Reinforcement Learning for Urban Mobility

Michele Tirico[1,2,*], *Maxime* Tréca[3], and *Tarek* Chouaki[2,3]

[1]Univ Lyon, Univ Eiffel, ENTPE, LICIT-ECO7, F-69518 Lyon, France
[2]Univ Paris-Saclay, CentraleSupélec, LGI, 91190 Gif-sur-Yvette, France
[3]Institut de Recherche Technologique SystemX, 91120 Palaiseau, France
[4]Univ Paris-Saclay, Univ Versailles-St Quentin, DAVID, 78035 Versailles, France

Abstract. Improving the efficiency of urban mobility is a crucial challenge with regard to climate change, human health and logistics. From a technological point of view, reinforcement learning, a branch of artificial intelligence, is emerging as a promising computational tool for managing decision-making processes. This chapter provides an overview of reinforcement learning methodologies and gives proposals for further research directions by focusing on two crucial challenges in urban mobility: traffic signal control and mobility on demand.

Améliorer l'efficacité de la mobilité urbaine constitue un défi crucial en matière de changement climatique, de santé humaine et de logistique. D'un point de vue technologique, l'apprentissage par renforcement, branche de l'intelligence artificielle, apparaît comme un outil informatique prometteur pour gérer les processus décisionnels. Ce chapitre donne un aperçu des méthodologies d'apprentissage par renforcement et donne des propositions d'orientations de recherche supplémentaires en se concentrant sur deux défis cruciaux de la mobilité urbaine : le contrôle des feux de circulation et la mobilité à la demande.

1 Introduction

Improving the quality of mobility of humans and goods is a critical challenge in modern societies. As a key sector of economic activity, transport was responsible for 27% of Europe's total greenhouse gas emission in 2017, and demand for human mobility is expected to double by 2050 [1]. Following current trends, this increase of demand will have a negative environmental impact through the increment of air pollution and noise, as well as a societal impact through the increment of congestion and accidents [2]. Consequently, improving the performance and efficiency of current mobility systems is one of the most critical challenges of society to reduce environmental and social impacts of mobility [3, 4].

Nowadays, many cities have to deal with several daily challenges. Some emerged in the last years or decades: the onset of the Covid-19 pandemic has led many workers to consider working from home as a viable long-term option and prefer activity modes or private vehicles to avoid crowing in public transport [5]. One of the realistic scenarios anticipates that mobility will be considered as a service, where consumers buy a service of transportation (trips) from their mobile device rather than the mean of transportation itself. This new mobility

[*] e-mail: michele.tirico@entpe.fr

paradigm opens a large spectrum of technological challenges, such as integrating those services in an ergonomic device and ensuring the proper coordination of several transportation modes in an intermodal trip [6]. Novel transportation modes, such as automated vehicles, are in an advanced stage of development, and many experiments are conducted in several cities around the world [7]. Combined with shared and on demand vehicles, those recent technological innovations provide promising solutions for urban mobility.

Technological innovation will be essential in tackling future challenges in this dynamic context. Among them, Artificial Intelligence (AI) methodologies are a powerful emerging set of tools that boast the performance of transportation systems, making them more resource-efficient and helping in decision-making processes. AI approaches can be roughly divided into three branches [8]: supervised learning algorithms (for map training inputs to labelled outputs with predefined indexes), unsupervised learning algorithms (for finding applications for classification and clustering dimensionality reduction), and Reinforcement Learning (RL) algorithms (for controlling and decision-making processes). This last branch is considered an active learning because it provides a framework by which an agent can learn from its environment the best action to take in a given state [9]. As we will summarise in the following, RL is a promising research field in mobility, and it found application in many mobility issues.

Several applications of reinforcement learning can be founded in urban mobility (for instance, autonomous vehicle navigation, public transportation operation optimization, fleet management, and infrastructure planning). In the sequel, we focus on two emblematic applications: traffic signal control and mobility on demand. The chapter is organized as follows: in section 2, we provide basic definitions of RL; in section 3, we compare the single-agent and multi-agent approaches in traffic signal control; in section 4 we discuss the advantages and the drawbacks of model-free approaches and the recent improvements via the deep reinforcement learning approach by discussing their application in mobility on demand domain; in section 5 we discuss and conclude the chapter by giving proposals for further research directions.

