Agent-based simulations of intermodal mobility-on-demand systems operated by reinforcement learning

Tarek Chouaki

université PARIS-SACLAY

Agent-based simulations of intermodal Mobility-on-Demand systems operated by Reinforcement Learning

Simulations multi-agent de systèmes de mobilité à la demande intermodaux opérés par des algorithmes à base d'apprentissage par renforcement

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°573, interfaces : matériaux, systèmes, usages (INTERFACES)
Spécialité de doctorat: Informatique
Graduate School : Sciences de l'ingénierie et des systèmes.
Référent : CentraleSupélec

Thèse préparée au **Laboratoire Génie Industriel** (Université Paris-Saclay, CentraleSupélec), sous la direction de **Jakob Puchinger**, Professeur, et le co-encadrement de **Sebastian Hörl**, Docteur.

Thèse soutenue à Paris-Saclay, le 04 juillet 2023, par

Tarek CHOUAKI

Composition du jury

Membres du jury avec voix délibérative

Dominique Barth Professeur des Universités, DAVID, Université de Versailles ST-Quentin-en-Yvelines	Président
Pierre-Olivier Vandanjon Chargé de Recherche dév. durable HDR, Université Gustave Eiffel, Nantes	Rapporteur & Examinateur
Mahdi Zargayouna Chargé de Recherche HDR, Université Gustave Eiffel	Rapporteur & Examinateur
Flore Vallet Enseignante-chercheuse HDR, IRT SystemX et CentraleSupélec	Examinatrice
Isabelle Nicolai Professeure des Universités, CentraleSupélec, Université Paris Saclay	Examinatrice
Moez Kilani Professeur des universités, Université du Littoral Cote d'Ôpale	Examinateur

THESE DE DOCTORAT
NNT : 2023UPAST094

université PARIS-SACLAY | ÉCOLE DOCTORALE Interfaces: matériaux, systèmes, usages

Titre: Simulations multi-agent de systèmes de mobilité à la demande intermodaux operés par des algorithmes à base d'apprentissage par renforcement

Mots clés: Simulations multi-agent; mobilité à la demande; apprentissage par renforcement; simulations prospectives; intermodalité

Résumé: Dans cette thèse, nous nous intéressons aux approches multi-agent de simulation de la mobilité et plus particulièrement des systèmes de mobilité à la demande. Un tel système consiste en une flotte de véhicules qui répondent à des requêtes de trajets au fur et à mesure qu'elles émanent de la part des voyageurs. Les véhicules peuvent donc servir plusieurs voyageurs au cours de la journée et offrir une alternative à la possession d'un véhicule privé.

Bien que nos approches soient génériques et peuvent s'appliquer à des cas d'usage variés, nous les illustrons sur un contexte localisé sur la région Île-de-France et le territoire de Paris-Saclay. Dans ce travail, un cas d'usage prospectif de la zone d'étude, incluant les lignes futur de transports ferrés, est construit. Plus précisément, nous tentons dans cette thèse de remplir les objectifs de recherche suivants : (i) évaluer l'impact des lignes de transports futures en Île-de-France et sur la zone de Paris-Saclay. (ii) Concevoir, simuler et dimensionner un système de mobilité à la demande intermodal et en évaluer l'impact sur les choix des voyageurs et sur les transports publics. (iii) Explorer l'utilisation des méthodes basées sur l'apprentissage par renforcement pour la gestion de ce type de systèmes ainsi que le potentiel de l'utilisation de simulations multi-agent pour l'évaluation de ces méthodes. Tous ces objectifs sont adressés avec un accent sur la réplicabilité et l'ouverture.

Title: Agent-based simulations of intermodal Mobility-on-Demand systems operated by Reinforcement Learning

Keywords: Agent-based simulation; mobility-on-demand; reinforcement learning; prospective simulations; intermodality.

Abstract: In this thesis, we focus on agent-based approaches for simulating mobility in general and mobility-on-demand systems in particular. A mobility-on-demand system consists in a fleet of vehicles that respond to trip requests from users in an online manner. Vehicles from the fleet can consequently serve multiple customers throughout the day and offer an interesting alternative to owning a private vehicle.

While our approaches are generic and can apply to various use cases, we illustrate them on a localized setting as we focus on the area of Île-de-France and Paris-Saclay.In this work, a prospective simulated use case for the study area which includes future rail lines is built. More precisely, we attempt, in this work, at fulfilling the following research objectives: (i) Assess the impact of future rail-based systems in Île-de-France/Paris-Saclay. (ii) Design, simulate and dimension an intermodal MoD system and assess its impact on travelers decisions and public transports. (iii) Explore the use of RL based methods for such a system and the potential of using agent-based simulations for the evaluation of such methods. All of these objectives are addressed with a focus on reproducibility and openness.

Acknowledgments

This research adventure of mine was completely unforeseeable to me five years ago. Now that I look back, it seems to be the obvious pathway for me. When I started this thesis in December 2019, all the ideas I had on how it would go on were contradicted by reality. A global pandemic, a few challenges and several awesome encounters later, I can only be thankful for the all the support I received, for this work could not have been completed if it was not for a many people related to me in so different ways.

I would like to first thank Jakob Puchinger, for wisely directing this thesis. The liberty accorded to me during this project has allowed me to explore different pathways and learn from my own mistakes.

I also thank Sebastian Hörl for his skillful co-supervision. His methodological guidance and technical help have been precious throughout this thesis. We have now accumulated several hours of discussions and they have taught me a lot.

Jakob, Sebastian, I very much thank you for your help and support. It has been a pleasure!

Special thanks to the Anthropolis Chair, its partners, IRT SystemX and the LGI for hosting and supporting this thesis. I had the chance of working in a healthy and motivating environment. This is mainly thanks to Flore, Sylvie, Tjark, Mariana and Yann. The Anthropolis team is definitely human centered.

I also thank Felix Carreyre with who enjoyed collaborating for more than two years now. My thesis would not have been the same without you. I hope most of our papers together are still to write.

At the end, I say thank you to all those I call friend or family, I indeed have the best. Especially, I would like to say thank you to my wife Sara for supporting me during not-so-easy times. You have endured me during my thesis, so we should be fine for the future. I also thank my brothers, Salim and Lyes, for being very differently but equally ...inspiring let's say. Finally, I thank my parents, Farida and Hocine. All that is good in me comes from you two. And for the rest, your contribution is much less important as I innovated. I promise I will try to not work as much when I visit you in the future as when I did in the past months.

Résumé

Aujourd'hui, la mobilité est le poumon de la civilisation moderne. Un système de transport de bonne qualité est considéré comme le reflet d'une civilisation moderne, et son bon fonctionnement est nécessaire à la productivité d'autres secteurs tels que l'industrie et le tourisme. Avec la majeure partie des populations vivant dans les villes, les défis auxquels celles-ci font face sont souvent des défis relatifs à la mobilité urbaine (c'est le cas pour la pollution, les nuisances sonores ou l'accès aux services publics). Grâce aux avancées majeures qu'ont connus ces dernières décennies sur les technologies d'information de communication, la disponibilité de l'information est aujourd'hui prise pour acquise. En utilisant un smartphone, il est possible de trouver le meilleur chemin pour atteindre une destination en utilisant divers modes de transport et en prenant en compte la situation de congestion de trafic et l'état des transports publics en temps réel. Forte de ces développements technologiques, la mobilité urbaine d'aujourd'hui ne consiste plus uniquement d'un réseau routier, de véhicules privés et de transports publics, mais comprend aussi un large spectre de modes flexibles se composant de trottinettes ou de vélos en libre-service et de plateformes de covoiturages. Ces nouveaux modes de transport offrent des perspectives de mobilité plus efficace avec plus de partage. À une époque où la nature limitée des ressources disponibles est mise en lumière et où les impacts négatifs de l'activité humaine sur l'environnement ne peut plus être niés et requière une action urgente, la soutenabilité est devenue un souci majeur des développements relatifs à la mobilité. Une solution de mobilité ne peut plus être uniquement évaluée sur sa vitesse, sécurité et coût, mais aussi sur son efficacité énergétique, inclusivité, potentiel de partage et émissions de gaz à effet de serre. Étant donné la complexité des systèmes de mobilité et l'important que joue le transport dans toutes les activités humaines, la mobilité n'est plus seulement un produit de recherche dans d'autres domaines technologiques. La mobilité se tient aujourd'hui comme un domaine de recherche à part entière. À l'intersection de nombreux domaines, la mobilité en tant que thématique de recherche est une discipline transdisciplinaire qui implique les sciences sociales, l'informatique, l'ingénierie et tant d'autres. Des outils toujours plus complexes sont mis en œuvre pour investiguer des sujets de mobilité et répondre à des questions qui vont du très stratégique (tel que l'acceptation des utilisateurs pour une technologie encore loin de la maturité) au très opérationnel (tel que le prix à mettre sur un service de mobilité). Un des outils utilisés de manière intensive consiste en la construction d'un modèle des aspects de la mobilité puis sa simulation dans un environnement informatique pour en observer le comportement. Ceci permet l'évaluation de l'impact d'une large étendue de variations des entrées du modèle. Nous sommes particulièrement intéressés par les modèles et les simulations basés sur les agents, ou dits multi-agent. Dans ce type de modèles, les concepts à niveau fin tel que voyageur,

