choose completely at random.

2.3.3 Learning agents

A more interesting case is to use the ϵ-greedy agents previously described, and allow them to learn open-ended. To model agents' beliefs, we use Q-learning and model the process of agents choosing paths on the Braess network as a *Markov De-*

cision Process (MDP). An MDP is a tuple of $(\mathcal{A}, \mathcal{S}, T, R)$ where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is a set of actions, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a transition function mapping state-action-state tuples to probabilities, and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function mapping state-action pairs to rewards. The goal of maximizing the cumulative reward function allows an agent to learn the appropriate action a to take, given a certain state s [72].

Q-learning [73] uses a function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ to map state and action pairs to the reward space. This equation estimates the *quality* of the current state from the perspective of the expected rewards for possible future states. The Q-values are then iteratively updated using the equation

$$\hat{Q}(s, a) \leftarrow (1 - l)\hat{Q}(s, a) + l \left[r + \gamma \max_{a'} \hat{Q}\left(s', a'\right) \right], \tag{6}$$

where l is a learning rate. The discount factor γ controls the importance of immediate compared to future rewards. Finally, the agent makes use of a policy π to choose actions based on the currently observed states and beliefs about the system. It is common to see the ϵ-greedy policy used to select actions in Q-learning:

$$\pi(s) = \begin{cases} \text{argmax}_a Q(s, a), & \text{w.p. } 1 - \epsilon \\ U[\mathcal{A}|s], & \text{w.p. } \epsilon, \end{cases} \tag{7}$$

where $U[\mathcal{A}|s]$ is a uniform distribution over the action set permissible at state s.

We then simulated $\mathcal{N} = 100$ agents, assuming all agents have the same set of parameters (ϵ, l, γ) and can perform one of three actions: $\mathcal{A} = \{\text{up, down, cross}\} \equiv \{u, d, c\}$. For simplicity, the agents do not take in state information, and we set $\gamma = 0$. The agents prefer minimum travel times, so the rewards are set to the negative of travel times in the Braess game.

Using ϵ-greedy Q-learning, we show uncoordinated agents' collective behavior lowers the Price of Anarchy of the Braess Paradox by increasing their exploration rates ϵ (Figure 8 [74]). While the initialization scheme of beliefs matters in the regime of $\epsilon < 0.11$, we find that learning agents eventually result in the price of anarchy values corresponding to the average travel times in the case of irrational agents. This result draws a conceptual link between irrational behavior and exploration during learning. Since time travelled is the metric of interest, we assume that a rational agent learns to pick the route they believe minimizes travel time.

Conversely, an irrational agent may not always pick the route that minimizes travel time. The ϵ value thus parametrizes the irrationality of an agent, where we equate irrational behavior with random behavior. However, we also extend the notion of irrationality to deviations from what would be considered rational behavior, which could be caused by external factors. These deviations could be due to errors in the measurements of travel times such that agents do not have perfect information, or

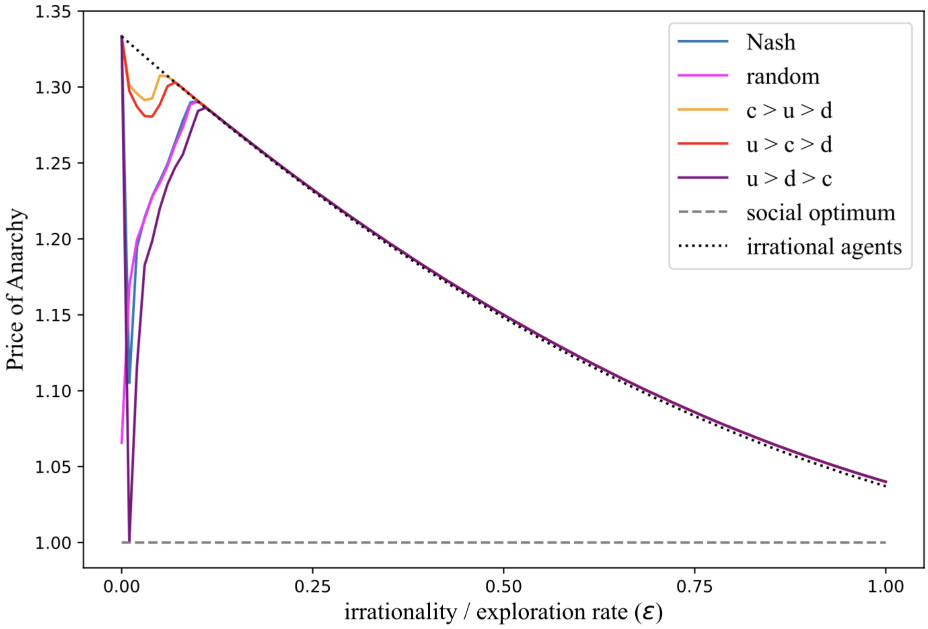


Figure 8: Plot of the Price of Anarchy (PoA) as a function of randomness. The irrational agents' curve shows how the best PoA is achieved for $\epsilon = 1$, where drivers pick fully randomly. The colored curves are empirical from experiments. A mismatch between the irrational agents' curve and the experimental curves is visible in the range $\epsilon \in [0, 0.11]$, while there is a close resemblance in the range $\epsilon \in [0.11, 1]$.

constraints unrelated to network structure that require individuals to pick routes with alternate criteria. There are many possible explanations for seemingly random behavior, which may not be irrational. Our simulations attempt to capture these deviations with randomness, and it appears that randomness, or irrational behavior, is beneficial in this setting. This benefit is accentuated by the Braess Paradox, where fully-rational behavior is expected to produce the worst possible outcome.

Viewing irrational behavior as beneficial for a system is somewhat in contrast to traditional economic theory, where all the benefits of efficient markets and equilibrium dynamics result from assumptions of fully-rational agents with perfect information. This paper thus contributes to an important direction of research that investigates the beneficial impacts that seemingly irrational behavior may have on the dynamics of systems with many agents.

3 Designing and implementing distributed systems in transportation

Motivated by the benefits of distributed approaches, we have developed distributed control architecture for different applications and modes in transportation systems. Multiple passenger-oriented distributed control strategies for large-scale urban systems are presented in Sections 3.1–3.3. In particular, Section 3.1 presents both distributed and hierarchical approaches for traffic signal control in general. Section 3.2 focuses on prioritizing public transportation systems by designing dynamic bus lanes, while Section 3.3 is concerned with the repositioning of on-demand transport. Section 3.4 presents auctioning schemes for multimodal mobility services, and Section 3.5 illustrates cooperative schemes for local bottleneck control.

3.1 Traffic signal control architectures

3.1.1 Distributed approaches

This section contains two different types of distributed traffic signal control approaches based on max pressure control and reinforcement learning, respectively. Max pressure control is designed to maximize the network throughput with nice theoretical foundation but relying on simplified assumptions regarding traffic condition, which are not necessary met in the real world. In contrast, the reinforcemnet learning approach provides a data-driven counterpart. This section provides a preliminary conceptualization of the above two control methods. More details about the integration of the control architecture in a network implementation will be provided in Deliverable 3.1.

A. Max pressure control

Max Pressure (MP) is a state-of-the-art distributed feedback-based controller proposed for isolated traffic intersections. Its function is based on a simple algorithm that adapts the right-of-way assignment between competing traffic signal phases in real-time, according to feedback information of the forming queues around the intersection (upstream and downstream), in a periodic cyclic process. In the core of MP lies a pressure component, which quantifies the weighted queue difference between upstream and downstream links of every phase in real-time. The pressure reflects a need for green time in each phase, by taking into consideration the capacity of the relative approaches in terms of flow and the existing queues upstream and downstream in relation to the capacity of the link. Based on the pressure values, a subsequent action is taken, which may be a phase activation of the maximum pressure phase or proportionate green time allocation between phases during the following control time slot. The formula for pressure calculation of a specific incoming link of the node for MP control can be similar to the following:

$$p_z(k_c) = \left[\frac{x_z(k_c)}{c_z} - \sum_{w \in O_n} \frac{\beta_{z,w} x_w(k_c)}{c_w} \right] S_z, \quad z \in I_n \tag{8}$$

In eq. 8, the pressure $p_z(k_c)$ of the incoming link z for the control segment k_c is calculated based on the queue length x_z of link z, the queue lengths of all downstream links w, x_w, the storage capacity c_i of each referenced link i, the saturation flow S_z of link z and the turn ratios $\beta_{z,w}$ between z and all downstream links w, which divide the queue of z to the intended next link in their path. The pressure of each phase is based on the pressure of all incoming links that take ROW during the phase.

Initially proposed for packet scheduling in wireless communication networks by [75], MP was formulated as a signalized intersection controller through the works of [76]–[78] and was theoretically proven by [76] to stabilize queues and maximize network throughput for a controllable demand, based on some specific assumptions, such as point-queues with unlimited capacity and separate queues for all approaches. The theoretical stability proof, despite the constraining assumptions on which it was based, together with the element of independence from any demand knowledge, and the decentralized and scalable layout, render MP a promising control strategy, especially for signalized networks facing unstable, excessive, or dynamic rapidly-changing congestion. Several research studies demonstrate the benefits of MP control application in specific case studies, mostly in terms of throughput and travel time improvement. Refs. [79] and [80] introduced normalized queues in the pressure calculation, thus implicitly taking into account link size and spill-back probability, while taking into account queue capacity, which was considered infinite in the initial MP of [76]. Ref. [81] presented an MP version with unknown turn ratios but existing loop detectors for all directions at the exit line. Ref. [82] and [83] proposed extended MP versions able to address bounded queue length estimation errors and incorporate online turn ratio estimation, as well as dynamically update control settings over space according to demand. Ref. [84] applied a strict cyclic phase policy, in contrast to phase activation based on pressure which can induce long waiting time for some drivers, and provided stability proof, which applied also for non-biased turn ratios. Ref. [85] integrated rerouting of vehicles in MP algorithms. Ref. [86] proposed a position-weighted back pressure control, on the basis of macroscopic traffic flow theory, and integrated spatial distribution of vehicle queues in pressure calculation. Ref. [87] proposed a delay-based version of the MP controller in order to increase equity of waiting time among drivers around the intersection. In [88], pressure is calculated by using travel time estimation instead of queue length, in an attempt of relaxing the need for expensive queue measuring equipment, and findings are supported by simulation and real field experiments. Ref. [89] proposed an alternative signal cycle structure with maximum cycle length.

On the other side, even though many studies provide stability and throughput maximization proofs of various MP variants, they usually refer to moderate and feasible demand sets, while MP performance in highly congested networks is questionable. Unstable behavior in such conditions can be attributed to the latency of local controllers, in general, in reacting to the rapid forming of congestion, given the lack of knowledge about traffic conditions upstream and out of the proximity of the controlled intersection. In the case of MP, there is little area for improvement if most (or all) controlled queues are saturated, in which case control tends to approximate the fixed-time plan. However, MP impact may still be significant even in over-saturated networks if they are combined with other congestion-preventing control strategies, e.g., in a hierarchical framework.

B. Reinforcement learning approaches

For large-scale traffic signal coordination, centralized control models may have some problems with the computational dimension, which increases exponentially with the number of signals and control periods. Although these models are efficient for offline signal control based on daily traffic patterns, their computation time is not appropriate for real-time applications [90]. Reinforcement learning (RL) is a type of machine learning that involves training agents to make decisions in complex, uncertain environments. In the context of traffic signal control, reinforcement learning algorithms can be used to optimize the timing or order of traffic signals in order to reduce congestion and improve the flow of traffic. Ideally, RL learns from experience and data to adapt to changing traffic patterns to achieve the goal of maximizing an objective function. Although there still exist some implementation challenges for RL, such as the policy immigration from the simulation platform to the real environment, RP provides an alternative way of data-driven traffic control with reduced modeling effort.

In recent years, further developments in ML methods have given rise to reinforcement learning (RL) approaches [41]–[45]. The RL methods can reduce the complexity of finding the optimal solution. However, they are not free from the dimensionality problem of the state-action space with centralized formulations. Thus, decentralized control methods like Multi-Agent Reinforcement Learning (MARL) control have been commonly used to overcome this shortcoming [91]. MARL divides the network into several subsystems, considered as agents. When considering large-scale real-time signal control, the multi-agent/decentralized framework and Deep Reinforcement Learning (DRL) are systematically integrated to simplify the state-action space of agents by approximating the value function with Artificial Neural Networks (ANNs) [92], [93].

UGE contributes by proposing a deep MARL framework for traffic signal control, considering car traffic and bus transit simultaneously. Under this framework, the traffic signal control process for each intersection is represented by an agent that dynamically triggers one of the predefined phases. The timing plan of the traffic signal needs to be evaluated and updated after each decision taken by the agent controlling that traffic signal. This interaction between the traffic light and traffic environment follows Markov Decision Processes (MDPs). In MDPs, the agent estimates the value of each action in each state and selects the optimal one, which exactly describes the signal controller's action. Each agent acts in a shared environment to achieve a common individual goal. Since car traffic and bus transit are considered simultaneously here, the optimal signal timing plan should reduce the overall car traffic delay and the variance of bus headways in the long term. Connections among agents exist for better coordination.