2 Reinforcement Learning Setting and Algorithms

At a given time step, an agent immersed in a computational environment is in a defined state, observes the environment, and makes an action. The action is in accordance with the observation. The action affects the environment and returns a reward value to the agent. Hence, the reward is a function of the state and the action. The agent aims to maximise the cumulative reward value by learning the optimal policy through several interactions with the environment [9]. Therefore, solving an RL problem means finding the policy to achieve a maximum reward in the long term.

In the mobility context, an advantage of RL approach to optimisation and heuristic methods is that it can improve the algorithm's performance by learning via the interaction with the environment. Improving the system with real data, RL algorithms provide a better decision-making process. However, in several cases, those improvements have a tremendous cost in terms of computation time and setup of parameters. Ensuring the convergence of a RL algorithm is hence not ensured, demanding a high level of experience and solving before deployment in the real world.

The Q-learning algorithm is a simple and popular Reinforcement Learning (RL) algorithm used for solving such problems. It is based on the manual design of a state-action table, where for each pair is computed a Q-value. This value represents the expected future reward derived as a long-term cumulative reward for each state-action pair. By visiting each pair, the agent explores all possible combinations and finds the best option for all initial configurations.

2.1 Reinforcement Learning: Basic Distinctions

The agent has a partial understanding of the environment. He interacts with the environment and tries to understand how the environment is modelled. In the model-free approaches, the agent improves its policy without knowing the environmental model. The learning is free, meaning that the algorithm does not need to know the full functioning of the environment to search for the optimal policy. The only requirement for the agent is the reward function. Due to its simplicity for implementation and the generality of the approach, model-free approaches are largely applied in mobility. As discussed in section 4, model-free provides suitable results to solve on-demand problems and optimise vehicle-shared fleets' location.

Another distinction is about the representation of the policy. we can distinguish between policy-based and value-based algorithms. In the first class, we explicitly build a representation of the policy function, keeping in memory the policy during the learning. In the second class, the function is not explicitly designed; we only store the value function. The policy can be derived in a second time from the value function. In the first case, the algorithm estimates how good it is for the agent to be in a state, and in the second case, the algorithm estimates how good it is to perform an action in a given state.

The third main distinction is about the time-lap of the experiences that the agent considers to improve its policy. Therefore, we have on-policy and off-policy algorithms. In the first class of algorithms, the learning is online. The policy is updated with data collected at the same learning step. In the second class of algorithms, the learning is offline. The agent's experience at a given step is updated in a data buffer containing previous experiences. The policy is a function of the buffer: all this data are used to find the action for the next step.

2.2 Recent improvements

Classical RL algorithms based on Markov Decision Process (MDP) may not find the optimal policy efficiently in high-dimensional complex systems such as transportation systems. Visitation of each state-action pair can become computationally infeasible. Moreover, in several cases, uncertainly and a continuous state-action possible configuration makes unrealistic the definition of a discrete and deterministic state-action table. For those reasons, a representation with a Deep Neural Network (DNN) is often suitable to tackle mobility issues. In order to tackle those limits, numerous non-linear function approximation are proposed. In this field, Deep Reinforcement Learning (DRL), where a multi-layered structure model a continuous state-action space, is highly suited for complex environments. Compared to the Q-learning algorithm, where the process iterative updates each state-action value, a deep Q-learning algorithm uses DNN to approximate a function that maps states to values. Methods from this approach provide a more flexible approximation for non-linear functions. Moreover, avoiding hard-specified state representations, those approaches did not require a manual state-action set design. Improvements in the computational performance are discussed in section 4.

2.3 Multiple agents

Many mobility problems should be considered as the co-existence of several agents. Learning with multiple agents is a challenging task. An agent's environment also contains all agents that interact at some time with him. As an agent changes its state over time, the environment for other agents becomes dynamic. Moreover, each agent partially understands its environment since it cannot be informed of the configuration of the whole system of agents. An increasing number of agents also imply an increased state-action set of the whole system, driving to high-dimensionality problems. One of them is that when an agent optimises its actions without considering other agents, the optimal learning of the system becomes non-stationary. Several approaches have been proposed to address this problem, mainly integrating competitive and collaborative mechanisms between agents (see section 3).