véhicule, route... sont modélisés à un haut niveau de détail. Ceci permet des études localisées dans lesquelles la mobilité d'un territoire donné est simulée et différentes solutions de mobilité sont évaluées par rapport à leurs impacts sur les usagers, les opérateurs ainsi que l'environnement. Aujourd'hui, les simulations multi-agent de mobilité sont utilisées pour différents types de recherches autour de la mobilité. Parmi les solutions de mobilité évaluées à l'aide de simulations multi-agents de mobilité, on trouve les systèmes de mobilité à la demande. Ces systèmes sont permis par des flottes de véhicules qui répondent à des requêtes de trajets émanant de différents utilisateurs au cours de la journée. Le potentiel de partage ainsi que la flexibilité rendus possibles par ces systèmes en font une alternative intéressante à la possession d'un véhicule privé. Ce qui peut contribuer à répondre aux défis environnementaux liés à la mobilité. Dans cette thèse, nous nous intéressons aux approches multi-agent pour la simulation de la mobilité en général et plus particulièrement de la mobilité à la demande. L'intégration de la mobilité à la demande dans une simulation réaliste, tant du point de vue de la demande que de l'offre, où elle est utilisée en intermodalité avec les transports publics constitue un objectif principal de ce travail. Une telle intégration est nouvelle au regard de l'état actuel de la littérature et soulève des défis méthodologiques et techniques. L'état de l'art dans ce domaine est étendu sur plusieurs fronts : l'intermodalité, l'opération des services de mobilité à la demande et l'impact sur les choix des voyageurs. Bien que nos approches soient génériques et peuvent s'appliquer à des cas d'usage divers, nous les illustrons principalement sur la région Île-de-France et plus précisément sur le territoire de Paris-Saclay. Nous tentons, par ce travail, de répondre aux questions de recherches suivantes : (i) Quels impacts, sur les voyageurs, auront les futures lignes de transports ferroviaires qui sont prévues sur la zone d'étude ? (ii) Le niveau actuel de définition de l'offre future permet-il d'estimer les impacts de cette offre de manière complète ? (iii) Comment est-ce que l'architecture existante de simulation multi-agent de mobilité, ainsi que le modèle de choix associé, peuvent être étendus pour permettre l'intégration de systèmes de mobilité à la demande intermodaux ? (iv) Quelle serait l'efficacité d'un système de mobilité à la demande intermodal sur la zone d'étude ? Quel dimensionnement est requis pour que ce dernier puisse bien servir la demande ? Et comment mieux prendre en compte les dimensions du système dans le modèle de choix ? (v) Comment est-ce que l'apprentissage par renforcement pourrait aider à mieux opérer ce type de systèmes ? Quel est l'état actuel de la littérature ? (vi) Les outils existants de simulations multi-agent peuvent-ils être adaptés pour permettre l'intégration d'algorithmes à base d'apprentissage par renforcement pour la mobilité à la demande ?

Les questions de recherche ci-dessus sont adressées avec un accent sur l'ouverture des données et des outils et la réplicabilité des résultats.

Contents

List of Figures

List of Tables

1 - Introduction

1.1 . About mobility

Human history can be, and often is, told as a story of mobility. We as humans need to move ourselves as well as our goods. As a result, the evolution of human societies has always been intertwined with the evolution of the means by which people move. It is only natural that mankind has always looked, and is still looking, for ways to travel faster, with increased efficiency, safety, comfort and decreased cost.

First, the human body was the main mean of transportation. Later, it was the bodies of other animals and then vehicles carried by animals. Since the industrial revolution, mobility evolved to rely on vehicles powered by an energy source rather than animals. And each evolution of the energy source that is used was immediately followed by a revolution in mobility both in terms of quality and in availability for the population. On the other hand, the more elaborate the transportation mean, the more complex the management of transportation and the challenges associated to it.

Today, mobility is the lung of modern civilization. A good quality of a transportation system is considered to be a good reflection of a modern society, and its good functioning is necessary for the productivity of other sectors such as industry and tourism. With most of the people living in cities, challenges that are faced by cities are very often closely related to urban mobility (This is the case for pollution, noise and accessibility of public services).

One of the criteria of a good transportation system, whether it is based on private vehicles or shared ones, and whatever technologies it is based on, is the availability of real-time information regarding the state of the system. Thanks to the major advances in information and communication technologies, this availability of information is now taken for granted. Using a smartphone, one can retrieve the fastest route to reach a destination with a wide range of modes and taking into account the current situations of road congestion and public transports' temporary perturbation. Powered by these technological developments, urban mobility now not only consists of a road network, private cars, public transports and active modes, it also comprises a wide spectrum of flexible modes powered by free floating bikes, scooters, ride-hailing and car-sharing platforms.

These new modes of transportation come with a promise of more efficiency and sharing. In a period of time where the limited nature of available resources has come to light and the negative impacts of human activities on the environment can no longer be denied and require urgent mitigation, sustainability has become a core focus of mobility related developments. A mobility solution is no longer just evaluated by its speed, safety and cost, but also its energy efficiency, inclusiveness,

potential for sharing and GHG emissions.

Given the complexity of mobility systems and the importance that transportation plays in all human related activities, mobility is no longer just a product of research in other technological fields. Mobility has become a field of research in and of itself. At the intersection of a wide range of fields, mobility as a research area is a trans-disciplinary domain that involves social sciences, computer science, engineering and many others. Increasingly more complex tools and methodologies are being put to use to investigate mobility topics and answer questions that range from the very strategic, as what would be the user acceptance for a technology that is still far from mature, to the very operational, like what price to put on a given mobility service.

One of the tools that are used extensively in mobility studies consists in building a model of mobility related aspects and then running computer simulations to observe the behavior of the model. This allows to assess the impact of a wide range of changes in the model inputs. We are particularly interested in agent-based models and simulations of mobility where low-level notions of traveler, vehicle, road... are directly considered in detail. This allows for localized studies in which the mobility of a given territory is simulated and different mobility solutions evaluated regarding their impact on users, operators and the environment. Today, agent-based mobility simulations are used for various types of mobility researches.

Among the mobility solutions evaluated using agent-based simulations are mobility-on-demand systems. These systems are ensured by vehicles that respond to trip requests emanating from different users throughout the day. The sharing potential as well as the flexibility made possible by these systems make them a possibly interesting alternative to owning a private vehicle and can help addressing the environmental challenges raised by mobility.

This thesis is hosted by the Anthropolis Chair [1], operated jointly by IRT SystemX and LGI (Laboratoire Génie Industriel) at CentraleSupélec since 2019. This research project focused on mobility employs various qualitative and quantitative methodologies to investigate three topics: future mobility and urban life, mobility as a service and future infrastructure. Agent-based simulations are used to answer research questions on the three aspects in relationship to mobility-on-demand, many of these questions lay in the scope of this thesis.

In this thesis, we focus on agent-based approaches for simulating mobility in general and mobility-on-demand systems in particular. The integration of mobility-on-demand in a realistic simulation, both in terms of mobility demand and offer, where it is used in intermodality with regular public transports constitutes a core objective of this work. Such an integration is novel in regards to the current state of the literature and involves methodological and technical challenges. The state of the art in this topic is extended on various fronts: intermodality, service operation, impact on user choices. Another major focus of this PhD project is

[1] https://www.chaire-anthropolis.fr

the investigation of the use of reinforcement learning based approaches for the operation of mobility-on-demand systems.

While our approaches are generic and can apply to various use cases, we illustrate them on a localized setting as we focus on the area of Île-de-France and more precisely on the territory of Paris-Saclay. This territory is interesting to us for three reasons: (i) it presents research opportunities as it is the subject of developments regarding the offer of public transports with new rail-based lines being built. (ii) We emphasize the importance of reproducibility of our work, which requires openness of the tools, approaches and data that we use. The availability of a wide range of open data sets in France has allowed the research community to build reproducible work that can be extended. (iii) The Anthropolis Chair hosting this PhD project has among its partners the inter-council partnership in charge of the planning in the area (including mobility). This presents an opportunity of acquiring additional non-open data as well as the ability to discuss and get feedback on simulation results from a party that has a knowledge of real-world territory. We consequently build a prospective simulated use case for the study area which includes future rail lines.

More precisely, we intend, in this thesis, to provide answers for the following research questions:

1. What impact will have the future planned rail-based lines in the study area on traveller trips and their modal decisions?
2. Does the currently available level of planning regarding the future offer allow to fully assess the impact of these future lines?
3. How can the existing open-source agent-based mobility simulation architecture, and the associated traveller mode choice model, be extended to support the support of intermodal mobility-on-demand?
4. How would an intermodal mobility-on-demand system perform on our study area? How should the system be sized to appropriately satisfy the demand? And how to better take into account the dimensions of the system in the mode choice model?
5. How can reinforcement learning help to better operate such systems? What is the current state of the literature?
6. Can the existing agent-based mobility simulation framework be adapted to accommodate such reinforcement learning algorithms for future testing?

1.2 . Contributions

With the research questions listed above in mind, the work performed throughout this thesis presents several contributions:

1. A first agent-based simulation assessment of the impact of future rail-based lines planned for the study area is presented. This assessment merges open data regarding future developments with existing open public transport schedules as simulation inputs and shows the missing items for more realistic analyses.

2. An existing open-source simulation framework supporting mobility-on-demand systems is extended to support intermodality between mobility-on-demand and public transport. The mode choice model is accordingly extended to consider intermodal trip alternatives.

3. An intermodal mobility-on-demand system is studied alongside existing and future rail-based public transport lines. Its potential for further increasing the attractiveness of public transports and to mitigate the absence of data regarding future complementary bus offers.

4. Responding to a need that was discovered during the thesis project, a novel approach is proposed for taking into account the low availability of a mobility-on-demand system with rejectable requests in model choice models.