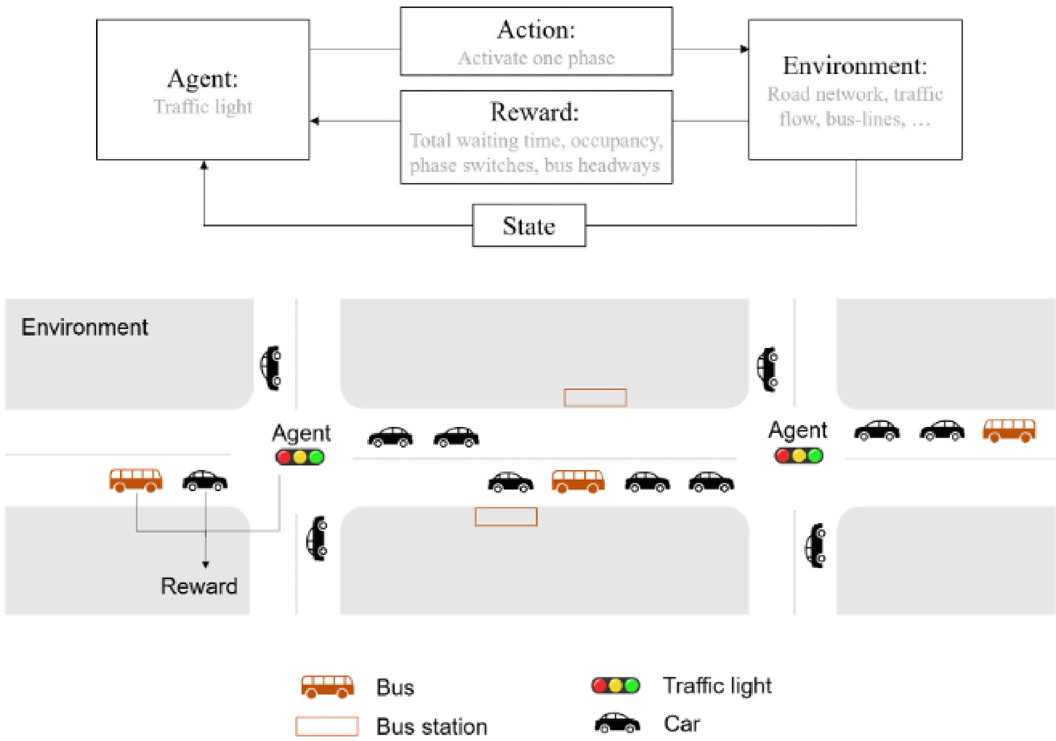


Figure 9: Multi-agent reinforcement learning framework for traffic network.

To build the traffic signal control approach into the MARL framework, it is essential to determine the critical elements (shown in Figure 9). The environment consists of intersections, the road network, car traffic flow, and bus transit. Each signalized intersection is regarded as an agent. The agent's action involves allocating the green to one of the predefined phases for the next decision step. In this framework, the state observations consist of real-time traffic information, bus headways, and the past actions of neighboring agents (i.e., neighboring signalized intersections). The same variables are used to define the reward function to enhance traffic efficiency at the intersection and to homogenize (keep constant) bus headways.

Compared to the centralized RL algorithm, the state-action space of MARL agents is reduced significantly, leading to better convergence performance for training models. According to several numerical tests, the MARL algorithm decreases the average queue length and the standard deviation of bus headway by 13.55% and 27.16%, respectively, compared to the best performance of the centralized control method and model-based adaptive methods. Both scalability and portability are demonstrated by transferring trained models to similar intersection configurations. Details about the proposed framework, agent design, and the test results are found in the deliverable of Task 3.1, as well as [92], [93].

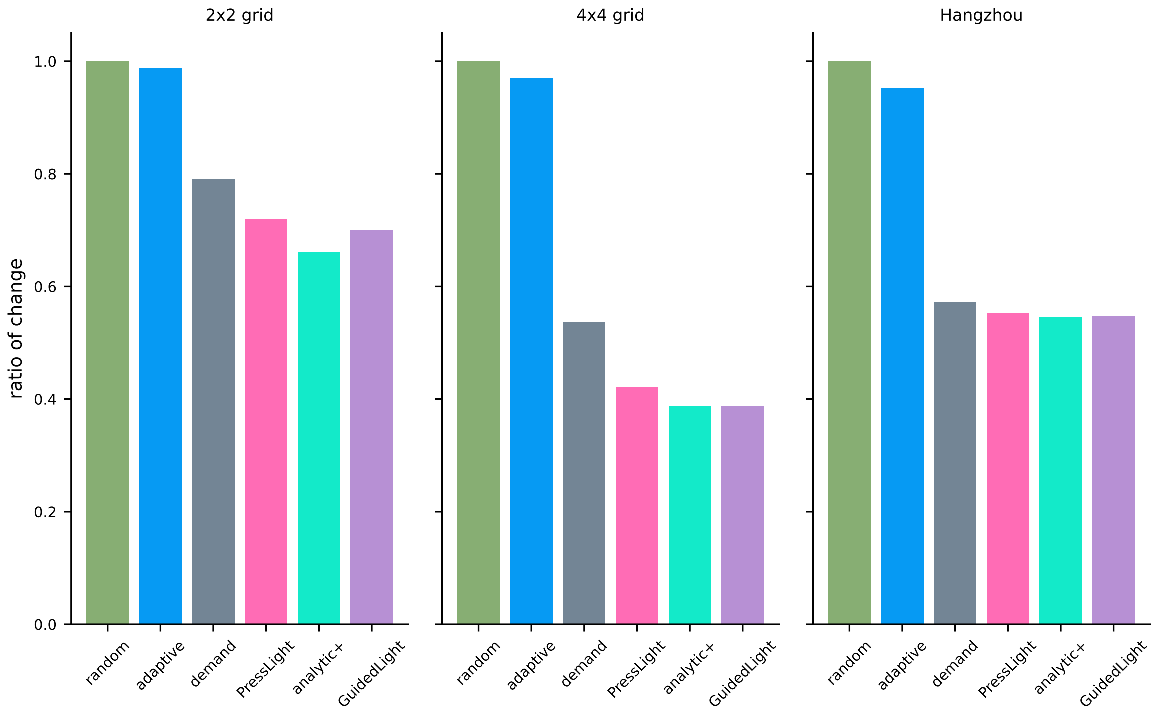


Figure 10: Average travel time of various traffic light control methods relative to the average travel time for fixed time schedules for two synthetic (2x2 and 4x4 grid) and one real-world (Hangzhou) scenarios. Lower values are better.

While deep RL approaches in literature boast impressive performance improvements, these benefits are underpinned by deep RL's tendency to overfit training scenarios [94], [95], and the large amount of computational power required to train such models.

ETHZ contributes by examining state-of-the-art techniques to reduce the training times in transportation networks in practice. In previous works of Korecki and Helbing, they combined these RL approaches with self-organizing control algorithms [96] to create **GuidedLight**, a *hybrid* approach that allows RL to learn from insights of an analytically-derived approach [44], [51]. This *hybrid* approach demonstrated by **GuidedLight** showed the capability of RL to learn traffic signal control schemes that reduce travel times of vehicles in the network (see Figure 10).

One of the challenges faced by RL approaches is the computationally intensive training times, which can be very long when dealing with much larger networks. While the distributed, multi-agent design can make training more sample efficient, learning from the highly varied traffic flows experienced by the many intersections in large networks is still slow. However, by employing tricks such as pre-training on a smaller, but richer environment for experiencing different traffic states, Korecki and Helbing were even able to reduce training times needed for these RL algorithms without sacrificing performance in managing network travel times [45]. These algorithms are currently being developed for Task 1.3.

3.1.2 Hierarchical approaches

This section provides a brief conceptualization of a hierarchical control scheme developed by EPFL, combining centralized perimeter control with distributed Max Pressure control. A combination of upper and lower layer schemes in highly congested environments helps the system to stabilize in more favorable and higher mobility states.

Additionally, we introduce the Jam-tree algorithm, developed by BIU, as a potential algorithm for identifying bottleneck areas and pinpointing key intersections for MP control.A detail introduction about the integration of the control architecture in a network implementation will be provided in Deliverables 3.1 and 3.2, respectively.

Max pressure and perimeter control

Developing multi-layer control policies comprising both local and aggregated strategies, seem an intuitive way of achieving network control with multiple objectives. Different layers may have different control mechanisms, act on different scales (local node/link or network) and take decisions that can benefit multiple layers. A two-layer hierarchical adaptive signal control framework for a network-wide application, combining centralized perimeter control (PC) with distributed Max Pressure control (see Section 3.1.1) is proposed here. The motivation for such a control system lies in the potential collaborative effect of these controllers, which relates to their characteristics. Among centralized approaches, perimeter control (PC) based on the concept of the Macroscopic Fundamental Diagram (MFD), has been effective in improving traffic performance of single or multiple neighborhood-sized, homogeneously congested regions. It consists of regulating inter-regional incoming and outgoing flows by adjusting the green light duration of the respective approaches at the intersections located on the network perimeter or at the boundaries between regions. The objective is to maintain maximum travel production within the protected regions, according to the specific MFD law that associates regional vehicle accumulation and travel production. Perimeter control has been intensively studied on the basis of MFD modeling, and a large number of MFD-based

PC schemes have been proposed, analyzed, and evaluated in recent years, utilizing different modeling and control methods and focusing on different control aspects. As a reference, Proportional-Integral (PI) feedback regulator for single- and multi-regional networks is implemented in [97], [98], [99], [100], optimal MPC is implemented in [101], [102], [103] with boundary queue consideration, while route guidance is incorporated in PC schemes in [104], [105].

Combining PC strategy with MP local regulators is expected to benefit network-wide signal control as follows. PC is shown effective in preventing high-demand regions from reaching very congested states. However, PC is more effective when traffic is homogeneously distributed in the network. On the other side, MP is known to be effective mainly in low to moderate congestion by increasing throughput and decreasing queue variance around intersections, thus increasing traffic homogeneity locally. However, its performance is questionable in oversaturated conditions, where it may approximate fixed-time control. Combining both controllers in a two-layer framework can benefit the global performance of the network since MP can help in maintaining homogeneity while PC can help avoid oversaturated states.

A question that arises regarding the architecture of the two-layer system refers to the location of the nodes where Max Pressure regulators should be installed across the network. The effects that MP might have on the PC control design parameters (such as critical accumulation of maximum network throughput) is also a topic to be investigated.

In detail, a schematic representation of the proposed two-layer controller is shown in Figure 11. In the upper layer, perimeter control is applied in an aggregated scale between a set of homogeneously congested regions and is activated when one or several regions reach a minimum congestion level. Gating can be applied on the external perimeter as well as on the boundaries between 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 and the external inflows for the next cycle, which are translated into the respective inter-regional average green times between every pair of adjacent regions and from the external perimeter to the interior of each region. The average green times are then translated to exact green times per approach of every PC controlled intersection, located on the regions' boundaries. Only a set of selected nodes is used for the boundary and perimeter control. The queues of the controlled nodes are used as weights for the green time distribution between nodes that belong to the same inter-regional approach. Perimeter control requires proper clustering of the network into homogeneous regions, where low-scatter MFDs can be identified. A simple way to implement PC is through a Proportional-Integral (PI) regulator, with a setpoint of target accumulations that consists of the critical accumulation of each region, as indicated by the respective MFD.

In the lower layer, distributed control based on the Max Pressure regulator is ap-

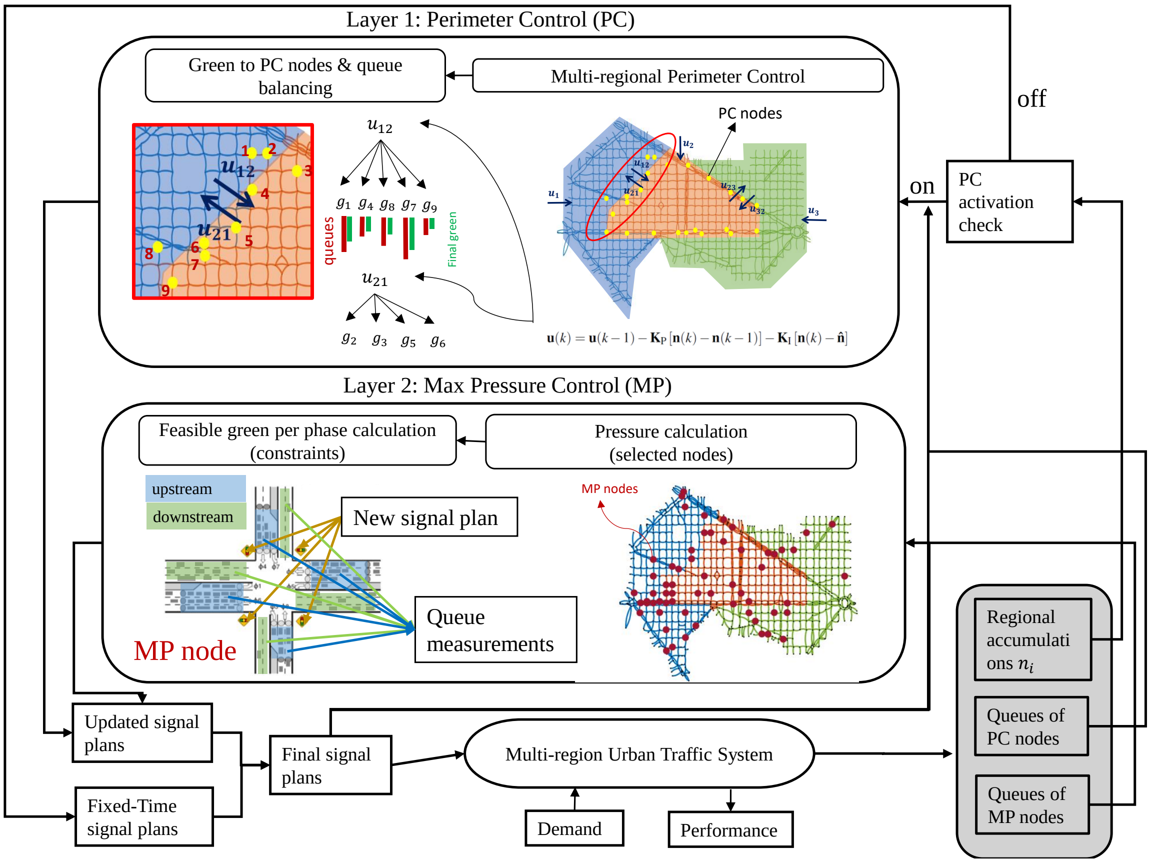


Figure 11: Structure of the proposed hierarchical control scheme combining perimeter control and Max Pressure control.

plied to a set of selected intersections, in the interior of the regions. This set can contain either all or a fraction of signalized intersections of the region, with the exception of those used for PC (if PC is applied in parallel). While many theoretical studies claim that MP should be applied to all intersections for network control, this practice leads to a significant implementation and maintenance cost related to the measuring equipment that needs to be installed on every road upstream and downstream of the intersection. It seems however that some nodes are more critical than others with respect to MP control and can make a significant difference in the global network performance if they get under MP control. Effective identification of the most critical locations for MP control installation can be based on various traffic information of the actual state and is on its own a topic to investigate. A selection method is proposed for identifying the most critical nodes for MP in combination with PC. The method is based on information about the mean and variance of queue lengths of the links around the controlled nodes, as well as on the spillback activity of any of the adjacent links. These three metrics are defined and calculated for every node based on peak-time traffic information. Then, they are linearly combined to create a metric expressing the overall importance of the node for MP control, by which nodes are ranked and selection is made starting from those with higher importance.