3 Traffic Signal Control: Multi-Agent Applications

At the heart of any RL approach is the notion of *agent*, which influences the environment and learns from its feedback. Since this notion of the agent is paramount in the context of RL, its definition quite logically impacts the models in which they are applied. As mentioned in section 2, defining the agent or agents is one of the leading choices to perform when designing an RL algorithm. This draws the distinction between single-agent and multi-agent approaches. This choice greatly impacts the quality of the solution reached by the algorithm, the convergence speed and the results' stationarity.

In order to illustrate this impact, we consider the Traffic Signal Control (TSC) domain, which is an essential tool in managing modern urban mobility. The TSC aims to optimize traffic flows over a road network, typically by minimizing travel time through traffic light signals at the intersection level. The application of RL for TSC, a field colloquially known as RL-TSC, appeared in the 1990s using genetic RL algorithms to route traffic [10].

At their core, RL-TSC models are fairly simple. Given the current traffic configuration around an intersection (i.e. the environment state s_t), the agent chooses a traffic light configuration (i.e. the action a_t) influencing the flow of vehicles around the intersection. After this configuration is applied at the intersection level, the resulting traffic state (i.e. the next state s_{t+1}) and associated delay (i.e. the reward r_t) are passed back to the agent so that it can learn from its previous action. By iteratively testing multiple actions on different traffic states, the agent is expected to learn an optimal mapping between traffic states and traffic light settings, optimizing the traffic flows at the intersection.

3.1 Single Agent and Multi-agent Reinforcement Learning

Although straightforward in scenarios consisting of a single isolated intersection, modelling choices expand when tackling road networks composed of multiple intersections. Indeed, deterministic traffic management systems such as `Claire-Siti` or `GERTRUDE` optimize traffic flows using a central planner controlling multiple traffic lights simultaneously. However, this approach poorly translates to RL-TSC models since a single agent would observe the entire state of the network at once, which would cause significant dimensionality issues for the underlying RL algorithm. Such a limitation explains why most RL-TSC models on road networks with multiple intersections feature multiple agents learning in parallel [11], in a Multi-Agent Reinforcement Learning (MARL) setting.

In a MARL setting, agents usually have a limited and local view of their environment, which can be modelled as a Partially-Observable Markovian Decision Process (POMDP). Such models usually feature increased performance compared to centralized approaches in various RL tasks, improved robustness and fault-tolerance, and allow experience-sharing between agents (i.e. one agent can learn from the experiences of another) [12]. However, this standard model choice opens up the question of how multiple agents of the same network interact.

A simple and somewhat familiar approach to MARL models is to avoid explicitly modelling interactions between RL agents of the same system. On the one hand, this method has the advantage of staying relatively simple since each agent optimizes traffic locally without acknowledging others. Furthermore, local traffic optimization usually translates to good global performance metrics, as shown by the excellent results obtained by independent MARL models in the RL-TSC literature [13]. On the other hand, an inherent limitation of independent MARL models is the *non-stationarity* associated with each independent learning model. Indeed, since agents do not explicitly model the behaviour of other agents, their model of the local environment is skewed since it depends on the behaviour of other agents that changes over time. This flaw can cause the policies of local agents to oscillate and never reach an optimal equilibrium. A standard solution to the issue of non-stationarity in MARL models is to include information about other agents in the state definition of a local agent.

3.2 Coordination mechanisms in Traffic Signal Control

Modern RL-TSC models further push the interactions between neighbouring intersections and agents to maximize their optimization potential. Indeed, traffic light optimization benefits from large-scale strategies such as creating *green waves* across the main arteries of a road network. How do such methods model the interactions between neighbouring learning agents?

The first approach is to consider the environment state of neighbouring agents when choosing a local traffic light configuration. For instance, seeing that an intersection upstream has many vehicles heading towards the local intersection will likely influence the local traffic light setting. Such a model, in which agents do not explicitly communicate but observe the state of their neighbours, also known as *indirect coordination*, is featured in the `MARLIN-ATSC` method [14]. The `MARLIN` method has achieved state-of-the-art results in the early 2010s by using joint state-action optimization. Each agent observes the state of its neighbouring agents, predicts which traffic light settings they will implement, and choose its traffic light settings based on these predictions.