5. A literature review on the use of reinforcement learning methods for mobility-on-demand systems. While presenting the state of the art in a high-level manner, our review also dives deep in the algorithmic details of the approaches and discusses methodological differences and evaluation methodologies to connect the transport simulation and reinforcement learning research communities. Gaps and development pathways are identified in all these aspects.

6. A software architecture allowing to easily implement reinforcement learning algorithms for mobility-on-demand within an open-source agent-based simulation framework and compare them in different use cases is developed and tested.

1.3 . Structure of the thesis

The following of the manuscript is organized as follows:

Chapter 2 presents a background of the concepts and methodologies that are used throughout the PhD work. First we define MoD systems as they are considered in our work. Second, we give an overview on agent-based simulations of mobility and the various tools and frameworks that are available as well as a detailed description of the solutions used in the scope of our research. We then present a literature review of use of Reinforcement Learning based methods for the operation of MoD systems by detailing the approaches presented a some of the

related works and comparing them in terms methodology, use case, and algorithmic details (Contribution n°5). The algorithms are located and compared in a novel framework for sequential decision making under uncertainty. This literature review is the subject of a journal paper which is, at the time these lines are written, under review at the Transportation Research Part C: Emerging Technologies.

We dive in Chapter 3 in the development of the Île-de-France and Paris-Saclay simulation use case. We first present the area, its current mobility situation and the planned developments. We then detail the technical aspects related to this part of the work, the generation of the synthetic population (mobility demand) and the integration of current and future public transports (mobility offer). The simulation scenarios and evaluation methodologies are presented before detailing and discussing the simulation results. This constitutes the contribution n°1 announced above.

Chapter 4 details three of our contributions. First, our work related to the implementation, in simulation, of an intermodal MoD system and its evaluation is presented (Contributions n°2 and n°3). This first part of this chapter as well as the work presented in Chapter 3 are presented in a conference paper submitted to the 102^{nd} annual meeting of the Transportation Research Board (TRB 2023) Chouaki et al. (2023). Later in Chapter 4, we present a novel approach to better take into account the impact of performance of an MoD system on travelers choices and consequently the demand for the system. We then detail the implementation of our approach in the simulations. Finally we present and discuss the obtained results of simulations of MoD systems that build on the use case developed in Chapter 3 (Contribution n°4). This approach is first presented in a conference paper submitted to the 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS) and will be presented between 14 and 16 June 2023.

Chapter 5 presents a reinforcement learning approach for the operation of the simulated MoD system. A novel architecture for reinforcement learning in MATSim is proposed. We show the feasibility of studying and evaluating such an approach in our simulation environment under a simple use case (Contribution n°6). This work is the scope of a conference article submitted to the 13th International Conference on Ambient Systems, Networks and Technologies (ANT 2022) Chouaki et al. (2022).

Chapter 6 concludes this manuscript by summarizing the contributions of this thesis and the works that have been bootstrapped as a result of the technical developments and methodological contributions of this thesis.

2 - Background

In this chapter, we present the key concepts that will serve as a foundation to the work presented in the remainder of the manuscript. We first describe the notion of a mobility-on-demand system and the decisions involved in designing such a system. We then present agent-based modelling and simulation as a tool that is used for mobility related studies in general and for the design evaluation of mobility-on-demand systems in particular. We then present in details the platform that is used in this PhD project. In section 2.3, we give an overview of reinforcement-learning algorithms and sequential decision making methods and in section 2.4 we present a literature review of the use of reinforcement learning in the design of mobility-on-demand systems. These two sections mainly draw from a journal article that has been submitted to Transportation Research part C: Emerging Technologies and is currently under review.

2.1 . Mobility-on-Demand

The concept of Mobility-on-Demand (MoD) referred to all along this thesis defines mobility systems that consist in a fleet of vehicles that are available to users when needed, hence on-demand. These vehicles can be booked by customers at time of departure, user requests are handled by a central fleet operator that assigns a vehicle (if any is available). Trips can be shared between multiple users if they have origins and destinations that are close enough. Therefore MoD systems are also often referred to as ride-pooling or shared mobility.

The advances made in information and telecommunication technologies have facilitated the implementation of MoD systems that can be accessed via mobile applications with geolocation capabilities. The assigned vehicle's driver is then able to travel to pick-up the traveller at his or her exact location and then ride to the drop-off location. Such services are implemented by various operators throughout the world (Uber, DiDi, Lift, Heetch...) (Schaller, 2018).

With the evolution of autonomous driving technology, MoD is expected to evolve into Autonomous MoD (AMoD): a system consisting of a fleet of fully autonomous and connected vehicles that alleviate constraints related to human drivers. Mainly, an AMoD system would be able to ensure a consistent offer throughout the day as the vehicles (often called robo-taxis in this case) do not depend on the working schedules of human drivers. This increases the sharing potential presented by MoD systems and can further reduce roads congestion, GHG emissions and the number of privately owned vehicles. These AMoD (Autonomous MoD) systems are increasingly studied in the literature. These researches span the ethical implications of AVs (Martinho et al., 2021), their effects on road infrastructure planning policies (Storsæter et al., 2021), the users' readiness for AMoD

services and their sustainability impacts (Cugurullo et al., 2020; Williams et al., 2020), their interaction with other modes of transports (Pinto et al., 2020; Heilig et al., 2017; Hancock et al., 2019) and the overall impact of these systems on people's mobility (Pernestål and Kristoffersson, 2019; Hamadneh and Esztergar-Kiss, 2019; Harper et al., 2018). The research in this area has matured enough to reach the design of these services on the operational level while taking into account close to real-life settings (Golpayegani et al., 2022). Gurumurthy et al. (2019) presents advances made in the literature across various aspects of AMoD systems. Our work presented throughout this manuscript applies to both MoD and AMoD systems.

The design of MoD systems, whether they rely on autonomous vehicles or not, involves different decisions on various levels. The fleet size, vehicle characteristics, pricing, service area, operational strategy, all have an impact on the performance and attractiveness of these systems towards users (Vosooghi et al., 2019b). Operational strategies or policies refer to how a MoD system performs the day-to-day trip-level decisions. These consist in two main tasks: (i) Empty vehicle rebalancing: deciding where on the network empty vehicles relocate when they are not serving a user request, including recharging decisions for electric vehicles and (ii) Vehicle assignment: deciding which vehicle should be assigned to what request(s). In order to perform the best decisions, these tasks are considered as optimization problems in which a certain value is maximized (number of travellers, revenue) or minimized (waiting times, travel times, vehicle kilometers, cost).

Consequently, the design of MoD systems and the methodologies for assessing the performance of different designs constitute a very active area of research that spans the field of economics, simulation, artificial intelligence, optimization

2.2 . Agent-based modelling and simulation as a tool to study mobility systems

2.2.1 . Agent-based modelling and simulation in general

Complex systems, particularly real-life ones, involve a large number of entities interacting in a local manner. These local interactions affect the macro-level properties of the system. Deriving the macro-level properties from the micro-level interactions is not a straightforward task. This is due to the phenomenon known as "emergence" where effects that are not intuitively expected on the macro level arise from seemingly independent local interactions. An example of this is the ability of ants to coordinate on a large scale and efficiently find shortest paths to resources and gather them by only relying on pheromones left in the environment. Consequently, in many problems, it is necessary to simulate the entities individually with their local interactions in order to observe the global effects.

Such systems are known as multi-agent systems. The challenges related to modelling, simulation and management of these systems emerged as major field of research (Uhrmacher and Weyns, 2009). In a multi-agent system, the atomic entity

is the agent. An agent perceives the environment (including other agents) and acts according to its own goal. Consequently, the agent's actions are influenced by and also influence the environment. The interaction between agents can be direct (by exchanging messages) or indirect (leaving hints in the environment). When the objectives of the agents align with each other, the system is said to be cooperative, otherwise it is competitive. Agent-based modelling and simulation approaches are used in many fields as they allow to work on intuitive concepts.

In the next section we focus on the use of agent-based modelling and simulation for mobility and present the simulation framework that is used in our work.

2.2.2 . MATSim

In the context of mobility in particular, the use of agent-based approaches is increasingly popular. Already in 1997, Burmeister et al. (1997) identified the potential of multi-agent systems and agent-based simulations for mobility and transportation related domains. Balmer et al. (2004) presents the concept of an agent-based mobility simulation platform. Today, a wide range of agent-based mobility simulation tools are available, with a few of them being open-source (Abar et al., 2017).

In our work, we use the MATSim simulation framework (Horni et al., 2016). MATSim is a fully open-source tool in constant evolution with an extended community supporting it. Figure 2.1 presents an overview of the MATSim simulation process alongside the main input data for the simulation and output information obtained after the simulation. In the following we detail the MATSim platform and its features as the assumptions inherent to the tool and level of detail apply to all the work performed in this thesis.

Inputs

The minimum required inputs to run a simulation in MATSim are the network and the mobility demand. In MATSim, the road network is a directed graph where each edge, called *link*, represents a road and each *node* represents an intersection between roads. Nodes have coordinates, links have a free speed, allowed vehicle types and a capacity (maximum number of vehicles that can be present on the link at any given time). Every element of the simulation is located on a link. MATSim provides a tool for converting OpenStreetMap networks to MATSim networks, making it easier to build simulations on realistic networks.