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 of 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 directly, however, their combined effect is indirectly considered by both controllers through the real-time traffic measurements that both layers receive as inputs. The described architecture of the two-layer framework is expected to be flexible in adjusting to different traffic conditions and congestion states. For instance, in cases of lower demand, regional accumulation would be lower than the threshold that would activate PC, so the MP controller would apply on its own to improve homogeneity locally and increase throughput in cases of local gridlocks without any hindrance. In cases of high demand, when PC is activated, transfer flows are controlled, keeping the congestion level close to critical, which allows MP to continue functioning without loss in performance due to saturation, while regional service capacity remains high. The queues formed around the boundaries and outside the external perimeter of the network due to gating are faster discharged with the help of MP.

Details about the function and parameters of the two-layer control framework as well as regarding the critical node selection strategy can be found in Deliverable 3.1.

Jam-tree approach

In addition to the critical node selection strategy being developed for Deliverable 3.1, we also introduce the Jam-tree approach. The Jam-tree approach was originally designed as a tool to study the dynamics and formation of congestion in different cities, but can also be used to identify key bottleneck areas in real time. More details about the integration of the jam-tree approach in a network implementation will be provided in Deliverable 3.2, respectively.

To identify traffic bottlenecks, we converted datasets of urban areas to dynamic, directed traffic networks where each node represents a junction, and each link represents a street segment between two junctions. The direction of the links represents the allowed traffic on that street segment, and the weight of the link at time segment t, $W(t)$, represents the temporal traffic relative speed, i.e., the ratio between the temporal speed and the speed at its maximal flow. We defined a street segment as currently congested if $W(t) < 1$. Next, we construct for a given time t a new dynamic weighted network, where $W'(t)$ is the sum of $W(t)$ of all times each link has been considered as congested up to time t (see Figure 12) and used the following process to create tree-shaped clusters of congested links:

1. At each time t, we identify the links with the highest weight W' (i.e., those that have been congested for the longest time) and define them as potential trunks of a jam-tree (JT). Next, we identify the branches of the JT by adding

links or other trunks, connected to each trunk, with $W' \leq W'_{trunk}$. By doing so, we identify links that became congested no more than a predefined parameter θ—in this case defined as 2 measurement units—after the trunk or after a neighboring road. The value of θ is only used to limit the connections of new branches to a JT; in other words, it reflects the maximal duration that a congested street segment is considered as the cause for the congestion in its upstream section. High values of θ allow a street segment to connect to its downstream section longer times after its downstream section became congested. This leads to larger JTs on one hand but reduces the probability of causality on the other. In other words, in our analysis, if a street segment became congested no more than 30 minutes after its trunk we can assume that the traffic load in these links resulted from the trunk of the JT. To test this assumption, we compared the result of the analyses of the real data to those of a controlled random model. The results of this comparison present a qualitative difference, which strengthens our assumption of causality. By using this definition, we consider the street segment that acts as the trunk as a bottleneck of the JT. Note, that we chose $\theta = 2$ as our datasets had 15 minutes time-intervals and thus, our analysis considered the macro-dynamics of urban traffic. Not that for other datasets with higher resolution of shorter time intervals, lower values of θ have been used.

2. We continue assigning connected links to these JTs in the same iterative process until no more connected links (roads) with $W' \leq \theta$ for the last added branches are found.
3. We start again at stage 1, but now we look for the link with the highest weight W', that has not been assigned to an existing (JT).
4. We continue this process until there are no more congested links that are not assigned to any JTs.

The resulting clusters represent JTs and the time each of their links was loaded. Examples of JTs are shown in Figure 12.

Stage #1: Economic Cost—Prioritization Strategy
While some traffic congestions can last many hours, their economic cost might be marginal if, for example, they occur in small peripheral streets. To assign prioritization for traffic congestions, we measure their cost in vehicle hours (VH). We introduce the four formulas to calculate the cost at different times of the JTs and the links they include.

The cost of a link $C_{ij}(t)$ is calculated for every measurement unit—15 minutes in this case—relative to its cost U_f, free-flow speed. This measurement unit demonstrates the meso-dynamics of urban traffic. Indeed, using shorter periods of time will allow for following the micro-dynamics of urban traffic. This cost represents the time it takes to cross a road (link) in comparison to the time it takes to cross

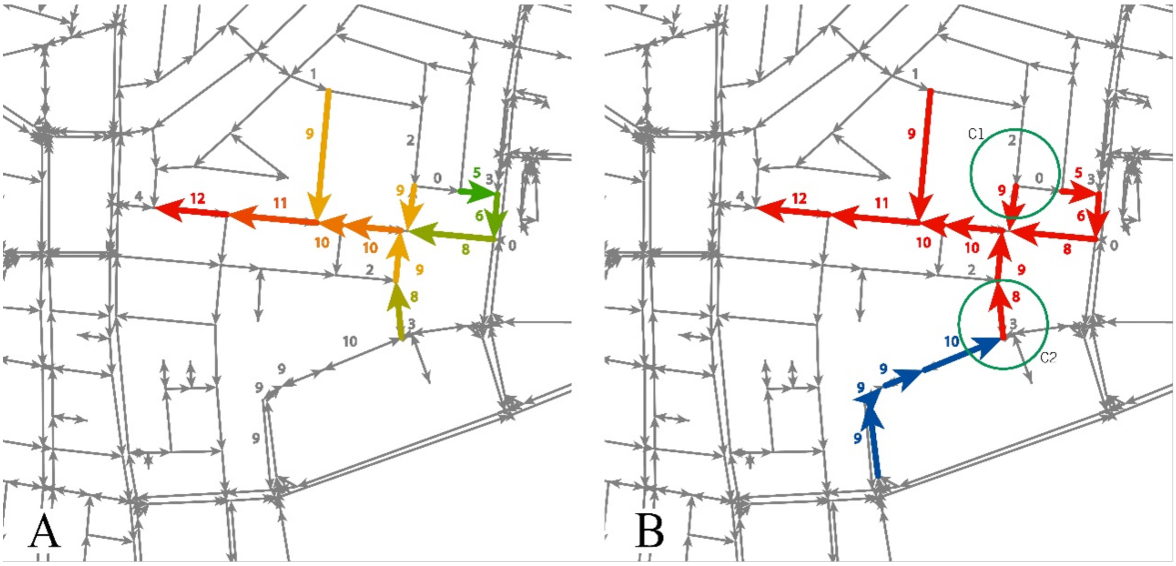


Figure 12: Clusters of Jam Trees (JT). The numbers represent the time (in 15 minutes units) each street segment was congested. (A) All the colored streets are part of one JT where the red street represents its trunk: its duration (12 successive measurements that represent 3 hours) is the longest, which indicates it was the first street with traffic load in this JT. (B) Two JTs (represented by red and blue colors). The red JT does not include the street that has been loaded for 2 measurements, as the time gap between this street and its adjacent one is larger than the pre-defined threshold θ (see the upper green circle). The blue JT cannot be considered as part of the red JT, as the duration of its trunk is longer than that of its adjacent street in the red JT (see the lower green circle). When a bottleneck is released but the JT that follows it remains congested, the next street segment with the longest duration becomes the new trunk of the JT and carries the cost of the remaining branches of the JT.

this road in its maximal flow (calculated for each link), multiplied by the number of drivers who crossed the endpoint of this link at a specific time:

$$C_{ij}(t) = dist_{ij} * \left(\frac{1}{u_{ij}(t)} - \frac{1}{u_{q\text{max}_{ij}}} \right) * \frac{q_{ij}(t) * l_{ij}}{\frac{60}{T}} \tag{9}$$

Here, $dist_{ij}$ is the length of the link in km, $q_{ij}(t)$ is the current flow on the link, $u_{ij}(t)$ is the current speed on the link, $u_{qo_{ij}}$ is the speed when the flow is optimal, T represents a measurement unit which corresponds to 15 minutes (in the present case) and l_{ij} is the number of lanes in the link (i.e., the number of lanes in each street segment of the JT).

The momentary cost of a JT represents the sum of the costs (Equation 10) of all the links that are included in it at a specific measured time:

$$MomentaryCost(t)_{JT} = \sum_{b_{ij}}^{n} (C_{ij}(t)) \,. \tag{10}$$

And the cumulative cost of a JT is the cost of the JT from the moment it was created until the time (t), which is calculated as:

$$CumulativeCost(t)_{JT} = \sum_{b_{ij}} \left(\sum_{t_I \leq t}^{t} c_{ij}\,(t_I) \right) . \tag{11}$$

Here, b_{ij} is a branch (i.e., link) in the JT, and t_I is the time (in units of 15 minutes) that each branch b_{ij} was a part of the JT.

Lastly, to follow the spatio-temporal dynamics of the system, we combine all the different JTs that had the same street as their trunk throughout the entire examined week and refer to them as Repetitive Jam Trees (RJT). The cumulative cost of the RJTs represents the sum of all the JTs they contain at a specific time window (e.g., day or week):

$$TotalCost_{RJT} = \sum TotalCost_{JT}. \tag{12}$$

Equations (10)–(12) allow calculating not only the cost of each JT from the moment it became congested until it was dissolved but also its dynamics and temporal costs at different times (see Figure 13).

Stage #2: Extended Definition of Jam Trees

We extend the definition of a jam-tree to include more general cases where several jam trees may overlap and share the same trunk or branches. This is highly sig-

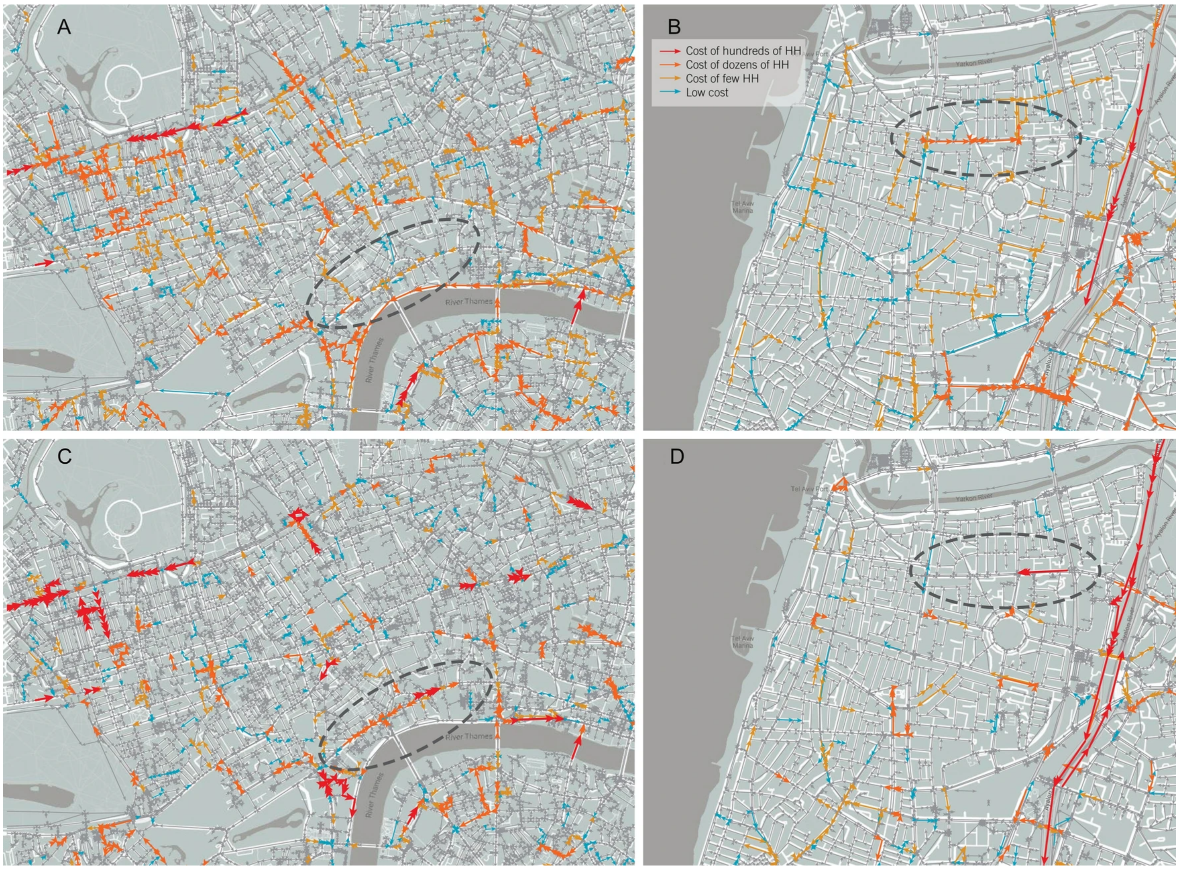
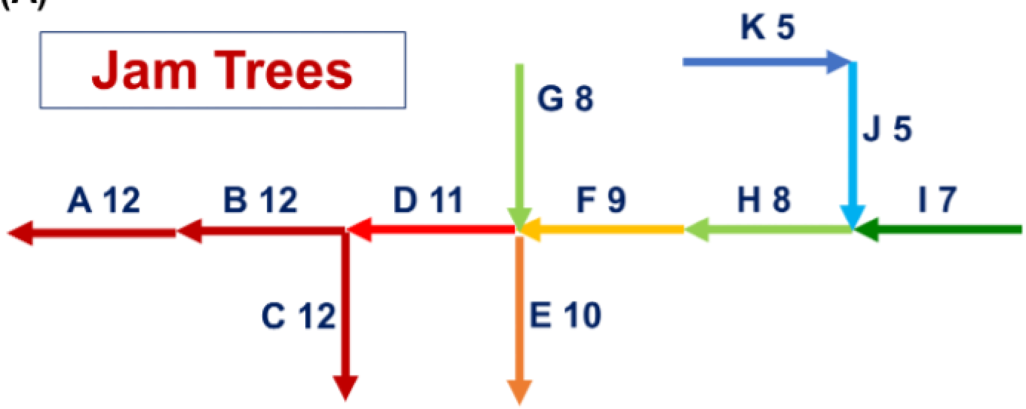