The second and more deeply integrated approach is making agents coordinate directly instead of predicting their future action choices. Although more computationally and theoretically complex, these methods are currently state-of-the-art in the RL-TSC literature and offer many possibilities. For instance, agents can automatically form green waves over arterials through forced coordination [17]. Using emergent communication protocols is another impressive and highly efficient approach for direct coordination. The DIAL model [16] defines a coordination model in which agents can send and receive messages without defining any common language beforehand. The language is learned through deep reinforcement learning *while solving the task concurrently*. This highly innovative approach to agent coordination has been applied to a TSC context [29] and has showcased excellent performances.

4 Mobility on Demand: Model-free Models and Deep Reinforcement Learning

4.1 Model-free and Model-based Models

RL algorithms can be distinguished according to whether they explicitly model their environment dynamics or not. A model of how the environment reacts to the learning agent(s) decisions can be built, learned and used to forecast the future states and consequently choose the action leading to the most preferred situation. From the perspective of MDP, this is equivalent to knowing the transition function. Approaches that take into account a model of the environment are called model-based approaches. Moerland et al. [18] comprehensively review these approaches and successful usages in the literature. However, these approaches are rarely used in highly complex and dynamic environments, as it becomes hard to properly model all the system components. This is especially true for real-world environments such as urban mobility.

In contrast to model-based approaches, model-free methods do not rely on learning the dynamics of the environment. Only the obtained rewards are used to build a better policy throughout interactions. A model can be built to predict the rewards associated with a state or a station-action pair, which can be used to derive the optimal policy. The Q-learning method mentioned in section 2 is an example of a model-free approach. Consequently, almost all of the RL approaches that are implemented for tasks related to urban mobility fall into the model-free category.

To illustrate this, we consider the emerging field of using RL approaches to operate Mobility-on-Demand (MoD) systems. Jiao et al. [19] use a SARSA method to relocate empty vehicles. Feng et al. [20] combine RL with Integer Linear Programming (ILP) to operate an intermodal MoD system that can either bring the travellers directly to their destinations or to public transport stations from which they can continue their trips. More precisely, RL is used to learn the values of each vehicle's state-action pairs of possible alternatives (assigned trips). The values are then used to build the ILP problem that is solved to determine the trip assignments. Wang et Chang [21] explore using RL to operate autonomous bus services and decide when buses should be sent on the line. Chouaki et al. [22] present a Q-learning approach for rebalancing empty vehicles in a MoD system to maximise the number of trips served by the fleet.

4.2 Dealing with dimensionality: Deep Reinforcement Learning

The high complexity of mobility-related problems does not just make using a model-based system challenging. Even with model-free approaches, the high dimensionality of state and action spaces needs to be taken into account and appropriately addressed. This is especially the case for the state space design and the value function. In classical tabular methods, the state value function (resp the action-state value function, which is, for instance, used to populate the Q-table in the Q-learning algorithm) is represented by storing its value for every possible state (resp state-action pair). The use of tabular methods for the operation of on-demand mobility systems has been investigated in the literature [23, 24]

Given that the size of the state space (or state action space) is exponential to the number of considered features, it rapidly becomes impossible to use tabular methods with high-dimensional problems. For example, considering a decentralized algorithm for MoD operation with a state space that includes the position of the vehicle and the origin and destination of each passenger leads to 10^{18} possible states considering 100 possible locations on the network and a $4-$seated vehicle. Moreover, a tabular representation updates the value of each situation in isolation from the others and cannot generalize to similar situations. In order to tackle these issues, various approaches for value function approximation are used in the RL literature. The idea is to model the state value function (or the state-action value function) as a parametric function and learn, through interaction, the correct parameter values.

A particular type of value function approximation technique, Artificial Neural Networks (ANN), has gained significant attention in recent years due to its successful use in supervised learning. ANNs can represent a wide range of functions, including non-linear ones. The back-propagation algorithm allows them to adapt their parameters to better fit into the observations.

RL approaches that use a deep ANN (an ANN with many layers) are known as deep reinforcement learning methods [9]. Given the high dimensionality of mobility-related problems, as illustrated above, many studies investigating RL use in mobility focus on deep-RL approaches [25]. Al-Abbasi et al. [26] present a Deep-RL framework to learn the dispatch decisions for on-demand vehicles. They combine three deep ANNs to learn travel times in the network, predict the following traveller demands and learn the value of state-action pairs. The ANN used for the latter purpose is known as a deep Q-network. Tang et al. [27] show that a value function represented using a deep ANN can be used to operate an on-demand system. Unlike previously mentioned approaches, Mao et al. [28] use a policy-based RL approach where the policy is represented as a deep ANN.