Arguably the most important (and the most challenging to obtain) input for an agent-based mobility simulation is the mobility demand. In MATSim this is referred to as the *population*. Each traveller in MATSim is modelled as a separate *agent*. MATSim is *activity*-based in the sense that each agent has a sequence of activities to perform throughout the simulated period (typically one day). These activities are located in space and in time, consequently an agent needs to travel in order to perform the activities. For each trip between two consecutive activities,

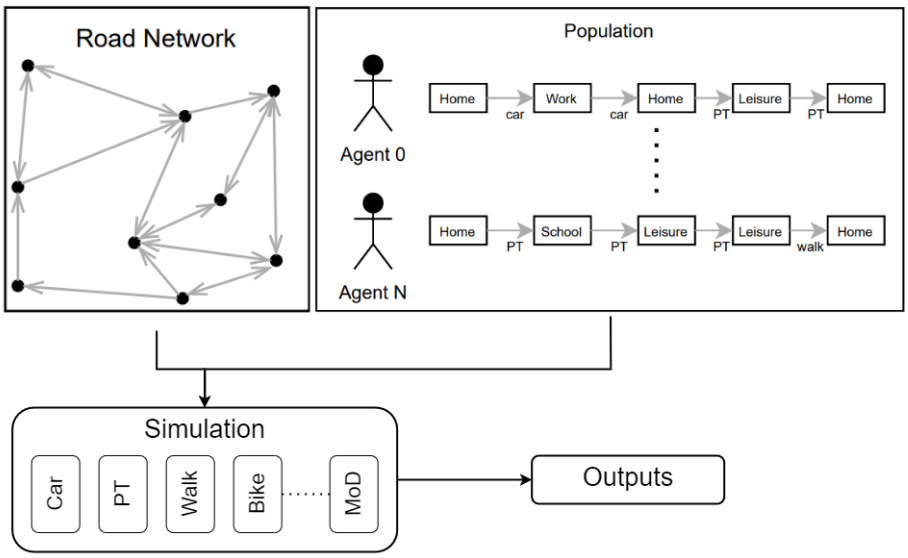

Figure 2.1: Overview of MATSim

the agents can use one of the modes available in the simulation. The activities and the travel routes between them constitute the agent *plan*. Tools for generating *synthetic populations* from real-world data are proposed in the literature. More on the approaches used in this work can be found in Chapter 3.

Alongside the network and the population, other inputs can be supplied to MATSim according to the user's needs. MATSim is built in a modular manner which allowed the community to contribute with various extensions adding features to the tool. It is possible to simulate Public Transport (PT) systems by supplying a MATSim transit schedule specifying the stops, lines and departure times. A tool for converting a PT schedule in the GTFS format to a MATSim transit schedule is provided alongside MATSim.

Another extension of MATSim, known as the DRT module (Bischoff et al., 2017), allows to simulate MoD systems by specifying the fleet characteristics: a list of vehicles each with a capacity, a service time period and an initial location. The user can also choose the operation strategies for the rebalancing and vehicle assignment tasks among the algorithms implemented in the module.

The main entry to a MATSim simulation is a configuration file which specifies general and module-specific parameters and points to other input elements (population file, transit schedule, MoD fleet...). MATSim uses XML as a format for input files and most output files.

Simulation

The simulation consists in running all the agent plans, i.e. travelling between activities using the indicated modes and following the indicated routes. Some

travel modes (car, PT, MoD) involve using a vehicle that travels throughout the links of the network while trips of other modes are simply teleported because they are assumed to not generate any congestion (walk and bike). During the simulation, each link is processed as a queue with a maximum throughput speed (number of vehicles that can leave the link per time step) and maximum capacity (number of vehicles that can be in the queue at any time). Consequently a vehicle can leave a link only if the next link on its route is below maximum capacity, which causes congestion to emerge and propagate through the network. This level of abstraction differs from other agent-based simulation tools such as SUMO (Lopez et al., 2018) where roads are more precisely modelled and acceleration and steering angles are computed for every vehicle at every time step. The simplification made in MATSim allows it to be able to perform simulations on much larger scales with a higher number of vehicles.

Public transport is simulated following the information specified in the provided MATSim transit schedule. Each PT line has a set of routes, each serving a sequence of transit stops (where travellers get in or out of PT vehicles). Each route has a list of departures, i.e. times at which vehicles depart from the beginning of the route. PT users walk towards the departure stop at which they wait for the PT vehicle. Transfers between different PT lines through walking are allowed. The PT vehicles can have limited capacities, which if taken into account can lead travellers to wait more. Moreover, PT vehicles can compete with other vehicles on the network thus possibly generating delay due to congestion. However they can also be teleported in order to ensure that the schedule is fully respected.

A MoD system, implemented by the DRT module, is simulated by moving vehicles through the network by a central operator. A vehicle starts at its initial position specified in the input file and awaits the orders of the operators. An order can be : (i) Pickup passenger at a given location (ii) Dropoff passenger at a given location (iii) Remain idle at the current location (iv) Rebalance to another location. The first two decisions are part of the vehicle-assignment decisions in which passenger requests are matched to vehicles. The operator uses an insertion heuristic to identify where to insert the pickup and dropoff related to the request in the vehicle's schedule in order to achieve the earliest arrival time. MoD trips can be shared, i.e. more than one passenger can be present in a vehicle at a time. Ride-sharing related detours can result in longer trips for passengers. To mitigate this issue, requests can be rejected if no insertion that meets criteria of maximum waiting time and maximum detour factor (the difference of distance between the actual trip and the unshared one). These rejection criteria can be adjusted through the configuration. Due to the possibility of rejecting requests, the rebalancing strategy is of key importance since an available vehicle needs to be near enough from the pickup location to ensure a good waiting time. The pickup and dropoff locations can either be the origin and destination of the user's trip if the service is operated in a door to door manner. If it is operated in a stop based

manner, pickups and dropoffs can only occur at certain stopping places (given as inputs). In this case, the passenger takes the MoD vehicle from (resp. to) the closest stopping place to the origin (resp. destination).

Replanning

MATSim's key feature is certainly the replanning step and the iterative manner in which simulations take place. This allows agents to change their plans according to observed performance in previous iterations in order to build better plans on the individual level. Consequently, the changes in agent plans at the end of the simulation reflect the impact of the mobility offer on the user choices.

The performance of an agent plan depends on other agents' plans. For instance, the travel time of a car trip depends on the congestion on the route which in turn depends on the number of travellers on the route. In order to prevent oscillation effects in the plans, not all agents should perform replanning at every iteration. Consequently, a certain proportion (usually 5%) of the population is randomly selected at the beginning of every iteration for replanning. The other agents carry on with the same plans. Consequently, several iterations are needed to reach system equilibrium.

The default MATSim replanning method consists in maintaining a list of plans for every agent. Each plan is assigned a score reflecting the performance observed by running the plan. The score aggregates the travel times, wait times, delay times regarding arrival at activities and various other metrics each specifically weighted before being summed. At the end of an iteration, the scores of performed plans are updated using the latest observed values for each component. If the agent is selected to perform replanning, an innovation strategy is used to generate a new plan from existing ones. Different innovation strategies can be enabled (e.g. changing the transport mode for a trip, delaying activity times). If the maximum size of the plans list is reached, the plan with the smallest score is removed. More details about plan scoring in MATSim can be found in Horni et al. (2016)

The MATSim scoring approach for agents replanning offers the advantage of taking into account several aspects of the plan performance. The drawback of this is that the weights related to each parameter need to be calibrated properly. Moreover the strategies for generating new plans explore a large space of possible plans, including ones that are highly unlikely (e.g. walking for a very long distance). This increases the time needed for the system to converge to equilibrium.

In order to mitigate the previously mentioned issues, another approach for agent plans replanning that relies on Discrete Mode Choice (DMC) models has been proposed into MATSim (Hörl et al., 2019). This approach exclusively focuses on chosing the transport modes for the trips. When an agent is selected for replanning, its journey through the simulation is divided in tours that start and end at a *home* activity. E.g. a sequence home→school→shop→leisure→home→leisure→home is divided into two tours home→school→shop→leisure→home and home→leisure→home.

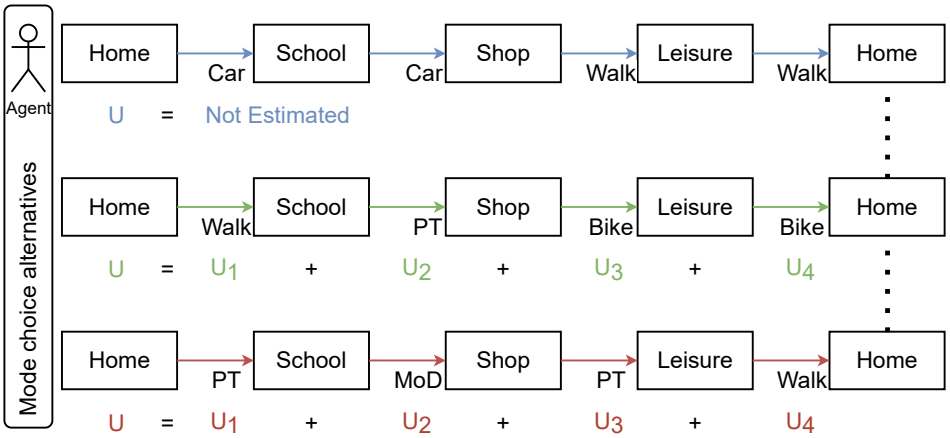

Figure 2.2: Summary of the DMC model implemented in MATSim. For a tour starting at ending at the activity type "home", All possible mode sequences are considered. In this example, the utility of the top alternative is not evaluated due to breaking the constraint that states that the car must be at the home location at the end of the tour

Afterwards, and for each tour, all the mode sequence alternatives are considered. A first filtering is performed to filter-out the alternatives that violate the pre-evaluation constraints (e.g. taking car without possessing a license). Then, utilities are computed for each tour alternative as the sum of utilities of its trips as illustrated in Figure 2.2. The utility of a trip depends on the considered mode but generally involves computing the trip route and estimating travel time, wait time and cost. These are weighted and summed with a modal constant to obtain the utility of the trip. A second filtering takes place to rule-out tour candidates that violate post-evaluation constraints (e.g. utility below a certain threshold or a PT trip route that only consists in walking). Among the remaining candidates, one is chosen using a selection method based on the estimated utility. The most often used method is the Multinomial Logit selection (Train, 2009).