*Figure 13: Graphical representation of the JTs: snapshots of JTs in (**A**, **B**) London center and (**C**, **D**) Tel Aviv and the $CumulativeCost(t)_{JT}$ of the entire congestion, associated with each bottleneck, during morning rush hours (top row) and evening (bottom row). While some traffic congestions appear in both morning and afternoon rush hours (e.g., Marylebone street in London, or the Ayalon Highway in Tel Aviv), others are congested in only one of the rush hours. For example, Victoria Embankment road between Black friars Bridge and Waterloo Bridge (see black circle) is heavily congested only in the morning rush-hour snapshot; and Pinkas St. in Tel Aviv (see blue circle) is congested only in the evening rush hours snapshot). The maps were created using Snazzy Maps (`https://snazzymaps.com/help`), Rhino5 (`https://www.rhino3d.com/download/archive/rhino/5/latest/`), and Grasshoper plugin (`https://www.grasshopper3d.com/page/download-1`).*

nificant as these cases are very common, particularly in megacities with complex traffic patterns (Figure 14).

(A)



(B)

Tree	Trunk	Size	Cost
{A, B, D, F, H, I}	A	6	1A + 1B + ½D + ⅓F + ⅓H + ⅓I
{C, D, F, H, I}	C	5	1C + ½D + ⅓F + ⅓H + ⅓I
{E, G, F, H, I}	E	5	1E + 1G + ⅓F + ⅓H + ⅓I
{J, K}	J	2	1J+1K

Figure 14: Definition of a jam tree. (A) An example of several potential jam trees. Each directed link represents a road segment, with an arrow indicating the direction of traffic flow on it. The number above each link indicates the jam duration i.e., successive time intervals of traffic jam. Here, each time interval represents 10 minutes. If the difference in jam duration between two nearby segments is less than a threshold, (in this work 2 intervals that represent 20 minutes), they are considered part of the same "jam tree." This is based on the assumption that there is a causal relationship between upstream and downstream traffic flow. (B) Key information about the presented jam trees. The Table shows the information on the tree trunk, size, and cost of each specific jam tree in (A). Each jam tree has only one specific trunk based on the above definition. The size of a jam tree is determined by the number of segments that belong to it, including both the main trunk and any branches., while the temporal cost of each jam tree is defined as the weighted sum cost of its trunk and branches.

As can be seen in Figure 14, each letter (**A** to **K**) in the figure represents a road segment, with an arrow pointing in the direction of the traffic flow on that segment. The number above each link is the jam duration (time units) of the link at a given time. The table in Figure 14b, shows how we calculate the tree size and cost. The tree size is defined as the number of road segments that a jam tree contains. To calculate the cost of each road segment and sum them up, we followed stage #1. If a road segment can be associated with multiple trunks, e.g., link **F** in Figure 14, which is a shared branch of trunks **A**, **C**, and **E**, the cost of this segment is divided equally among all the trunks it is associated with, in this case, 1/3 of the cost would be assigned to each trunk. This is because each trunk has an equal probability to induce congestion.

Stage #3: Spatio-Temporal Evolution of Jam Trees
We expand the method of jam tree to the spatiotemporal Evolution of Jam Tree (EJT) that describes the evolving features of MTJs originating from the same traffic bottlenecks B to observe the evolution patterns of the traffic congestions. An EJT is expressed as

$$EJT = \{MJT(t_G), \cdots MJT(t_P), \cdots, MJT(t_R)\}, \tag{13}$$

where $MJT(t)$ is the set of congested roads caused by a bottleneck B at time t, where t_G, t_P, and t_R are respectively the emergence time, peak time, and dissolving time of a jam tree. The size of a jam tree at the entire lifecycle can be given by

$$\begin{aligned} S &= \{S(t_G), \cdots, S(t_P), \cdots, S(t_R)\} \\ &= \{|MJT(t_G)|, \cdots, |MJT(t_P)|, \cdots, |MJT(t_R)|\}, \end{aligned} \tag{14}$$

where size $S(t)$ is the number of the congestion roads included in the momentary jam tree at time t originated from bottleneck B.

The Jam-Tree algorithm serves as a tool for studying the dynamics and evolution of congestion in cities.

3.2 Dynamic bus lanes under distributed control

This section includes a distributed control of dynamic bus lanes envisioned by EPFL, which regulates the number of allowed vehicles in the bus lane according to real-time traffic information. Under the dynamic bus lanes control, a set of corridor links of the network can be set as links with dynamic priority lanes and their operation can be handled in a distributed way with controllers monitoring the car density at the mixed-use lanes and dynamic bus lanes. Similar to Section 3.1.2, such distributed control schemes can be further integrated with the perimeter control scheme in a hierarchical control manner. A detailed introduction about the integration of Dynamic bus lanes with perimeter control in a network implementation will be provided in Deliverable 3.1.

3.2.1 Preliminary

In the scope of alleviating congestion in urban networks, several strategies have focused on prioritizing public transportation systems in various ways, as an indirect way of reducing the number of cars competing for limited road space. A popular strategy refers to providing dedicated road space to buses, in order to separate them from slowly moving traffic in congested conditions. This concept applies in the cases of Dedicated Bus Lanes (DBL), where a lane is reserved for bus traffic only and is not available to private cars. Given that DBLs may result in a waste of space in case they are underutilized—mainly due to reduced bus circulation—research is focused on more flexible types of lanes, where some private cars are allowed in the lane when no buses are using them. This concept relies on the fact that mixed circulation of cars and buses does not affect bus speed in cases of low traffic. On this base, Bus Lanes with Intermittent Priority (BLIP) and Intermittent Bus Lanes (IBL) have been proposed [106], [107]. These lane types use transit signal priority to empty the lane from cars or force cars to change lanes prior to bus arrivals, respectively, while allowing cars inside in case of bus absence. A more versatile version of this concept is proposed by Anderson & Geroliminis known as Dynamic Bus Lanes [108]. These are lanes of mixed-use in principle, where private car entry can be controlled with the scope of protecting bus operations performance, according to the prevailing traffic conditions. The signaling regarding the dynamic lane status (open or closed to cars) can consist of pavement lights, special lane traffic lights, or variable message signs. Dynamic bus lanes can be seen as a middle solution between mixed traffic and dedicated bus lanes. This is because they provide uncongested road space to buses in case of increased congestion in the adjacent lanes, by decreasing car entry, while allowing some cars to enter as well. In this way, the negative effect of reserved road space on car traffic, which can lead to bottlenecks even in lower congestion, is reduced.

3.2.2 Dynamic bus lanes under distributed control

The idea behind Dynamic Bus Lanes is to regulate the number of allowed vehicles in the bus lane according to real-time traffic information, such as the location and speed of buses, the density and speed of cars in mixed-use lanes, and others. The dynamic feedback regulator should adjust the number of allowed cars in order to allow fewer when congestion is high and more in less congested conditions while taking into account performance measures of buses and cars. With such an operation, overall road space use can be increased leading to less traffic in mixed-use lanes during congestion peaks. Based on the results of [108], for a corridor road, dynamic adjustment of the fraction of cars riding in the priority lane is more profitable than fixed car fraction during rush hour, which opens the way for research in dynamic control of car entry rate in dynamic bus lanes as part of a distributed traffic-responsive network control system. A set of corridor links of the network can be set as links with dynamic priority lanes and their operation can be handled in a distributed way with controllers monitoring the car density at the mixed-use lanes and dynamic bus lanes.

The allowed car rate in a specific time window for any dynamic bus lane corridor can be the output of a simple controller in the form of a proportional regulator as follows:

$$p_t = p_{t-1} + K_m(k_{m,t} - k_{m,s}) - K_{DL}(k_{DL,t} - k_{DL,s}) \tag{15}$$

where p_t is the car rate of time step t, $k_{m,t}$ is the average car density over the mixed-use lanes of the corridor and $k_{DL,t}$ is the car density at the dynamic bus lanes at time step t, while $k_{m,s}$ and $k_{DL,s}$ are the respective set points (targets).

The control of car entry of dynamic lanes can also be integrated into a hierarchical control framework similar to the one described in Section 3.1.2, combined with a perimeter control layer, which will make sure that high demand areas will not reach very congested states. In this way, public transportation vehicles will be more protected from high congestion. However, measures should be taken to adjust the perimeter control implementation in a way that will facilitate the entry of buses in the protected areas (e.g. with pre-signals or dedicated bus lanes on the borders between regions. More details about the implementation of this multi-layer network control will be given in Deliverable 3.1.

3.3 Repositioning of on-demand transport

In this section, we demonstrate how the distributed control architecture could benefit on-demand transportation by presenting the rough idea of three different approaches to re-balance the temporal and spatial distribution of on-demand transport vehicles. The first two repositioning methods are contributed by EPFL, while the latter is by UGE. More details about integrating the vehicle repositioning methods in a network implementation are provided in Deliverable 3.3.

3.3.1 Method A

In implementing a controller for a large-scale system, one may face problems such as high computational effort due to complex models and high dimensions required for accurate network modeling, especially if the model and controller are developed to compute control actions for every individual vehicle over the whole network. One way to solve this problem is to build a hierarchical control structure. Such structures decompose the control problem into a hierarchy of decision-making levels and operate via coordinating between the actions of an upper layer controller (operating at the aggregated traffic level) and a lower layer controller (managing individual vehicles). The control structure is shown in Figure 15.

The upper layer controller collects aggregated information, such as how many empty vehicles are in each region, from all urban regions at a relatively large update period T_u. The control action generated from the upper layer determines how many vehi-

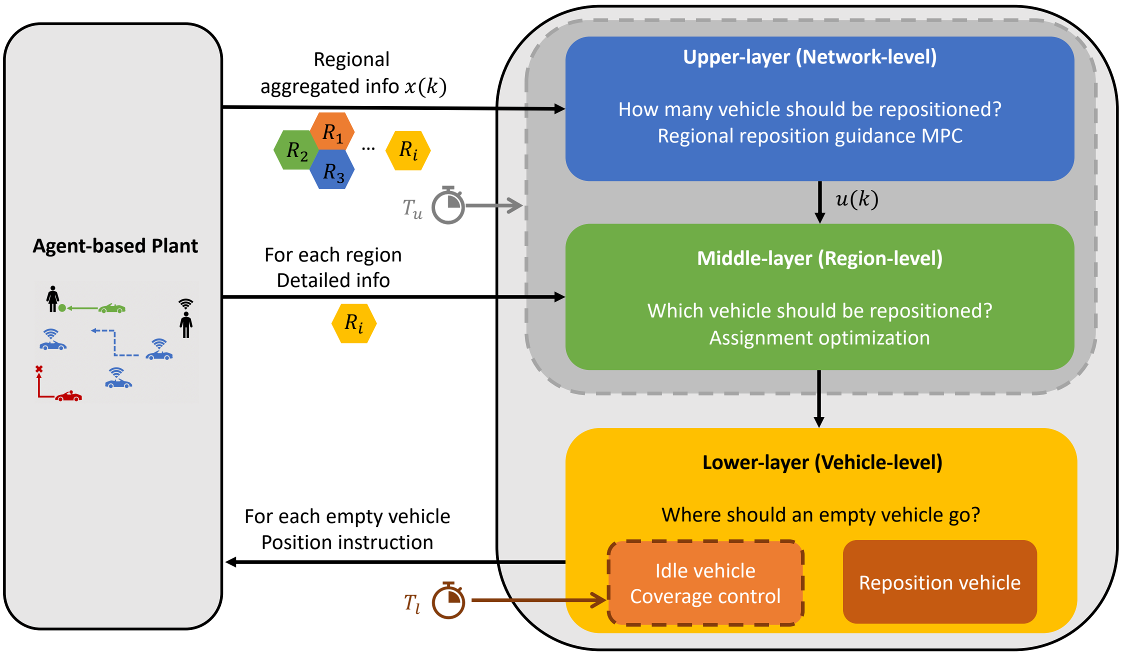


Figure 15: A hierarchical control framework for vehicle rebalancing.

cles should stay in current regions and how many vehicles should relocate to other regions, in order to improve availability and thus minimize the total waiting time of passengers. Furthermore, the middle layer transfers the obtained upper layer guidance to the lower layer and specifies which vehicle should stay or move, considering the travel costs caused by repositioning. It is operated within each region and requires relatively more detailed information, such as the coordinates of each vehicle and whether it is occupied or not. Note that the middle layer can only be activated when the upper layer is active. The lower layer is operated in a distributed manner so that each vehicle can obtain its own control action, which facilitates its implementation at a fast update period T_l. The empty vehicles that are commanded to stay in the current region (i.e., idle vehicles, see the left part of the lower layer in Figure 15) communicate and cooperate with each other to achieve better vehicle position configuration, while the rest of the vehicles (i.e., repositioning vehicles, see the right part of the lower layer in Figure 15) are be guided to other desired regions as per the relocation commands. Details about each one of the layers of this hierarchical strategy, with their mathematical formulations, are found in the deliverable for Task 3.3.