The use of deep reinforcement learning methods is already ubiquitous in some areas of the mobility literature. For instance, almost all novel traffic signal control methods use ANNs to compute optimal traffic signal settings [13]. As the performance of deep reinforcement learning continues to increase, allowing us to take more environment features into account, the use of deep ANNs is expected to continue to increase in RL methods that address mobility challenges.

5 Conclusion and Discussion

Artificial Intelligence (AI), and more precisely Reinforcement Learning (RL), show promising perspectives in the context of mobility. However, as this chapter shows, there are limitations to the current applications of reinforcement learning in the context of mobility. Indeed, the transportation field usually features highly non-linear and high-dimensional problems. Classical approaches based on MDPs can fall in a large and hand-specified design of a large and discrete action-state table. The most relevant drawback of this formalism is that it can provide a high sensitivity to learning parameters, providing some issues in finding the algorithm's convergence and demanding considerable computational time. In the mobility-on-demand problem context, those limits impact the capacity to design all the possible state-action combinations, preventing the system from providing the well-adapted action for a given state in an adequate time. Deep Reinforcement Learning (DRL) algorithms provide a consistent perspective, allowing to design a continuous state-action space, integrating uncertainty and obtaining a policy in a reasonable time.

Considering the example of traffic flow control, we can see that two RL approaches are applied: single agent and multi-agent RL. In the first approach, the transportation system is seen as an agent which learns from the environment the optimal strategy to maximize the reward. In the second approach, the transportation system is seen as a set of agents that co-evolve and co-learn their strategy. Cooperation and/or competition mechanisms can be integrated to improve the algorithm's ability to find the optimal strategy and/or reduce the learning time. Both approaches present advantages and drawbacks from computational and conception points of view. Decentralized approaches are often designed with a small set of states and actions for each agent, providing a better understanding of the policy. However, convergence is hard to find because each agent is immersed in an evolving environment composed of agents that changes their state during the simulation. Explicit and direct coordination between agents is often preferable with regard to model performance at the cost of computational and theoretical complexity. This observation can be taken even further when considering models featuring multiple agent types, such as hybrid models in which traffic light controllers and vehicles are two distinct types of communicating agents. Given the increased communication capabilities of many actors of modern mobility (e.g. vehicles, people and infrastructure), such complex MARL models are likely to become commonplace in the future.

In a model-based approach, agents often learn faster than those modelled with a model-free approach. An agent can reuse information previously achieved. However, the inconvenience is that those algorithms required a greater size storage cost than the other approach. Moreover, they are more sensible on the accuracy of the environmental model: the state-action table should be designed with accuracy to ensure a convenient algorithm performance.

Mobility systems are often characterized by a high degree of uncertainty. Since they are complex, unexpected and unpredictable behaviour often emerge from decentralized interactions of their elements. Moreover, since a mobility system is composed mainly of humans, individuals' unpredictability impacts the whole behaviour system. The design of individual behaviour and their interaction is one of the most relevant challenges in transport. For those reasons, uncertainty is crucial to understand the system well and to design strategies to improve its performance. In some circumstances, the MDP approach shows some limits because it assumes that the system's next state depends only on the present state and the action to take. Partial observable Markov Decision Process and Deep Reinforcement Learning show promising perspectives to consider those aspects better.

Future transportation systems are expected to include more autonomy, such as decision-making processes in traffic management, vehicle driving and vehicle co-existence. The whole system's performance can be improved, incrementing human quality of life and reducing urban impacts of climate change. In order to make this scenario possible, Reinforcement Learning will play a major role, providing a consistent technological background for decision processes.

References

[1] European Environment Agency. Transport and environment report 2021. Technical report.

[2] Guy Fournier. The New Mobility Paradigm. Transformation of Value Chain and Value Proposition Through Innovations. In Danielle Attias, editor, *The Automobile Revolution: Towards a New Electro-Mobility Paradigm*, pages 21–47. Springer International Publishing, 2017.

[3] United Nations General Assembly Transforming Our World. Transforming our world: the 2030 Agenda for Sustainable Development | Department of Economic and Social Affairs, 2015.

[4] IPCC. Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, 2014.

[5] Jonas De Vos. The effect of COVID-19 and subsequent social distancing on travel behavior. *Transportation Research Interdisciplinary Perspectives*, 5:100121, May 2020.