Using the DMC model approach described above, calibration is made easier by focusing only on the transport modes and filtering out "bad" plans beforehand, only acceptable plans are simulated. This reduces the number of iterations that is needed to reach the equilibrium.

Outputs

One advantage of using an agent-based simulation is the possibility of observing every event that happens on the micro-level of the simulation. MATSim writes by default the events of the last iteration into an XML file at the end of the simulation. These observations can then be aggregated to obtain desired metrics

on different levels such as time periods, links, agents. Moreover, MATSim outputs the resulting agents' plans in the same format as the input. This allows to analyze the plan changes on the agents level.

However, a few aggregated results and visuals are generated by MATSim by default such as the evolution mode shares across the iterations, the distribution of number of trips throughout the day. This allows non-expert users to directly have an overview of a simulation's results.

Conclusion

We conclude this section by describing a general workflow for agent-based mobility studies using MATSim and DMC models depicted in Figure 2.3. First, a synthetic population reflecting the current state of the demand is generated, the parameters of the DMC model are then calibrated on a selected set of metrics to reduce the gap between the simulation and the real-world in regards to these metrics. When the generated synthetic population is simulated on the real world mobility offer (imported network and transit schedule), the simulation should reflect the current state of mobility, thus making a baseline scenario. One or more components of the scenario, on the demand side or on the offer side, can then be modified to obtain one or more prospective scenarios reflecting possible futures that are studied. The impact of the changes that are introduced can then be assessed through comparisons between the outputs of the baseline and prospective simulations.

2.3 . An overview on Learning based methods

In this section, we first focus on Reinforcement Learning (RL) and present the different types of approaches that we find in the literature of MoD. We then present a recent framework under which methods for sequential decision-making under uncertainty can be classified and compared.

2.3.1 . Reinforcement Learning

The term reinforcement learning refers to three separate but related aspects: a class of problems, the algorithms for solving those problems and the field of research that studies the design of algorithms to solve these problems(Sutton and Barto, 1998).

A RL problem is any problem where we consider an *agent* or multiple agents evolving in an *environment* and interacting with it through a loop of *actions* performed by the agent(s) and a *reward* signal returned by the environment indicating how well the agent has performed. More formally, RL problems can be represented with Markov Decision Processes (MDP)(Littman, 1994) which consist of the following components: (i) a set of states S, (ii) a set of actions A, (iii) a transition function $p(s'|s,a) = Pr(S_{t+1} = s'|S_t = s, A_t = a)$ where S_{t+1}, S_t and A_t respectively indicate the state at step $t+1$, the state at step t and the action taken

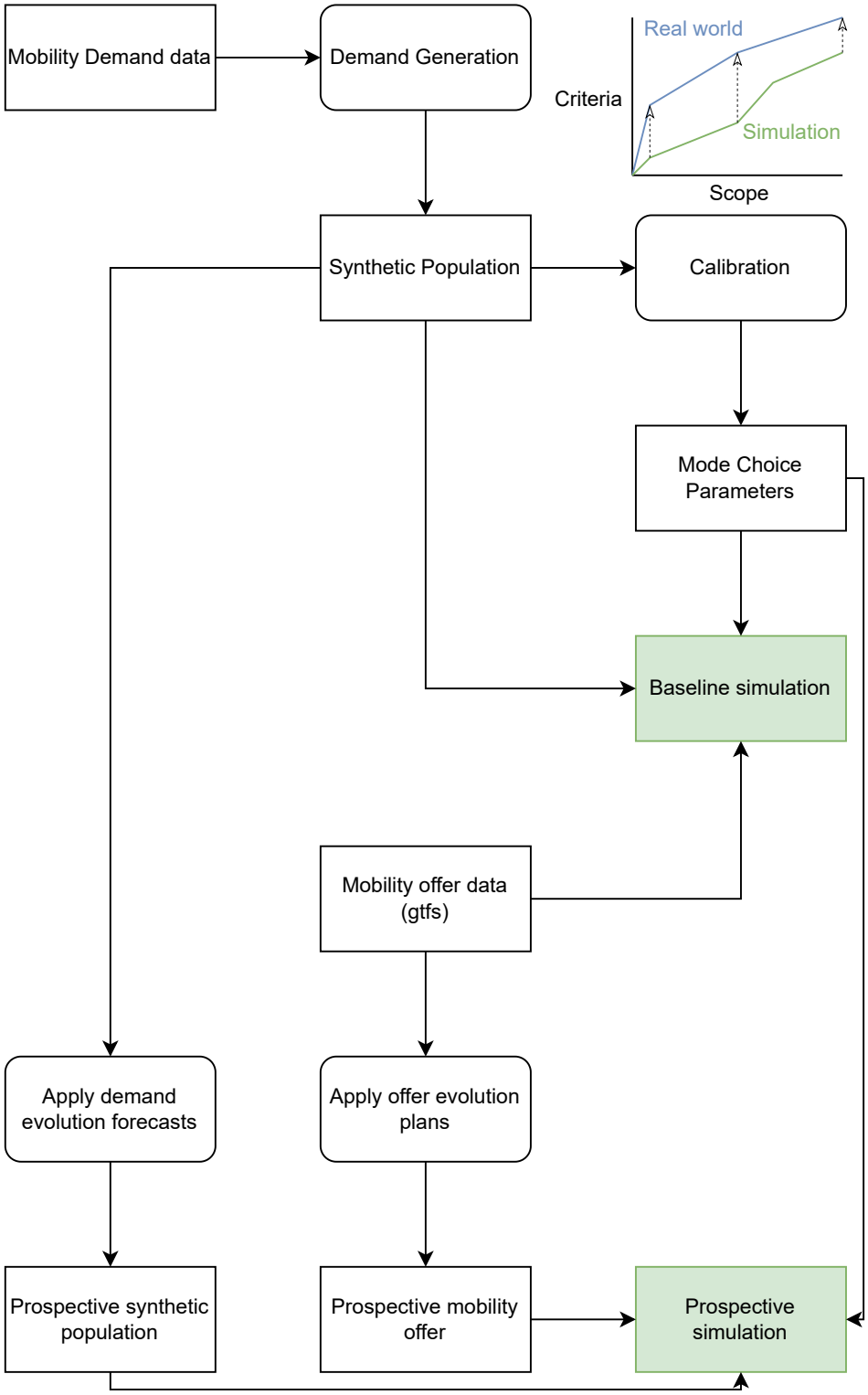

Figure 2.3: A general workflow for agent-based mobility studies using MATSim and DMC models

at step t after observing S_t and before observing S_{t+1}, and (iv) a reward function $r(s, a, s') = \mathbb{E}(R_{t+1}|S_t = s, A_t = a, S_{t+1} = s')$ where R_{t+1} indicates the reward obtained at time $t + 1$ after observing S_t and performing A_t and arriving at S_{t+1}.

The agent then needs to learn, through trial and error, a *policy* $P(s, a) = Pr(A_t = a|S_t = s)$ specifying which action to take given the *state* of the environment so as to maximize the long-term reward $R_1 + \gamma R_2 + \gamma^2 R_3 + ...$ The parameter γ is called the *discount factor* and specifies the relative importance of future rewards with regard to the immediate reward. In general, actions do not only affect the reward that immediately follows, but also the whole chain of reward signals that come after it. An RL algorithm, then, is a method that uses the agent's interaction with the environment and the obtained rewards to learn better policies over time. Consequently, RL is a particular type of Machine Learning where the data acquisition is part of the learning process (Sutton and Barto, 1998).

In RL, the learning agent itself is not necessarily aware of the MDP, or more particularly the transition function. This sets the first distinction between two main approaches of addressing an RL task. They consist of:

(1) Model-based RL approaches use a description of the environment (transition and reward function) that is either known beforehand or learned through experience. In the latter case, Dynamic Programming methods can be used to build an optimal policy. However, having a model of the environment is typically not an option in many use cases, especially if the problem itself changes over time.

(2) Model-free RL approaches do not take into account an explicit model of the environment but rather use trial and error to estimate the state value function $V(s)$ which indicates the expected long-term reward that can be obtained from encountering the state s. Alternatively, the action-state value function, $Q(s, a)$ (also called Q-function), has been used in several RL algorithms. It indicates the expected long-term reward that can be obtained by performing action a from state s. Values of such a function are called Q-values.

How value functions are updated then constitutes another way of categorizing RL approaches. A very popular approach is Temporal Difference Learning, where the value of a certain configuration ($V(s)$ or $Q(s, a)$) is updated after encountering the given configuration and then observing the received rewards over a certain time period. The value of the configuration is then adjusted to better match the observed reward sequence and taking into account the expected long-term reward from the end of the sequence. In on-policy methods, the update rule supposes that future actions are taken using the current policy, whereas off-policy methods do not. A well-known example of an on-policy temporal difference learning is SARSA (state-action-reward-state-action), for which the update rule is:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{s+1}, A_{t+1}) - Q(S_t, A_t)]. \quad (2.1)$$

In this equation, the new Q-value of the pair (S_t, A_t) is computed by shifting the old value towards the most recent observation, which consists of the immediately

obtained reward and the estimation of the long-term reward that can be obtained from the new state S_{t+1}. Note that to perform the update, A_{t+1} must be already selected (by the policy). In contrast, the popular off-policy temporal difference learning algorithm, called Q-learning, uses the following update rule:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{s+1}, a) - Q(S_t, A_t)]. \quad (2.2)$$

It is supposed that the value function is kept in a tabular presentation (the value for each possible input of the value function is stored). The optimal policy can then be derived from the converged Q-values by selecting the action that maximizes the long-term reward from the current state.