3.3.2 Method B

For a platform serving both solo and pool trips, the second proposed rebalancing strategy utilizes pool trips as a rebalancing tool. The operator can employ this strategy by extracting information on demand intensity and demand loss distributed in different areas of the network, namely any asymmetry in the origin and destina-

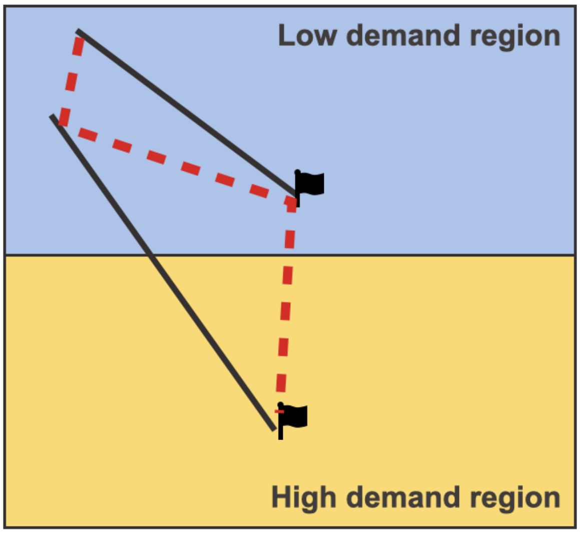


Figure 16: Illustration for a rebalancing candidate: black solid lines: solo trips with destinations at black flags; red dotted line: rebalancing pool trip

tion demand pattern, and recommend vehicles to serve a specific set of pool trips so that these drivers can drop off their final passenger in a high demand area, whereas without this rebalancing strategy, vehicles are likely to end up in low demand areas. An illustration is provided in Figure 16.

In other words, the in-service vehicle rebalancing strategy takes on a distributed approach to consider a number of nearby options and proposes one that best meets the rebalancing objectives. A pricing incentive is further proposed to encourage riders to comply with the rebalancing option that can better serve the overall heterogeneous demand in the long run.

In order for the platform to successfully meet its rebalancing objective, an appropriate monetary incentive should be offered to increase the attractiveness of the pool option. The reason is that pooled trips may be perceived by passengers as a less convenient option compared to an available solo alternative.

When a new rider places a request, given the respective origins and destinations of potential pooling, the platform identifies the respective zones as exhibiting high or low demand to assess the necessity of rebalancing actions. A rebalancing pool trip is loosely defined as one where the vehicle has its final destination in a zone with high demand loss, whereas if two separate solo trips were to be served, one vehicle would drop off its passenger in a low demand zone, hence contributing to the accumulation of empty vehicles in the area, unable to fulfill requests in high demand areas that are far away. For this purpose, the platform should aggregate ride-sourcing travel demand over a predefined number of region centroids according to the network partitioning, in turn allowing for a level of aggregation that highlights any spatial and temporal demand pattern.

The next step consists of setting the respective prices for the solo and pool trips. Besides a predetermined fare for each option, when a pool trip is considered to be desirable as a rebalancing trip, the platform suggests this trip to the rider and offers a slight discount on the trip price.

When traveler i is presented with one solo and one pool trip option, their choice probability is related to their perceived costs associated with each option. The proposed model draws a random number x_h^i with a uniform probability between 0 and 1, and compares this value with the perceived attractiveness of the pooling option $\overline{h^i}$, which represents the pooling attractiveness h_i normalized with an estimated lower and upper bound on its parameters. Define the attractiveness of pooling h^i for a rider i as

$$h^i = \alpha \cdot \nu^i \cdot (t_{travel}^{i,\, solo} + t_{wait}^{i,\, solo} - t_{travel}^{i,\, pool} + t_{wait}^{i,\, pool}) + c_{pool} + (p^{i,\, solo} - p^{i,\, pool}) \tag{16}$$

where, from left to right of Equation (16), α is a parameter tuned based on the lower and upper bounds of h^i, ν^i is the passenger's value of time in \$/hour; the next four terms are the travel and waiting times of the solo and pool options, respectively; c_{pool} is another tuned parameter that reflects the relative comfort of a pool trip compared to a solo one, and is identical across the population; finally, the final two terms together represent the price difference between the solo and pool option. As for the normalization of h^i for obtaining $\overline{h^i}$, this is performed by considering the lower and upper bounds for travel time, waiting time, and price, which ensures that the normalized values are mainly between 0 and 1. If $x_h^i \leq \overline{h^i}$ holds, the rider chooses pool, and otherwise solo; in other words, a higher $\overline{h^i}$ is related to a higher probability of acceptance for pooling. Both riders must accept pooling for a match to be successful.

Details on the network demand visualization and case study results can be found in Task 3.3.

3.3.3 Method C

We define a third fleet management strategy for on-demand mobility services, focusing specifically on the distribution of empty vehicles on the urban network. We design a decentralized decisional architecture consisting of a mesh of controllers that divide the urban network into as many service areas. These agents are considered to be at the service of a public authority (e.g., a transport agency or a local authority) and aim to satisfy the demand within their service area. They can be coupled with physical infrastructures, such as parking lots or depots for vacant vehicles.

To guarantee the fast pick-up of the local demand, the controllers are first in charge of predicting the number of travelers requesting a ride within the service area. This prediction can be based on local demand history and known pre-booked requests if any. This prediction allows controllers to estimate the number of vehicles needed to meet local demand and to implement a negotiation process with cars to attract the required number.

These negotiations between vehicles and controllers are done simultaneously and in a decentralized way through a two-sided matching market. Vehicles apply to their favorite relocation offer, *i.e.*, the offer that will maximize their expected revenue, while controllers aim at ensuring the fastest service for the local passengers. This reconciliation may require several iterations of the process, with vehicles that have been rejected by their preferred region applying for the next one. At the end of this process, vacant vehicles are assigned to a service area to which they will relocate. It is assumed at this point that vehicles comply with the outcome of this process. Figure 17 illustrates this communication protocol.

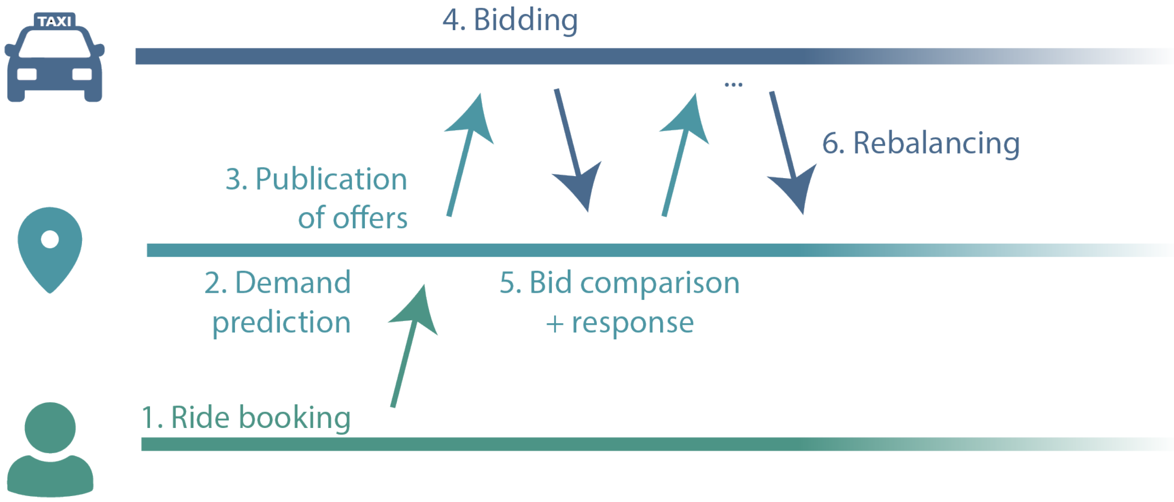


Figure 17: Communication protocol between travelers, controllers, and vehicles

We discuss the details of the method and algorithm in the Task 3.3 deliverable. Here we discuss the potential strengths of this method. Its main advantage is that the rebalancing control is decoupled from the internal management of the fleet by the on-demand service company and is entrusted to an external operator, whose agents issue the vehicles with authorizations to relocate within their perimeters. It is, therefore, a top-down control of the service and its vehicles by a regulatory authority. Three main advantages can be seen in this property.

First, by controlling the influx of vehicles into a zone and limiting it to what is strictly necessary, this architecture can have the advantage of limiting the contribution of service vehicles to the demand of the area and contribute to its congestion.

Second, this strategy can also limit the competition between drivers. By simply informing drivers of high-demand areas, services such as Uber encourage vehicles to reposition themselves massively, maintaining intense competition between drivers and lowering the ride price. In addition to limiting this vehicle accumulation, our method seems to guarantee a better distribution of activities and incomes among drivers.

Finally, although our studies have focused exclusively on a single-service context, externalizing fleet reallocation decision-making holds promise for studying more complex operating environments with competing mobility services. In this case, the network of local controllers will be able to interact simultaneously with the vehicles of different mobility services, mediating their competition and coordinating, to some extent, their fleets.

3.4 Auctioning schemes for multimodal mobility services

In this section, we present an auctioning scheme in the context of generic (multimodal) mobility services with heterogeneous VOTs and behavioral characteristics, aiming to enhance the utilization of intersection capacity. This approach is developed by TUD. More details about integrating the auctioning schemes in a network implementation are provided in Deliverable 4.2.

3.4.1 Conceptual framework

The introduction of connected vehicles across different modes has created considerable research efforts, in recent years, as to identify approaches for management that exploit this added layer of communication. When considering network intersections as a spatially complex set of scarce commodities, market-based instruments can be considered as a means to manage their use ensuring temporal segregation (i.e., ensuring that the resource is not used beyond its capacity) [109] [110] [111] [112] [113].

Aspects of market-inspired intersection control can be summarized in Figure 18:

In such approaches, traffic signal phases are assumed to be the auction participants, while unconstrained sequence transitioning between green phases is allowed to enhance the utilization of intersection capacity. The auction process determines which phase should become active next or whether the current green phase should be expected. Each vehicle within a time-dependent lane-specific communication range is considered as an agent bidding for green time. Under this configuration, transport users declare their VOT and behavioral characteristics, such as level of impatience, at the beginning of their trip so that the wallet agents can then bid on their behalf based on the time needed to cross the intersection and their current waiting time. Wallet agents are assumed to communicate bids to an intersection manager (e.g., roadside units (RSUs)) which executes the auction and assigns

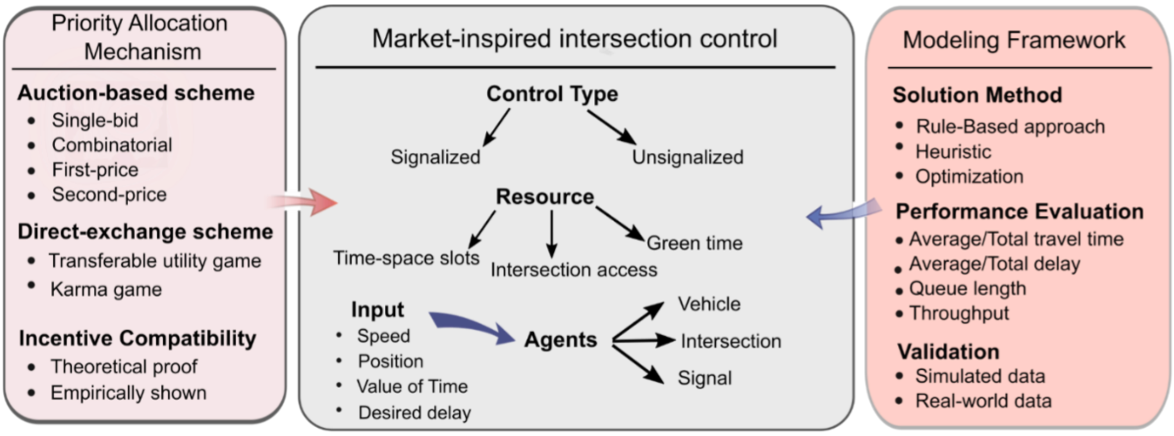


Figure 18: Market-inspired intersection control framework.

green time accordingly.

3.4.2 Bidding distance determination

Determining which vehicles can participate in the bidding process is crucial for the efficiency of traffic auction schemes. To constitute a fairer and more efficient practice, we aimed to incorporate the waiting time for each lane within bidding distance determination. Under this context, we proposed an online scheme where the bidding distance dynamically varies depending on the average waiting time for each lane. In this way, delay accumulation is accounted for in a dynamic manner. More specifically, at each time step t when an auction is triggered, an initial bidding distance factor z_l^t is determined based on the average waiting time for each lane l with a red light indication. Subsequently, the proposed reservation distance per lane d_l^t is calculated as the sum of a fixed component of the default bidding distance d^b and a variable component depending on the current waiting time observed at the specific upstream link. In order to avoid strategic misreporting, this is not reported by the drivers, but measured through infrastructural capabilities under a CAV setting. Finally, to ensure that the same bidding range applies to all vehicles q_l^t waiting in lanes serving a specific phase s, the maximum of the values d_l^t in the set D_s^t divided by the number of corresponding lanes is assigned as the reservation distance for all lanes in Ls. The process is outlined in Algorithm 1 in D2.3.

3.4.3 Bidding scheme

In the proposed approach, drivers are assumed to have a known VOT, yet are allowed to disproportionally increase their bid relative to their waiting time at the junction, as it accumulates. In this way, we consider the dynamic bidding behavior of users, contrary to assuming a deterministic user behavior at each time step. The concept of user impatience and the bidding rules are outlined in the following.