[6] David A. Hensher. Future bus transport contracts under a mobility as a service (MaaS) regime in the digital age: Are they likely to change? *Transportation Research Part A: Policy and Practice*, 98:86–96, April 2017.

[7] Eliane Horschutz Nemoto, Ines Jaroudi, and Guy Fournier. Introducing Automated Shuttles in the Public Transport of European Cities: The Case of the AVENUE Project. In Eftihia G. Nathanail, Giannis Adamos, and Ioannis Karakikes, editors, *Advances in Mobility-as-a-Service Systems*, Advances in Intelligent Systems and Computing, pages 272–285, Cham, 2021. Springer International Publishing.

[8] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. Google-Books-ID: EoYBngEACAAJ.

[9] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, Massachusetts, second edition edition, November 2018.

[10] Sadayoshi Mikami and Yukinori Kakazu. Genetic reinforcement learning for cooperative traffic signal control. In *Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence*, pages 223–228. IEEE, 1994.

[11] Patrick Mannion, Jim Duggan, and Enda Howley. *An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control*, pages 47–66. Springer International Publishing, Cham, 2016.

[12] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[13] Mohammad Noaeen, Atharva Naik, Liana Goodman, Jared Crebo, Taimoor Abrar, Zahra Shakeri Hossein Abad, Ana LC Bazzan, and Behrouz Far. Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications*, page 116830, 2022.

[14] Samah El-Tantawy, Baher Abdulhai, and Hossam 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*, 14(3):1140–1150, 2013.

[15] Maxime Tréca, Mahdi Zargayouna, Dominique Barth, and Julian Garbiso. Green wave coordination for traffic signal control using deep reinforcement learning. In *Urban Mobility 2*, 2022.

[16] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[17] Maxime Tréca, Mahdi Zargayouna, Dominique Barth, and Julian Garbiso. Green wave coordination for traffic signal control using deep reinforcement learning. In *Urban Mobility 2*, 2022.

[18] Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

[19] Yan Jiao, Xiaocheng Tang, Zhiwei Tony Qin, Shuaiji Li, Fan Zhang, Hongtu Zhu, and Jieping Ye. Real-world ride-hailing vehicle repositioning using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 130:103289, 2021.

[20] Siyuan Feng, Peibo Duan, Jintao Ke, and Hai Yang. Coordinating ride-sourcing and public transport services with a reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 138:103611, May 2022.

[21] Sung-Jung Wang and SK Jason Chang. Autonomous bus fleet control using multiagent reinforcement learning. *Journal of Advanced Transportation*, 2021:1–14, 2021.

[22] Tarek Chouaki, Sebastian Hörl, and Jakob Puchinger. Implementing reinforcement learning for on-demand vehicle rebalancing in matsim. *Procedia Computer Science*, 201:134–141, 2022.

[23] M. Gueriau, F. Cugurullo, R. A. Acheampong, and I. Dusparic. Shared autonomous mobility on demand: A learning-based approach and its performance in the presence of traffic congestion. 12(4):208–218, 2020.

[24] Christian Fluri, Claudio Ruch, Julian Zilly, Jan Hakenberg, and Emilio Frazzoli. Learning to operate a fleet of cars. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2292–2298. IEEE, 2019.

[25] Zefang Zong, Tao Feng, Tong Xia, Depeng Jin, and Yong Li. Deep reinforcement learning for demand driven services in logistics and transportation systems: A survey. *arXiv preprint arXiv:2108.04462*, 2021.

[26] Abubakr O. Al-Abbasi, Arnob Ghosh, and Vaneet Aggarwal. DeepPool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. 20(12):4714–4727, 2019.

[27] Xiaocheng Tang, Fan Zhang, Zhiwei Qin, Yansheng Wang, Dingyuan Shi, Bingchen Song, Yongxin Tong, Hongtu Zhu, and Jieping Ye. Value Function is All You Need: A Unified Learning Framework for Ride Hailing Platforms. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pages 3605–3615, New York, NY, USA, August 2021. Association for Computing Machinery.

[28] Chao Mao, Yulin Liu, and Zuo-Jun (Max) Shen. Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 115:102626, June 2020.

[29] Maxime Tréca. *Designing Traffic Signal Control Systems Using Reinforcement Learning*. PhD thesis, Université Paris Saclay, 2022.