In the learning process, however, it would not be effective to always select the current best action; this would result in the algorithm only exploiting a subset of the potential solutions and not exploring other areas. This is known as the exploration-exploitation dilemma, and various strategies have been used to address it. One of them is the $\epsilon-$greedy method, where at each decision step, a random action is selected with probability ϵ and the current best action is selected with probability $1 - \epsilon$.

Given the possible multidimensional nature of the state values (e.g. a vehicle's state can be comprised of its location, number of passengers, state of charge and current time), in most problems, a tabular representation is difficult to achieve, since the state space can be extremely large. Especially if the state is a vector containing different features, then the size of the state space is exponential to the number of features. This is known as the curse of dimensionality. Another drawback of a tabular representation is that the value of a particular configuration will remain completely unknown until it has been encountered at least once. No generalization is performed from the potential encounters of similar configurations. To solve these issues, value function approximation methods are used. Value functions can be approximated using a more compact representation, for which the number of parameters is considerably less than the number of possible configurations. The approximation then needs to be updated to minimize the error between the observed rewards and the estimated long-term values. Even though each update is typically performed using a given configuration, it updates the values of other configurations. A wide range of value function approximators have been explored in the literature, from linear and polynomial models to Fourier bases and coarse coding. Geist and Pietquin (2013) present a review on the different methods of approximating value functions that are used in RL, along with advantages and drawbacks of each method.

Recently, a particular kind of approximation has become increasingly popular, that consists of Artificial Neural Networks (ANNs). An ANN, with multiple layers (called deep ANN) can be used to approximate a very wide range of non-linear functions and has proven to be very efficient in supervised learning tasks (in which the learning is performed from labelled data). The interest of using deep ANNs

for RL gave rise to the branch of Deep-RL, which is currently extensively used. Q-learning based methods where the Q-function is represented with a deep neural network (deep Q-network) are called deep Q-learning methods.

In contrast to extracting the policy from the learned value function, another approach is to represent the policy as the parameterized function $\pi(a|s,\theta) \in [0,1]$ (indicating the probability of selecting an action given the state and the policy parameters). The parameters θ of the policy are then updated during the interaction with the environment in an attempt to converge to the optimal policy. RL methods using this approach are known as Policy Gradient Methods, among which we find the generic REINFORCE algorithm (Williams, 1992) which involves generating multiple sequences of states, actions and rewards (keeping the policy constant during each sequence) and then using the following update rule for θ.

$$\theta_{t+1} = \theta_t + \alpha G_t \frac{\nabla_\theta \pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)} \tag{2.3}$$

Where $G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \ldots\ldots$ is the sum of rewards in the following of the sequence. This update rule updates the policy parameters such as to increase the probability for an action at a certain state if it yielded a positive cumulative reward during the sequence and decreases it otherwise.

Actor-Critic methods (Konda and Tsitsiklis, 1999) combine a parameterized policy function (actor) with a parameterized value function (critic) $v(s,\mathsf{w})$ that is learned and used to guide the update of the policy. The one-step actor critic algorithm, which does not require sampling a whole episode before the update. It uses $\delta_t = R_{t+1} + \gamma v(S_{t+1},\mathsf{w}_t) - v(S_t,\mathsf{w}_t)$, which indicates the difference between the observed and expected reward, to update both the value function and the policy function.

$$\theta_{t+1} = \theta_t + \alpha^\theta \delta_t \frac{\nabla_\theta \pi(A_t|S_t,\theta_t)}{\pi(A_t|S_t,\theta_t)} \tag{2.4}$$

$$\mathsf{w}_{t+1} = \mathsf{w}_t + \alpha^\mathsf{w} \delta_t \nabla_\mathsf{w} v(s,\mathsf{w}_t) \tag{2.5}$$

This method presents the advantage of taking into account the experience accumulated by the critic throughout the whole learning process, thus reducing the chances of iterating around a local optimum for too long.

2.3.2 . Sequential decision-making under uncertainty

For MoD applications, several RL approaches have been presented in literature. To discuss them in a systematic way them, we refer to Powell (2019) who proposes a modelling framework for sequential decision-making problems and methods for solving them. The framework aims to unify all the methods and fields that address the general idea of sequential decision-making, such as stochastic optimization, reinforcement learning and optimal control. Powell (2019) suggests that sequential decision-making methods can fit into four classes.

In *Policy Function Approximations* (PFA), the policy is a parameterized function of the state. The goal is then to find the values of the parameters that maximize the objective function. Consider the task of controlling a fleet of electric vehicles. Deciding when to send them to charging stations can be parameterized with a threshold θ on the state of charge (The vehicle is sent to recharge if its state of charge is below θ). θ is the policy parameter that is learned.

In *Cost Function Approximations* (CFA), an embedded optimization problem is solved to find the best decision at each time (e.g., a linear programming algorithm that takes direct observations as input). However, the input of this optimization is parameterized, and good parameter values need to be learned in order to maximize the performance. For example, the expected occupancy of charging stations can be learned by time of day and used to parametrize an optimization algorithm to schedule slots at charging stations.

Third, Powell (2019) finds that *Value Function Approximation* (VFA) techniques constitute a large share of the literature on sequential decision-making. In this area, the goal is to build the value function and then derive the optimal policy. In contrast to CFA methods, what is learned here is the actual interest that situations represent (obtained rewards). These values are then used to orient the actions in order to reach good situations in which the reward is maximized. For instance, a VFA method can learn the expected reward of sending vehicles to charging stations as a function of the current state (time of the day and state of charge).

Finally, in *Direct Lookahead Approximations* (DLA), policies explicitly consider possible future situations, often by using a lookahead model that approximates the true behavior of the problem. Instead of approximating the value function, here what is approximated is the problem itself and the future states that will be encountered. The decision can then be taken by looking ahead using the approximated model. Such an approach can be used to learn a model for the evolution of the state of charge depending on the decisions, which can be taken such that they yield the best outcome according to the model.

2.4 . Reinforcement Learning methods for MoD

In this literature review, we focus on papers with an emphasis on the use of RL for the operation of MoD systems and use cases that attempt to reflect real-world settings. Our review is not exhaustive as we identified recent and relevant papers for our research. The literature was searched with the keywords *Mobility on Demand*, *taxi*, *ride-hailing*, *operation*, *fleet management*, *rebalancing*, *relocation*, *repositioning*, *dispatch*, *vehicle assignment* in combination with *reinforcement learning* and *learning*. To extend the set of covered papers, we performed a manual snowball search through the referenced articles.

We review the approaches under three axes: (i) the algorithmic core of the

approach, i.e., the RL method used, the task, the settings and the algorithmic fit in the framework presented in Subsection 2.3.2; (ii) the evaluation methodology, i.e., exploring how the algorithm is assessed, what parameters are studied (sensitivity analyses) and against what approaches it is compared; and (iii) the use case aspect, determining whether the study considers an urban vs. a rural setting, and whether it takes into account congestion and public transport.

We structure this section with respect to the MoD tasks that are performed by the approaches presented in the papers we reviewed: rebalancing only, dispatch only and joint rebalancing and dispatch.

2.4.1 . Rebalancing

In Fluri et al. (2019), a reinforcement learning technique is used to perform the rebalancing task. The problem is characterized by a set of disjoint zones $z \in Z$ and customers waiting in each zone C_z. The goal of the algorithm is to decide the target numbers of on-demand vehicles V_z in each zone z by using the negative sum of waiting customers $-\text{card}(C_z)$ as a reward.

Two variants were implemented, one using a classical tabular reinforcement learning where the parts of the network are all considered together on the same level in the value function. The second variant is a cascaded reinforcement learning, where the network is hierarchically divided using the Lloyd k-means algorithm, with $k = 2$ areas on each level (Lloyd, 1982). The training part is then performed from the upper level to the lower ones. A Q-Learning approach is used in both approaches, with an epsilon-greedy policy. An integer linear program is then used to compute the rebalancing decisions for each vehicle in order to match the demand in each zone while minimizing the overall travel distance for the vehicles. Consequently, Reinforcement Learning is used in this approach to build the input of an optimization problem. Therefore, we classify this method as a CFA.

The model is evaluated using publicly available data from the city of San Francisco with the agent based simulation framework AMoDeus (Ruch et al., 2018). The paper does not present sensitivity analyses, but rather focuses on the comparison of the two tested approaches with other control theoretical algorithms that exist in the literature (Pavone et al., 2012). The results on the tested cases show the advantage of using the cascaded reinforcement learning algorithm.