Impatience function

Across the respective literature, users are typically assumed to bid according to their known VOT [114] [115]. However, such an assumption overlooks the effect of accumulated delay on bidding behavior. Indeed, a high-VOT driver may not bid very high if they enter an intersection during a green phase, whereas a low-VOT user may grow more impatient as their delay disproportionally increases and decide to significantly increase their bid. To account for such cases, we use a sigmoid function to represent what we refer to as driver impatience. To capture dynamic bidding we use a sigmoid-based function (17), which is used as a multiplicative factor for driver VOT, ranging from 1 to 2.

$$P_i(w_i^t) = min(P_{imax}, 1 + \frac{1}{1 + e^{-c1_i \cdot (w_i^t - c2_i)}}) \tag{17}$$

where $c1_i$ governs the impatience of the driver, i.e. the larger $c1_i$ is the more impatient the driver is and $c2_i$ is the time where the impatience function becomes 1.5. These parameters essentially describe driver behavior and have unique values for each driver; we assume these as uniformly distributed within a specific range, and that these parameters can be declared by the driver at the start of their trip, along with their VOT. The wallet agent can then compute bids automatically based on current conditions. It should be noted that the developed function increases the initially declared VOT, thus the driver's lower bid is bounded by the declared VOT under this configuration.

Bid calculation

The bid for each driver is calculated based on the current traffic light phase, their VOT, impatience, and their current waiting time. It should be noted that due to the relationship between maximum distance and minimum green time, winners are guaranteed to cross the intersection. This ensures that users cannot pay multiple times for crossing the intersection. Two cases may be discerned in this respect:

1. For users who are currently in a queue at the red light, the time needed to cross through the intersection depends on their location in the queue and the saturation headway.

2. For users who have entered the lane-specific auction zone and face a green light indication, the time needed to cross the intersection depends on their current speed and distance from the junction.

The bid calculation is shown in Equation (18), as follows:

$$b_i^t = \begin{cases} P_i(w_i^t) \cdot \rho \cdot n_i^t \cdot VOT_i & s_i^t = \text{red} \\ P_i(w_i^t) \cdot v_i^t \cdot x_i^t \cdot VOT_i & s_i^t = \text{green} \end{cases} \tag{18}$$

Due to hysteresis phenomena that may be observed in traffic light queues if a vehicle facing a green light indication has a speed of zero, then its bid is calculated based on case 1.

3.4.4 Second-price sealed-bid auction

In the proposed setting, the traffic signal phases are assumed to be the auction participants. Unconstrained sequence transitioning between green phases is allowed to enhance the utilization of intersection capacity [116]. Bids from drivers in corresponding movements are combined to form the final bid for each phase. Under such a mechanism, the winner is the phase submitting the highest bid, yet the corresponding payment is equal to the second-highest bid. Drivers in the winning movements are then required to pay a fee that is proportional to what they originally bid. To ensure proper operation of the intersection for pedestrians as well as prevent unwarranted phase switching and associated time loss, a minimum green time is guaranteed for any phase that wins the auction. As such, if the winning phase is different than the active green phase at the time of the auction, a yellow phase is activated for the corresponding phase and subsequently, the winning phase is allotted a minimum duration of green time. If the winning phase is already active, the corresponding green time is extended by 5 seconds, and a new auction is subsequently triggered. If a phase exceeds its maximum green time, it is deactivated during the next auction round. The overall process is outlined in the pseudocode provided in Algorithm 2 in D2.3. Dynamic bidding auction strategies are promising venues for decentralizing traffic management. Our research showcases how such an approach could lead to considerable benefits, ensuring incentive compatibility, reducing intersection delay and increasing throughput, further showcasing that early adoption might be viable, given the rather positive results obtained for relatively low penetration rate scenarios. In multimodal mobility services, auctioning strategies can therefore play a fundamental role in reshaping traffic signal priority so as to take into account e.g. vehicle occupancy, public transport services, ride-sharing, etc, contributing to sustainability objectives.

3.5 Cooperative schemes for local bottleneck control

Next, we proceed to present a conceptual framework of cooperative schemes for local bottleneck and intersection control in a (fully or partially) connected environment, which is developed by TUD. More details about integrating the auctioning schemes in a network implementation are provided in Deliverable 2.4.

3.5.1 Conceptual framework

At the level of individual bottlenecks and intersections, vehicle connectivity technology (V2X) offers promising venues for improved management. Thanks to these ITS capabilities, better information can be collected by RSUs when determining how to manage the intersection, not only in relation to measurements directly related to traffic itself (e.g., vehicle distance from stop line, expected vehicle trajectory) but further capturing behavioral aspects, such as vehicle occupancy, Value of Time, vehicle class, vehicle type, paving the road for more efficient and detailed control [117] [118] [119]. Thanks to this added communication layer, approaches and techniques that, in real-time, adjust signal timings and phases adaptively based on the information retrieved from connected participants became therefore a feasible approach [120] [117] [121]. Cooperation between the different users (classes) can be achieved, under ideal conditions, by considering maximum social welfare as the overarching objective of the intersection controller. Optimization-based approaches, such as Model Predictive Control, have found considerable success in intersection management, allowing on the one hand to express complex multicriteria objectives, while on the other ensuring strict compliance with constraints arising from practical deployment, e.g., in relation to safety margins in operations, phase/movement compatibility, etc. [122] [123].

To achieve cooperative management considering connected vehicles, we introduce this information source in an MPC framework, developing appropriate prediction models for both non-connected and connected traffic participants, reflecting the different information quality and granularity. Consequently, we developed an MPC controller that, considering the users' objectives and preferences as well as road managers' policies (expressed through weights), optimizes the combination of signal timings and block sequences (see Figure 19) over a rolling horizon, in a mixed context of both macroscopic measurements (flows and queues) and microscopic measurements (connected users' states: positions and speeds). Below are detailed theoretical frameworks and algorithms for the MPC controller to optimize for the collective of the users' utilities, considering the policy weights.

Since real-time performance is a necessity in intersection control, the allocation rule necessitates a direct-revelation mechanism where agents are asked to reveal directly their private valuation in the first step, in contrast to the ascending mechanisms. It is assumed that connected users' biddings are automatically determined by wallet agents that valuate delay through a utility function, considering factors like their value of time, travel purpose, if they are going to be late and by how much, the weather conditions and if they are affected by it, etc. Each connected user communicates with the signal controller their bid, preferences, real-time state (i.e., their positioning and speed), and intended turning direction. Trajectory models are developed to predict connected vehicles and cyclists' future states to inform the objective function of the approaching connected users' travel costs in any given

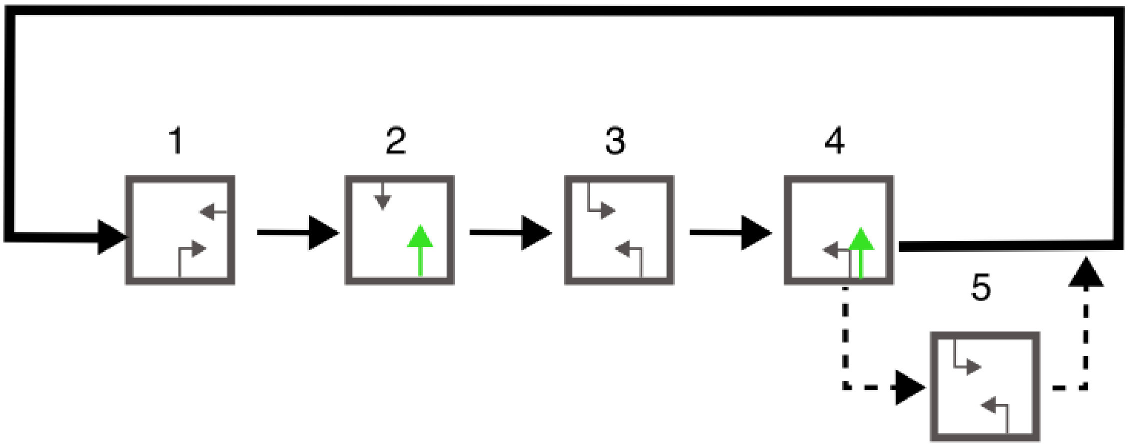


Figure 19: Example control structure. After block 4, either 1 or 5 will follow.

signal timing and block sequence, according to their predicted delay and if they would stop.

Figure 20 depicts the architecture of the MPC controller and how they relate. Block sequence construction and green-time optimization constitute the two main components of this controller. First, are generated the set of all candidate block sequences $\{B_j\}$ that last at least for the prediction horizon. This is done, more specifically, through algorithms A1–A3 (discussed in detail in D2.2.) that respectively compute the earliest possible end of greens for the current block (t^{ge}), end of all movements given a block sequence $(t^{(ge,out)})$, and Candidate block sequences (B_j). Once the candidate block sequences $\{B_j\}$ are generated, the green times (t^g) are optimized for all $B_j \in \{B_j\}$, considering the approaching traffic of connected and non-connected users on different movements, their biddings, state and predicted trajectory, and the policy weights. In each optimization iteration, algorithms A4 and A5 respectively compute the signal color sequence and timing (Q^{time}, Q^{color}) and their corresponding green fractions $(\gamma_m \forall m \in 1, 2, \ldots, M)$ for different values of decision variable t^g. Ultimately, the best-performing solution is selected and planned for realization in the next time step.

Optimization is triggered either at regular time intervals or when new information is available, e.g., when a queue is detected to have become empty and the movement has green (the same with the push-button for cyclists and pedestrians), when a detector becomes occupied while the corresponding light is red (the same with push-buttons), when an approaching tram or emergency vehicle is detected, or when connected users are detected. Further details of this method and algorithm, as well as its application in the pilots, are discussed in Deliverable 2.2.

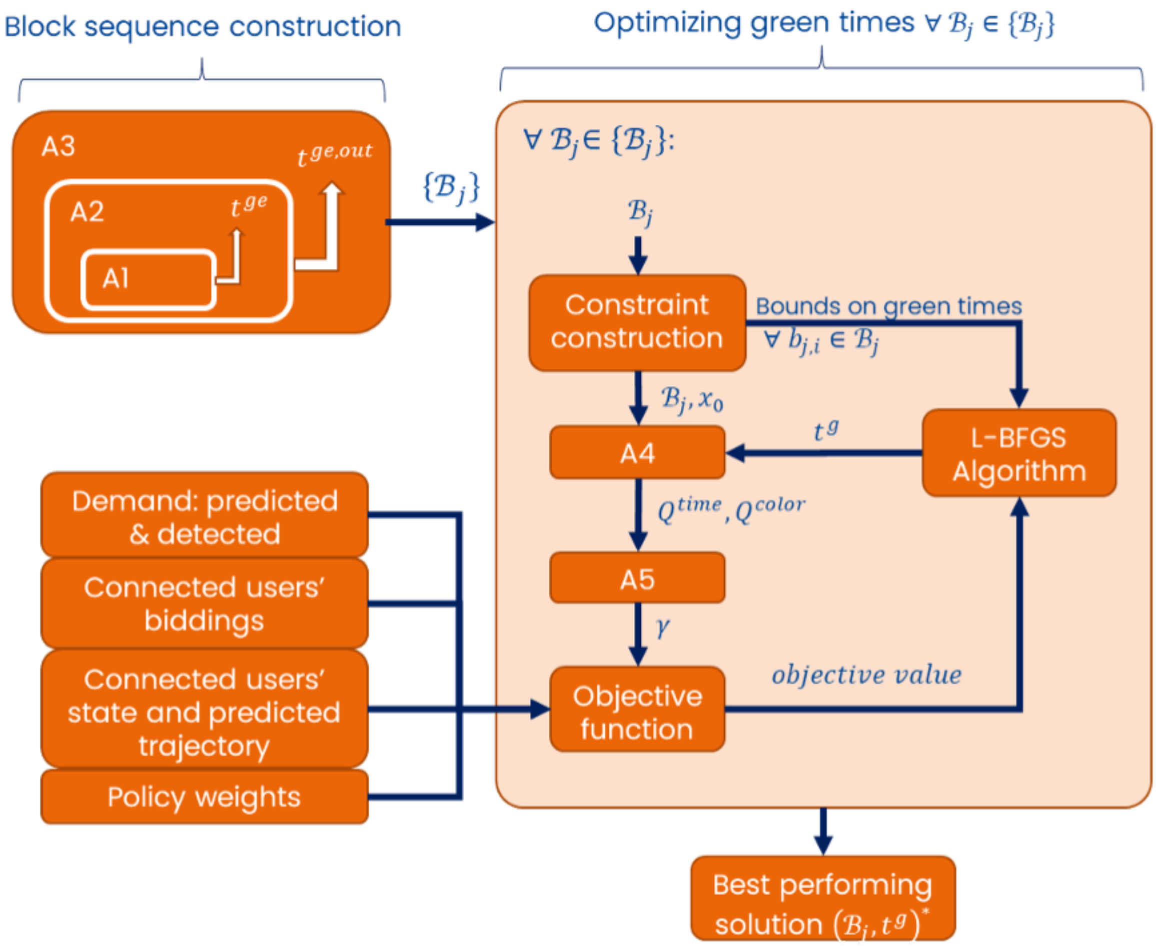


Figure 20: Configuration of the MPC controller.

4 Conclusion

In the deliverable, we investigate the benefits of a distributed approach to traffic and mobility management, and the preconditions for its effectiveness. To overcome the "Price of Anarchy" or the possible inefficiency of distributed decision-making, we have proposed a variety of sophisticated control architectures based on an extended set of mathematical/formal relationships that allow one to improve system efficiency.

We begin by discussing the benefits of using decentralized, distributed systems in transportation, and examine the price of anarchy under decentralized systems. We find in a stylized model that, it is possible that distributed systems with suitable information exchange and feedback/incentive systems can actually perform optimally or close to optimal with the methodologies developed in DIT4TraM. We point out that, neglecting the effect of subsystem interactions and uncertainties due to humans in the loop, can result in an undesirable evolution of the system state.

Furthermore, we proceed to design the proper distributed architectures and information exchange requirements for different DIT4Tram subsystems, which achieve better efficiency and resiliency. We develop distributed control architecture for different applications and modes in transportation systems.

We start with designing various passenger-oriented control architectures for large-scale urban systems, which are concerned with traffic signal control, dynamic bus lanes, and the repositioning of on-demand transport, respectively. In particular, we highlight the importance of combining PC strategy with MP local regulators regarding effective traffic signal control. In contrast, we also introduce RL as a compatible approach, which learns from experience and data to adapt to changing traffic patterns. Last but not least, a jam-tree approach is developed to identify traffic bottlenecks and facilitate better signal control.

The results in Task 1.1 also benefit the development of auctioning schemes for multimodal mobility services. In multimodal mobility services, auctioning strategies can therefore play a fundamental role in reshaping traffic signal priority so as to take into account, e.g. vehicle occupancy, public transport services, ride-sharing, etc., contributing to sustainability objectives.

The report also discusses cooperative schemes for local bottleneck control. By introducing information sources from connected vehicles, we develop appropriate prediction models in an model predictive control (MPC) framework for both non-connected and connected traffic participants, reflecting the different information quality and granularity. Thus, cooperation between the different users (classes) can be achieved under certain conditions.

In summarize, Task 1.1 contributes in the early-stage development of control architecture of different systems. Inspired by the preliminary outcomes, the real design, control logic and methods are further developed and deepened in detail in Work Packages 2-5.

Dissemination of work for D1.1

Conference Papers

[1] C. V. Beojone and N. Geroliminis, "Providing a revenue-forecasting information scheme as an incentive to relocate compliant ride-sourcing drivers," in *102nd Annual Meeting of the Transportation Research Board*, 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] C. V. Beojone and N. Geroliminis, "A dynamic multi-region mfd model for ride-sourcing systems with ridesplitting," in *101nd Annual Meeting of the Transportation Research Board*, Washington, D.C., 2022.

[4] C. V. Beojone and N. Geroliminis, "An optimized driver repositioning strategy in ridesplitting with earning estimates," in *the 10th symposium of the European Association for Research in Transportation (hEART)*, Leuven, Belgium, 2022.

[5] C. V. Beojone and N. Geroliminis, "Repositioning ridesplitting vehicles through pricing: A two-region simulated study," in *the 11th Triennial Symposium on Transportation Analysis conference (TRISTAN XI)*, Mauritius Island, 2022.

[6] M. Wang and N. Geroliminis, "The impact of proactive ride-splitting incentives on revenue and service quality," in *the 10th symposium of the European Association for Research in Transportation (hEART)*, Leuven, Belgium, 2022.

Manuscripts under review

[1] C. V. Beojone, Z. Pengbo, I. Ilber Sirmatel, and N. Geroliminis, "A hierarchical control framework for vehicle repositioning in ride-hailing systems," in *The 25th International Symposium on Transportation and Traffic Theory (ISTTT25) (under review)*, Ann Arbor, Michigan, 2024.

[2] C. V. Beojone and N. Geroliminis, "A dynamic multi-region mfd model for ride-sourcing with ridesplitting," *Transportation Research Part B (under review)*, 2023.

[3] 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.

[4] C. Carissimo, "The Social Benefit of Exploration in Braess's Paradox," *Games and Economic Behavior (under review)*, 2022.

[5] 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.

[6] 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] J. Ouyang, M. Li, Z. Zhang, and T. Tang, "Multi-timescale active and reactive power-coordinated control of large-scale wind integrated power system for severe wind speed fluctuation," *IEEE Access*, vol. 7, pp. 51 201–51 210, 2019.

[2] Y. Xu, Z. Y. Dong, R. Zhang, and D. J. Hill, "Multi-timescale coordinated voltage/var control of high renewable-penetrated distribution systems," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4398–4408, 2017.

[3] F. Qiao, J. Ma, and Y. Li, "Bi-level multi time-scale voltage/var optimization and control in a hybrid distribution network," in *8th Renewable Power Generation Conference (RPG 2019)*, IET, 2019, pp. 1–7.

[4] R. Zafar, J. Ravishankar, J. E. Fletcher, and H. R. Pota, "Multi-timescale voltage stability-constrained volt/var optimization with battery storage system in distribution grids," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 868–878, 2019.

[5] J. O. Dada and P. Mendes, "Multi-scale modelling and simulation in systems biology," *Integrative Biology*, vol. 3, no. 2, pp. 86–96, 2011.

[6] C. H. Lee and H. G. Othmer, "A multi-time-scale analysis of chemical reaction networks: I. deterministic systems," *Journal of mathematical biology*, vol. 60, pp. 387–450, 2010.

[7] W. W. Grabowski and P. K. Smolarkiewicz, "A multiscale anelastic model for meteorological research," *Monthly Weather Review*, vol. 130, no. 4, pp. 939–956, 2002.

[8] G. Castorina, M. T. Caccamo, S. Magazù, and L. Restuccia, "Multiscale mathematical and physical model for the study of nucleation processes in meteorology," *Atti della Accademia Peloritana dei Pericolanti-Classe di Scienze Fisiche, Matematiche e Naturali*, vol. 96, no. S3, p. 6, 2018.

[9] A. Chapman and M. Mesbahi, "Multiple time-scales in network-of-networks," in *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 5563–5568.

[10] N. Fenichel, "Geometric singular perturbation theory for ordinary differential equations," *Journal of differential equations*, vol. 31, no. 1, pp. 53–98, 1979.

[11] D. R. Smith, *Singular-perturbation theory: an introduction with applications.* Cambridge University Press, 1985.

[12] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[13] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust, "Applications of the dynamic mode decomposition," *Theoretical and computational fluid dynamics*, vol. 25, pp. 249–259, 2011.

[14] R. Scattolini, "Architectures for distributed and hierarchical model predictive control–a review," *Journal of process control*, vol. 19, no. 5, pp. 723–731, 2009.

[15] S.-H. Wang and E. Davison, "On the stabilization of decentralized control systems," *IEEE Transactions on Automatic Control*, vol. 18, no. 5, pp. 473–478, 1973.
[16] M. Hovd and S. Skogestad, "Sequential design of decentralized controllers," *Automatica*, vol. 30, no. 10, pp. 1601–1607, 1994.
[17] E. Davison and I. Ferguson, "The design of controllers for the multivariable robust servomechanism problem using parameter optimization methods," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 93–110, 1981.
[18] A. İftar, "Decentralized estimation and control with overlapping input, state, and output decomposition," *Automatica*, vol. 29, no. 2, pp. 511–516, 1993.
[19] J. B. Rawlings, D. Bonné, J. B. Jorgensen, A. N. Venkat, and S. B. Jorgensen, "Unreachable setpoints in model predictive control," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2209–2215, 2008.
[20] A. Alessio and A. Bemporad, "Decentralized model predictive control of constrained linear systems," in *2007 European Control Conference (ECC)*, IEEE, 2007, pp. 2813–2818.
[21] X. Du, Y. Xi, and S. Li, "Distributed model predictive control for large-scale systems," in *Proceedings of the 2001 American Control Conference. (Cat. No. 01CH37148)*, IEEE, vol. 4, 2001, pp. 3142–3143.
[22] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE, 2005, pp. 6680–6685.
[23] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Implementable distributed model predictive control with guaranteed performance properties," in *2006 American Control Conference*, IEEE, 2006. DOI: 10.1109/ACC.2006.1655424.
[24] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. M. Raimondo, "Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 324–338, 2008.
[25] W. Findeisen, F. N. Bailey, M. Brdys, K. Malinowski, P. Tatjewski, and A. Wozniak, *Control and coordination in hierarchical systems*. John Wiley & Sons, 1980.
[26] R. Negenborn, B. De Schutter, and J. Hellendoorn, "Efficient implementation of serial multi-agent model predictive control by parallelization," in *2007 IEEE International Conference on Networking, Sensing and Control*, IEEE, 2007, pp. 175–180.
[27] R. R. Negenborn, B. De Schutter, and J. Hellendoorn, "Multi-agent model predictive control for transportation networks: Serial versus parallel schemes," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 353–366, 2008.

[28] R. Scattolini and P. Colaneri, "Hierarchical model predictive control," in *2007 46th IEEE Conference on Decision and Control*, IEEE, 2007, pp. 4803–4808.
[29] T. Backx, O. Bosgra, and W. Marquardt, "Integration of model predictive control and optimization of processes: Enabling technology for market driven process operation," *IFAC Proceedings Volumes*, vol. 33, no. 10, pp. 249–260, 2000.
[30] G. J. Pappas, G. Lafferriere, and S. Sastry, "Hierarchically consistent control systems," *IEEE transactions on automatic control*, vol. 45, no. 6, pp. 1144–1160, 2000.
[31] J. Kadam, W. Marquardt, M. Schlegel, *et al.*, "Towards integrated dynamic real-time optimization and control of industrial processes," *Proceedings foundations of computer-aided process operations (FOCAPO2003)*, pp. 593–596, 2003.
[32] A. C. Zanin, M. T. De Gouvea, and D. Odloak, "Integrating real-time optimization into the model predictive controller of the fcc system," *Control Engineering Practice*, vol. 10, no. 8, pp. 819–831, 2002.
[33] L. Adacher, E. Cipriani, and A. Gemma, "The global optimization of signal settings and traffic assignment combined problem: A comparison between algorithms.," *Advances in Transportation Studies*, no. 36, 2015.
[34] A. L. C. Bazzan and F. Klügl, "A review on agent-based technology for traffic and transportation," *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 375–403, 2014. DOI: 10.1017/S0269888913000118.
[35] S. El-Tantawy, B. Abdulhai, and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (marlin-atsc): Methodology and large-scale application on downtown toronto," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013. DOI: 10.1109/TITS.2013.2255286.
[36] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: An overview," in *Innovations in Multi-Agent Systems and Applications - 1*, D. Srinivasan and L. C. Jain, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 183–221. DOI: 10.1007/978-3-642-14435-6_7.
[37] N. R. Council, *Disaster Resilience: A National Imperative*. Washington, DC: The National Academies Press, 2012. DOI: 10.17226/13457.
[38] C. Poulin and M. B. Kane, "Infrastructure resilience curves: Performance measures and summary metrics," *Reliability Engineering & System Safety*, vol. 216, p. 107 926, 2021. DOI: 10.1016/j.ress.2021.107926.
[39] M. Akbarzadeh, S. Memarmontazerin, S. Derrible, and S. F. Salehi Reihani, "The role of travel demand and network centrality on the connectivity and resilience of an urban street system," *Transportation*, vol. 46, no. 4, pp. 1127–1141, 2019. DOI: 10.1007/s11116-017-9814-y.
[40] I. G. Kasmalkar, K. A. Serafin, Y. Miao, *et al.*, "When floods hit the road: Resilience to flood-related traffic disruption in the san francisco bay area and

beyond," *Science Advances*, vol. 6, no. 32, eaba2423, 2020. DOI: 10.1126/sciadv.aba2423.

[41] H. Zhang, C. Liu, W. Zhang, G. Zheng, and Y. Yu, "GeneraLight: Improving Environment Generalization of Traffic Signal Control via Meta Reinforcement Learning," *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020. DOI: 10.1145/3340531.

[42] X. Zang, H. Yao, G. Zheng, N. Xu, K. Xu, and Z. Li, "MetaLight: Value-Based Meta-Reinforcement Learning for Traffic Signal Control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 1153–1160, Apr. 2020. DOI: 10.1609/AAAI.V34I01.5467.

[43] H. Wei, C. Chen, G. Zheng, *et al.*, "Presslight: Learning Max pressure control to coordinate traffic signals in arterial network," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 1, pp. 1290–1298, 2019. DOI: 10.1145/3292500.3330949.

[44] M. Korecki and D. Helbing, "Analytically guided reinforcement learning for green it and fluent traffic," *IEEE Access*, vol. 10, pp. 96 348–96 358, 2022. DOI: 10.1109/ACCESS.2022.3204057.

[45] M. Korecki, "Adaptability and sustainability of machine learning approaches to traffic signal control," *Scientific Reports*, vol. 12, no. 1, p. 16 681, Oct. 2022. DOI: 10.1038/s41598-022-21125-3.

[46] J. Barceló and J. Casas, "Dynamic network simulation with aimsun," *Simulation approaches in transportation analysis: Recent advances and challenges*, pp. 57–98, 2005.

[47] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator vissim," *Fundamentals of traffic simulation*, pp. 63–93, 2010.

[48] A. Haydari and Y. Yılmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 11–32, 2020.

[49] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: An overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*, ThinkMind, 2011.

[50] H. Zhang, S. Feng, C. Liu, *et al.*, "Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario," in *The world wide web conference*, 2019, pp. 3620–3624.

[51] M. Korecki, D. Dailisan, and D. Helbing, "How well do reinforcement learning approaches cope with disruptions? The case of traffic signal control," *Royal Society of Open Science (under review)*, 2022.

[52] D. Chen, M. M. Chowdhury, and S. Latif, "Data breaches in corporate setting," in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2021, pp. 01–06. DOI: 10.1109/ICECCME52200.2021.9590974.