Wen et al. (2017) used a Deep Q-Learning approach for the rebalancing of on-demand vehicles. The optimal rebalancing problem definition in the paper also features a network divided into a set $z \in Z$ of disjoint areas where numbers of incoming requests are assumed to follow a Poisson process $A_z \sim \text{Poisson}(\lambda_z \Delta T)$, with λ_z being the arrival intensity for zone z, and ΔT the frequency of rebalancing. The decision variables consist of the matrix r_{ij} indicating the number of vehicles to be rebalanced from zone i to zone j, while $r_{i,} = \sum_{j \in Z} r_{ij}$ is the number of vehicles available in zone i. The objective is then to maximize $\sum_{j \in Z} b_j(v'_j) - \sum_{i,j \in Z} c_{i,j} r_{i,j}$. Here, v'_j is the expected number of assignable vehicles that will be available in zone

j at rebalancing time and $b_j(v'_j) = \sum_{k=0} \min(k, v'_j) \Pr(A_j = k)$ is the expected number of requests that can be served if there are v'_j vehicles in the same zone. $c_{i,j}$ indicates the cost of moving a vehicle from zone i to zone j, it is set to $cd_{i,j}$ if zone j is reachable from zone i within ΔT (with $d_{i,j}$ being the distance between the two zones); otherwise, it is set to a large constant $\bar{c}$.

Under the assumption of deterministic travel times and unchanged vehicle routes, and knowing the rebalancing decision, v'_j is defined as follows: the vehicles that should be at zone j at rebalancing time are considered with different weights according to the likelihood that a seat will be available depending on their current occupancy. The respective weights are 1, 0.4, 0.2, 0.1, 0 for loads 0, 1, 2, 3, 4 given 4-seated vehicles. These assumptions result in a Mixed Integer Nonlinear Programming (MINLP) problem that is solved approximately using a combination of incremental-optimal and branch-and-bound methods. This solving approach is not detailed further in the paper and is referred to as Heuristic Optimal Rebalancing (HOR).

In addition to comparing the deep Q-learning approach against HOR, the authors defined a Simple Anticipatory Rebalancing (SAR) strategy that rebalances a vehicle from zone i to a zone $j \in \overline{Z_i}$ where $\overline{Z_i}$ represents the current and neighboring zones of i ($i \in \overline{Z_i}$). The rebalancing zone is then sampled using probability $\Pr(\text{vehicle moves to j}) = \lambda_j / (\sum_{j' \in \overline{Z_i}} \lambda_{j'})$.

For the deep Q-learning algorithm, each vehicle considers its neighboring areas as the environment and their idle vehicles, in-service vehicles and predicted demands as constituting the state. The action is then chosen with an $\epsilon-$greedy policy, amongst one or none of the neighboring areas; if none is chosen, the vehicle does not rebalance. The reward design in this approach is twofold: if the vehicle is assigned, the setting is compared to the one without rebalancing and the saved wait time is taken as the reward (how this is computed is not detailed in the paper); and if the vehicle is not assigned, a constant penalty is applied. We note that this reward does not directly reflect the objective function that is presented in the optimal rebalancing problem formulation. Since the output of the algorithm is the straightforward location to which the vehicle needs to relocate to, we classify this approach as a VFA.

The algorithms are benchmarked on an abstract map with varying sizes under three different scenarios for the demand: uniformly distributed trips origins and destinations; two areas concentrating origins and two areas concentrating destinations; uniformly distributed origins with one fixed destination. The fleet sizes for the three different map sizes are 20, 125 and 810 respectively. For each setting, observed performances for HOR were better than for Deep Reinforcement Learning which were better than SOR. The three algorithms were all better than the absence of rebalancing. In terms of computation times, the increase was more drastic for HOR than for SOR and DQN. The rebalancing methods were then tested in a use case of shared MoD in Orpington, London using travel data spanning a 10 years

period. The results in this use case showed similar relative performance between the algorithms. The different tests were performed on the agent-based modeling platform detailed in Wen et al. (2018). Due to the assumptions of fixed travel times that were taken, we consider that this work does not take congestion into account.

In Yoshida et al. (2021), a decentralized deep Q-learning is used for the rebalancing task. The network is divided into a grid Z and each vehicle v considers a service area $S_v \subseteq Z$. The goal is then to adapt the service area for each vehicle. This is performed using a DQN for which the input (state) consists of the current demand d_z in each zone $z \in Z$; the demand forecast estimation d'_z in the next time period and the number of vehicles in the same area $n_v = \text{card}\{v' | v' \text{ located in zone z}, z \in S_v\}$ allowing the vehicles to take each other into account. Adapting the service area can be performed with six actions: enlarge, shrink, move up, down, left, right and stay (leaving the service area unchanged). The reward recieved by a vehice v is defined as $r_v = w_b b_v - w_d d_v - w_f f_v$ where b_v indicates the number of travelers assigned to v, d_v the distance between v and them and $f_v = 1$ if the last action selected by v is stay and $f_v = 0$ otherwise. The actions are selected following an ϵ-greedy policy with a linearly decaying ϵ. When idle, a vehicle is simply relocated to the center of its service area. We classify this approach as a VFA.

The algorithm proposed here is evaluated using the public benchmark dataset of taxi trips in Manhattan in a simulated grid environment. This study shows the advantage of using the proposed approach in comparison to the one presented in (Yoshida et al., 2020), another RL approach (Wen et al., 2017), and an approach that uses forecast data to move unassigned vehicles to areas with shortages using linear programming (Miao et al., 2016).

In contrast to previously mentioned works where the network is discretized into zones, Kim and Kim (2021) use a Deep Q-Learning approach in which the network is modelled as a directed graph $G = (V, E)$. V and E respectively denote the set of intersections and the set of roads linking them. At each time t and for each road $j \in E$, $v_{j,t} \in \mathbb{N}$ denotes the number of empty vehicles on road j, $n_{j,t} \in \mathbb{N}$ the number of waiting travelers and $p_{j,t}$ the speed on j. The deep Q-network takes as input the state of the road network that consists of a vector s_t containing the states $s_{j,t}$ $\forall j \in E$ with $s_{j,t} = (v_{j,t}, n_{j,t}, p_{j,t})$. After an action, each vehicle receives a reward value of 1 if it was assigned to a request and 0 otherwise. $Q(s_t, j)$ with $j \in E$ is the expected long term reward earned after moving vehicles to j from its neighboring roads. The update of the Q-values is not performed following the standard Q-learning equation, but rather an expected-SARSA (Equation 2.1). Using the Q-values, the policy selects stochastically the next road for each vehicle, such as the probability of moving from j to k is determined with the relative interest of road k among all roads adjacent to j. Therefore, we categorize this approach

as a VFA. The tests of this algorithm were performed in a custom-built simulator. However, the scalability of the method is not assessed as the size of the network considered in the use case has not been explicitly mentioned.

Gammelli et al. (2021) present a RL algorithm for rebalancing where the network is divided into zones that are then considered in a graph $G = (V, E)$ where each zone is linked to the nearest ones. Travelling from zone i to j $(i, j \in V)$ at time t has a cost $c^t_{i,j}$, a travelling time $\tau^t_{i,j}$, and a profit $p^t_{i,j}$ if the trip is transporting a passenger. The number of requests to travel from i to j at time t is noted $d^t_{i,j}$ and the number of successfully served ones among them is $x^t_{i,j}$. The RL rebalancing method is centralized and the state variable is composed of: (i) the adjacency matrix of the graph (ii) the numbers $m^t_i \in [0, M]$ $\forall i \in V$ of vehicles with M being the fleet size (iii) the projected availability of vehicles $m^{t'}_i$ $\forall i \in V, t' \in [t{+}1, ...t{+}T]$ with T being the planning horizon (iv) the current demand $d^t_{i,j} \forall (i, j) \in E$ and (v) the estimated future demand $d^{t'}_{i,j} \forall (i, j) \in E, t' \in [t{+}1, ...t{+}T]$. An action consists in choosing the desired distribution $a^t_{reb} = \{a^t_{reb,i}\}_{i \in V}$ of vehicles across the zones $(\sum_{i \in V} a^t_{reb,i} = 1)$. The reward r^t_{reb} that is considered in this approach is operator-centered, as it reflects the profit made by the service from served requests and the cost of moving vehicles $r^t_{reb} = \sum_{i,j \in V} x^t_{i,j}(p^t_{i,j} - c^t_{i,j}) - \sum_{(i,j) \in E} y^t_{i,j} c^t_{i,j}$. The rebalancing decision on the zone level, i.e. choosing the number $y^t_{i,j}$ $\forall (i, j) \in E$ of how many vehicles are relocated from zone i to zone j, is then performed using linear programming method minimizing the rebalancing cost. We then classify this approach as a CFA.

The policy is modelled with a graph neural network that allows to generalize beyond the order with which the zones are considered and make abstract conclusions that can be transferred between use cases. The policy is trained following the actor-critic approach and tested on two use-cases related to the cities of New York and Chengdu (China). The authors compare against equal distribution of vehicles among zones; the cascaded Q-learning approach presented in Fluri et al. (2019); and using feed-forward or a convolutional neural network instead of a graph neural network and the use of model predictive control methods in order to provide an upper bound of performance. In both cases, the presented approach is able to achieve the best reward performance excluding the model predictive control methods. To our knowledge, this paper is the only one in the literature that explores the potential of transferring a policy to either a completely new network or an extended one. The transferred policies achieve less reward than their counterparts learned on the new use case, but are still better than the other learning based approaches.