[53] J. A. Guerrero-Ibanez, S. Zeadally, and J. Contreras-Castillo, "Integration challenges of intelligent transportation systems with connected vehicle, cloud

computing, and internet of things technologies," *IEEE Wireless Communications*, vol. 22, no. 6, pp. 122–128, 2015. DOI: 10.1109/MWC.2015.7368833.
[54] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: Architecture, protocols, and security," *IEEE internet of things Journal*, vol. 5, no. 5, pp. 3701–3709, 2017. DOI: 10.1109/JIOT.2017.2690902.
[55] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.
[56] P. Kairouz, H. B. McMahan, B. Avent, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
[57] J. G. Wardrop and J. I. Whitehead, "Road paper: Some theoretical aspects of road traffic research," *Proceedings of the Institution of Civil Engineers*, vol. 1, no. 3, pp. 325–362, 1952. DOI: 10.1680/ipeds.1952.11259.
[58] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," *Computer Science Review*, vol. 3, no. 2, pp. 65–69, May 2009. DOI: 10.1016/J.COSREV.2009.04.003.
[59] J. F. Nash Jr, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.
[60] J. Thai, N. Laurent-Brouty, and A. M. Bayen, "Negative externalities of gps-enabled routing applications: A game theoretical approach," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 595–601. DOI: 10.1109/ITSC.2016.7795614.
[61] T. Cabannes, F. Shyu, E. Porter, *et al.*, "Measuring regret in routing: Assessing the impact of increased app usage," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2589–2594. DOI: 10.1109/ITSC.2018.8569758.
[62] T. Cabannes, M. Sangiovanni, A. Keimer, and A. M. Bayen, "Regrets in routing networks: Measuring the impact of routing apps in traffic," *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, vol. 5, no. 2, pp. 1–19, 2019.
[63] D. Braess, "Über ein paradoxon aus der verkehrsplanung," *Unternehmensforschung*, vol. 12, pp. 258–268, 1968.
[64] A. Belov, K. Mattas, M. Makridis, M. Menendez, and B. Ciuffo, "A microsimulation based analysis of the price of anarchy in traffic routing: The enhanced braess network case," *Journal of Intelligent Transportation Systems*, vol. 26, no. 4, pp. 448–460, 2022.
[65] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis, "The price of anarchy in transportation networks by estimating user cost functions from actual traffic data," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 789–794. DOI: 10.1109/CDC.2016.7798364.
[66] S. C. Dafermos and F. T. Sparrow, "The traffic assignment problem for a general network," *Journal of Research of the National Bureau of Standards B*, vol. 73, no. 2, pp. 91–118, 1969.

[67] H.-J. Youn, H. Jeong, F. Roth, M. Silver, M.-H. Clutier, and P. Ittzes, "Price of anarchy in boston road network," *Journal of the Korean Physical Society*, vol. 48, no. 9, p. 217, 2006.
[68] H. Youn, M. T. Gastner, and H. Jeong, "Price of anarchy in transportation networks: Efficiency and optimality control," *Phys. Rev. Lett.*, vol. 101, p. 128 701, 12 Sep. 2008. DOI: 10.1103/PhysRevLett.101.128701.
[69] J. Zhang, S. Pourazarm, C. G. Cassandras, and I. C. Paschalidis, "The price of anarchy in transportation networks: Data-driven evaluation and reduction strategies," *Proceedings of the IEEE*, vol. 106, no. 4, pp. 538–553, 2018. DOI: 10.1109/JPROC.2018.2790405.
[70] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.
[71] T. Roughgarden, "The price of anarchy is independent of the network topology," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, 2002, pp. 428–437. DOI: 10.1016/S0022-0000(03)00044-8.
[72] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[73] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992. DOI: 10.1007/BF00992698.
[74] C. Carissimo, "The Social Benefit of Exploration in Braess's Paradox," *Games and Economic Behavior (under review)*, 2022.
[75] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *29th IEEE Conference on Decision and Control*, IEEE, 1990, pp. 2130–2132.
[76] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.
[77] P. Varaiya, "The max-pressure controller for arbitrary networks of signalized intersections," in *Advances in Dynamic Network Modeling in Complex Transportation Systems*, Springer, 2013, pp. 27–66.
[78] T. Wongpiromsarn, T. Uthaicharoenpong, Y. Wang, E. Frazzoli, and D. Wang, "Distributed traffic signal control for maximum network throughput," in *2012 15th international IEEE conference on intelligent transportation systems*, IEEE, 2012, pp. 588–595.
[79] J. Gregoire, X. Qian, E. Frazzoli, A. De La Fortelle, and T. Wongpiromsarn, "Capacity-aware backpressure traffic signal control," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 2, pp. 164–173, 2014.
[80] A. Kouvelas, J. Lioris, S. A. Fayazi, and P. Varaiya, "Maximum pressure controller for stabilizing queues in signalized arterial networks," *Transportation Research Record*, vol. 2421, no. 1, pp. 133–141, 2014.

[81] J. Gregoire, E. Frazzoli, A. de La Fortelle, and T. Wongpiromsarn, "Back-pressure traffic signal control with unknown routing rates," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 332–11 337, 2014.
[82] N. Xiao, E. Frazzoli, Y. Luo, Y. Li, Y. Wang, and D. Wang, "Throughput optimality of extended back-pressure traffic signal control algorithm," in *2015 23rd Mediterranean Conference on Control and Automation (MED)*, IEEE, 2015, pp. 1059–1064.
[83] N. Xiao, E. Frazzoli, Y. Li, Y. Luo, Y. Wang, and D. Wang, "Further study on extended back-pressure traffic signal control algorithm," in *2015 54th IEEE Conference on Decision and Control (CDC)*, IEEE, 2015, pp. 2169–2174.
[84] T. Le, P. Kovács, N. Walton, H. L. Vu, L. L. Andrew, and S. S. Hoogendoorn, "Decentralized signal control for urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 431–450, 2015.
[85] A. A. Zaidi, B. Kulcsár, and H. Wymeersch, "Traffic-adaptive signal control and vehicle routing using a decentralized back-pressure method," in *2015 European Control Conference (ECC)*, IEEE, 2015, pp. 3029–3034.
[86] L. Li and S. E. Jabari, "Position weighted backpressure intersection control for urban networks," *Transportation Research Part B: Methodological*, vol. 128, pp. 435–461, 2019.
[87] J. Wu, D. Ghosal, M. Zhang, and C.-N. Chuah, "Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 896–909, 2017.
[88] P. Mercader, W. Uwayid, and J. Haddad, "Max-pressure traffic controller based on travel times: An experimental analysis," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 275–290, 2020.
[89] M. W. Levin, J. Hu, and M. Odell, "Max-pressure signal control with cyclical phase structure," *Transportation Research Part C: Emerging Technologies*, vol. 120, p. 102 828, 2020.
[90] T. Chu, J. Wang, L. Codecà, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086–1095, 2019.
[91] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010.
[92] 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.
[93] J. Yu, P.-A. Laharotte, and L. Leclercq, "Distributed signal control for multi-modal traffic network: A deep reinforcement learning approach," *Transportation Research Part C: Emerging Technologies (under review)*, 2022.
[94] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," in *Proceedings of the 33rd International Con-*

ference on International Conference on Machine Learning - Volume 48, ser. ICML'16, New York, NY, USA: JMLR.org, Jun. 2016, pp. 1225–1234.
[95] C. Zhang, O. Vinyals, R. Munos, and S. Bengio, *A Study on Overfitting in Deep Reinforcement Learning*, Apr. 2018. DOI: 10.48550/arXiv.1804.06893. arXiv: 1804.06893 [cs, stat].
[96] S. Lämmer and D. Helbing, "Self-control of traffic lights and vehicle flows in urban road networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 4, pp. 1–36, 2008. DOI: 10.1088/1742-5468/2008/04/P04019.
[97] M. Keyvan-Ekbatani, A. Kouvelas, I. Papamichail, and M. Papageorgiou, "Exploiting the fundamental diagram of urban networks for feedback-based gating," *Transportation Research Part B: Methodological*, vol. 46, no. 10, pp. 1393–1403, 2012.
[98] M. Keyvan-Ekbatani, M. Papageorgiou, and V. L. Knoop, "Controller design for gating traffic control in presence of time-delay in urban road networks," *Transportation Research Procedia*, vol. 7, pp. 651–668, 2015.
[99] K. Aboudolas and N. Geroliminis, "Perimeter and boundary flow control in multi-reservoir heterogeneous networks," *Transportation Research Part B: Methodological*, vol. 55, pp. 265–281, 2013.
[100] D. Ingole, G. Mariotte, and L. Leclercq, "Perimeter gating control and city-wide dynamic user equilibrium: A macroscopic modeling framework," *Transportation research part C: emerging technologies*, vol. 111, pp. 22–49, 2020.
[101] N. Geroliminis, J. Haddad, and M. Ramezani, "Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: A model predictive approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 348–359, 2012.
[102] J. Haddad, "Optimal coupled and decoupled perimeter control in one-region cities," *Control Engineering Practice*, vol. 61, pp. 134–148, 2017.
[103] J. Haddad, "Optimal perimeter control synthesis for two urban regions with aggregate boundary queue dynamics," *Transportation Research Part B: Methodological*, vol. 96, pp. 1–25, 2017.
[104] M. Yildirimoglu, M. Ramezani, and N. Geroliminis, "Equilibrium analysis and route guidance in large-scale networks with mfd dynamics," *Transportation Research Procedia*, vol. 9, pp. 185–204, 2015.
[105] I. I. Sirmatel and N. Geroliminis, "Economic model predictive control of large-scale urban road networks via perimeter control and regional route guidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1112–1121, 2017.
[106] J. Viegas and B. Lu, "Widening the scope for bus priority with intermittent bus lanes," *Transportation Planning and Technology*, vol. 24, no. 2, pp. 87–110, 2001.

[107] 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.
[108] P. Anderson and N. Geroliminis, "Dynamic bus lanes with restricted car usage for congested arterial routes," Tech. Rep., 2014.
[109] D. Lin and S. E. Jabari, "Comparative Analysis of Economic Instruments in Intersection Operation: A User-Based Perspective," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Sep. 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294641.
[110] I. K. Isukapati and G. F. List, "Agent Based Framework for Modeling Operations at Isolated Signalized Intersections," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, ISSN: 2153-0017, Sep. 2015, pp. 2900–2906. DOI: 10.1109/ITSC.2015.477.
[111] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, ISSN: 2153-0017, Oct. 2013, pp. 529–534. DOI: 10.1109/ITSC.2013.6728285.
[112] H. Schepperle and K. Böhm, "Auction-Based Traffic Management: Towards Effective Concurrent Utilization of Road Intersections," in *2008 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, ISSN: 2378-1971, Jul. 2008, pp. 105–112. DOI: 10.1109/CECandEEE.2008.88.
[113] M. Vasirani and S. Ossowski, "A Market-Inspired Approach for Intersection Management in Urban Road Traffic Networks," en, *Journal of Artificial Intelligence Research*, vol. 43, pp. 621–659, Apr. 2012. DOI: 10.1613/jair.3560. [Online]. Available: https://www.jair.org/index.php/jair/article/view/10759 (visited on 02/06/2023).
[114] D. Lin and S. E. Jabari, "Pay for Intersection Priority: A Free Market Mechanism for Connected Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5138–5149, Jun. 2022, Conference Name: IEEE Transactions on Intelligent Transportation Systems. DOI: 10.1109/TITS.2020.3048475.
[115] D. Rey, M. W. Levin, and V. V. Dixit, "Online incentive-compatible mechanisms for traffic intersection auctions," en, *European Journal of Operational Research*, vol. 293, no. 1, pp. 229–247, 2021, Publisher: Elsevier. [Online]. Available: https://ideas.repec.org//a/eee/ejores/v293y2021i1p229-247.html (visited on 02/06/2023).
[116] M. Covell, S. Baluja, and R. Sukthankar, "Micro-Auction-Based Traffic-Light Control: Responsive, Local Decision Making," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, ISSN: 2153-0017, Sep. 2015, pp. 558–565. DOI: 10.1109/ITSC.2015.98.
[117] Y. Feng, K. L. Head, S. Khoshmagham, and M. Zamanipour, "A real-time adaptive signal control in a connected vehicle environment," en, *Trans-*

portation Research Part C: Emerging Technologies*, Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue, vol. 55, pp. 460–473, Jun. 2015. DOI: 10.1016/j.trc.2015.01.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X15000091 (visited on 12/06/2022).

[118] E. Namazi, J. Li, and C. Lu, "Intelligent Intersection Management Systems Considering Autonomous Vehicles: A Systematic Literature Review," *IEEE Access*, vol. 7, pp. 91 946–91 965, 2019, Conference Name: IEEE Access. DOI: 10.1109/ACCESS.2019.2927412.

[119] K. Yang, S. I. Guler, and M. Menendez, "Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles," en, *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 109–129, Nov. 2016. DOI: 10.1016/j.trc.2016.08.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X16301437 (visited on 12/05/2022).

[120] Q. Guo, L. Li, and X. (Jeff) Ban, "Urban traffic signal control with connected and automated vehicles: A survey," en, *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 313–334, Apr. 2019. DOI: 10.1016/j.trc.2019.01.026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18311641 (visited on 12/06/2022).

[121] N. J. Goodall, B. L. Smith, and B. (Park, "Traffic Signal Control with Connected Vehicles," en, *Transportation Research Record*, vol. 2381, no. 1, pp. 65–72, Jan. 2013, Publisher: SAGE Publications Inc. DOI: 10.3141/2381-08. [Online]. Available: https://doi.org/10.3141/2381-08 (visited on 12/06/2022).

[122] L. B. de Oliveira and E. Camponogara, "Multi-agent model predictive control of signaling split in urban traffic networks," *Transportation Research Part C: Emerging Technologies*, Information/Communication Technologies and Travel Behaviour Agents in Traffic and Transportation, vol. 18, no. 1, pp. 120–139, Feb. 2010. DOI: 10.1016/j.trc.2009.04.022. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X09000540 (visited on 01/19/2015).

[123] A. Jamshidnejad, I. Papamichail, M. Papageorgiou, and B. D. Schutter, "Sustainable Model-Predictive Control in Urban Traffic Networks: Efficient Solution Based on General Smoothening Methods," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 813–827, May 2018. DOI: 10.1109/TCST.2017.2699160.

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.