2.4.2 . Dispatch

In the work presented in Qin et al. (2021), a RL algorithm is used to aid the vehicle assignment task by selecting which vehicles and requests to consider for the decision-making and which to delay to the next decision epoch. Similarly to other works in the literature, this approach considers a network divided in a set

of zones $z \in Z$. The approach is centralized, with the state $s(t)$ at time t being modelled as $s(t) = \{\{N_p(z,t), N_d(z,t), \lambda_p(z,t), \lambda_d(z,t)\} | z \in Z\}$ where $N_p(z,t)$ and $N_d(z,t)$ are the number of requests and idle vehicles in zone z at time t. $\lambda_p(z,t)$ and $\lambda_d(z,t)$ are the estimated arrival rate of requests and idle vehicles to zone z from t onward. An action $a \in \{0,1\}^{|Z|}$ is a vector composed of binary values $a_z \in \{0,1\}$ for each $z \in Z$ where $a_z = 1$ if the requests and drivers of zone z will be considered in the next dispatch decision and $a_z = 0$ otherwise. The reward that is considered here is the induced waiting times for the users.

A policy gradient method is used in this work. The policy is a neural network that takes as input the state $s(t)$. The output layer of the neural network contains $2^{|Z|}$ elements in $[0,1]$ specifying a probability distribution of actions. This probability distribution is then used to sample the action. The weights of the neural network (and thus the policy) are adjusted following the Actor-Critic and Actor-Critic with Experience Replay methods. Selected requests and drivers are matched by solving a Bipartite Matching Problem. We consider this approach to be a CFA.

This approach was evaluated in a numerical experiment using a real world one week dataset from Shanghai Qiangsheng Taxi collected in March 2011. It has been compared against fixed matching delays in terms of resulting waiting times for the users and showed better performance. However, in this study, the number of considered zones did not exceed 6 whereas the size of the action space is exponential in the number of zones.

Enders et al. (2022) present a deep RL approach to address the dispatch of MoD vehicles. The problem considered in this work is characterized by a service area modelled as a graph $G = (V, E)$. Each edge $e \in E$ is associated with a weight vector $w_e = (w_e^d, w_e^t)$ where w_e^d denotes the distance of the edge and w_e^t the time necessary to traverse it. The approach is centralized and the system state at time t is defined as $s_t = (t, (r_t^i)_{i \in \{1,...,R_t\}}, (k_t^j)_{j \in \{1,...,K\}})$ with R_t being the number of travel requests at time t and K the fleet size. A request r is defined as $r = (\omega, o, d)$ with $\omega \in \mathbb{N}_0 \cup \{\emptyset\}$ is the current waiting time (it is set to $\emptyset$ at pickup), $o \in V$ and $d \in V \backslash \{o\}$ refer to the origin and destination of the request. A vehicle state $k_t^j = (v_t^j, \tau_t^j, r_t^{1,j}, r_t^{2,j})$ consists of a position $v_t^j \in V$, the time τ_t^j left to reach v_t^j (the vehicle can be travelling) and up to two assigned requests $r_t^{1,j}$ and $r_t^{2,j}$. Possible actions to perform at time t are tuples $(a_t^1, ..., a_t^{R_t})$ where $a_t^i \in \{0, ..., K\}$ such as $a_t^i = 0$ if request r_t^i is rejected and $a_t^i = j$ if it is assigned to the vehicle j. Only actions that satisfy $a_t^i = j \in \{1, ..., K\} \Rightarrow r_t^{2,j} = \emptyset \ \forall i \in \{1, ..., R_t\}$ (no request is assigned to a vehicle that already has two requests assigned to it) and $\sum_{i=1}^{R_t} \mathbb{1}(a_t^i = j) \leq 1 \ \forall j \in \{1, ..., K\}$ (at most one new request is assigned to each vehicle). The goal is to maximize the profit generated by the fleet: when picking up a passenger (related to a request $r_t^i = (\omega, o, d)$), the system generates a revenue $rev(r_t^i) > 0$ if $\omega < \omega_{max}$ and $rev(r_t^i) = 0$ otherwise; after moving from node v to node v' through edge $e = (v, v')$, a vehicle generates a cost equal to w_e^t. At each time t and after performing the action, the system receives as a reward

the sum of revenues generated by picked up requests from which are subtracted the costs generated by moving vehicles.

In contrast to most decentralized RL approaches, where each vehicle is considered as an RL agent, here each vehicle-request pair is considered as a separate RL agent that provides the probability of the request being assigned to the vehicle. This allows to build a weighted bipartite matching graph that is solved to compute the effective assignments. This method is consequently classified as a CFA. The learning is performed following an Actor-Critic method using neural networks for both actor and critic functions. The method is tested using experiments based on the New York Taxi data (Tlc, 2020) and compared against a greedy method and a model predictive control method. The proposed RL method shows better performance in general and more stability across the testing period.

2.4.3 . Rebalancing and dispatch

In Gueriau et al. (2020), a decentralized Q-learning algorithm for dispatch and rebalancing is studied. The network is divided to a set of zones $z \in Z$. At time t, the state of a vehicle v is $s_{v,t} = (l_{v,t}, d_v, d'_v)$ where $l_{v,t} \in \{\text{empty}, \text{partial}, \text{full}\}$ denotes the vehicle's load and d_v and d'_v are equal to 1 if there is a pending request in the same zone as v or one of the neighboring zones, respectively. The possible actions for each vehicle to decide from are: picking up a passenger by choosing the closest open request; rebalance by choosing from 4 strategies for selecting which neighboring zone to relocate to (the one with most requests, the one with the biggest gap between vehicle number and vehicle demand, the one with most historical requests, the one with the biggest historical gap between vehicle number and vehicle demand); or do nothing. This allows partially occupied vehicles to rebalance and serve other requests on their route. A vehicle receives a positive reward when picking passengers and no reward otherwise. The decisions can be carried out straightforwardly without further optimization. We consequently consider this approach as a VFA.

The proposed model was evaluated using the SUMO simulator (Lopez et al., 2018) on generated ride-requests for Manhattan using the New York City taxi data set over 50 consecutive Tuesdays from July 2015 to June 2016. The study focused on the morning rush hour. To take congestion into account, private vehicle trips have been generated using a uniform distribution. No analysis of the impact of the RL parameters is described in this study. The performance of the system and its impact were evaluated on three aspects: The system perspective, with the amounts of served and timed-out requests (requests expire within 10 minutes); The rider perspective: waiting time, detour time, total travel time; The vehicle perspective: total Vehicle Miles Traveled (VMT), empty VMT, engaged VMT, shared VMT and also vehicle occupancy. The algorithm has been compared to simple strategies involving centralized and decentralized vehicle rebalancing (each selecting the request with the longest waiting time) and rebalancing vehicles in the centroid of their predetermined "home" zone and ride-sharing.

Al-Abbasi et al. (2019) present DeepPool, a decentralized Deep Reinforcement Learning approach to learn good dispatch and ride-sharing behaviors for the vehicles on the zones of the network. The authors first present a detailed mathematical presentation of the problem where the network is divided into zones as is done in other approaches and where vehicles with a least one empty seat can decide to take in new requests. Each vehicle in the system has a state $S_{t,v}$ that consists of the vehicle's location, number of vacant seats, the passenger pickup time and the destination of each passenger. The global system state s_t variable comprises the vehicles' states, the estimations of the number of vehicles in each zone in T time steps $V_{t:t+T}$ and the estimation of the future demand in each zone $D_{t:t+T}$. Three neural networks are used in this framework: one to estimate travel times across the network which, when combined with the current vehicles' plans, allows estimating $V_{t:t+T}$; one to estimate future demands $D_{t:t+T}$; and lastly a deep Q-network that is fed with the state variables (including the outputs of the other networks) and estimates the Q-values of doing each of the possible actions: (i) where to dispatch each vehicle and, (ii) whether it takes new requests. The reward signal is a weighted sum combining: (1) the difference between demand and supply; (2) the total dispatch time of the vehicles; (3) the overhead caused to passenger travel times by ride-sharing; (4) the number of used vehicles.

The learning process uses an $\epsilon-$greedy method with a linearly decreasing ϵ and also a linearly decreasing learning rate α. We classify this approach as a VFA. The decisions operated by this algorithm are directly performed by the vehicles without going through another optimization process. In this study, the algorithm is evaluated on the public dataset of taxi trips in Manhattan, New York (Tlc, 2020). These data are used to build requests that are fed to a built-in simulator and handled by the service operated by DeepPool. The main metrics used to assess the performance of the system are the ones present in the objective function detailed above, plus waiting time and the rate of rejected requests. The estimation of travel times uses historical trip data, which means that traffic and congestion are considered in this study, but the impact of the MoD service on traffic is not.

In Haliem et al. (2022), the authors extend the research conducted in Al-Abbasi et al. (2019) to address the issue of catastrophic forgetting observed in neural networks (Kemker et al., 2018). This is performed by considering the environment as changing between a set of models (more specifically transition functions) and employing a change point detection algorithm to switch between models (KJ et al., 2022). Each model is associated with a deep Q-network that is trained with experiences observed in the given context. Consequently, this algorithm learns different policies as well as the appropriate time to use each policy. This approach is used to learn diurnal variations of the demand in a scenario based on a real public dataset of taxi trips in Manhattan, New York City. The results show the advantage of this approach over having only one model.

Mao et al. (2020) use an RL approach for combined vehicle assignment and rebalancing. In this work, the service area is divided into a set of zones Z. The state variable is a vector consisting of the time of the day t; the matrix $R_{i,j}$ with $i, j \in Z$ of number of requests with origin in i and destination in j and a vector V_i indicating the number of vehicles in each zone. The action learned in this approach is a matrix $M_{i,j}$ specifying the number of vehicles to relocate from i to j with $\sum_{j \in Z} M_{i,j} = V_i$. The vehicle assignment is directly dependent on the relocations, as requests can be served by vehicles with the same origin and destination if available. Otherwise, they will remain pending. Consequently, vehicles can either serve a request if they are assigned to it or simply relocate in the network. The reward signal used in this approach combines unassigned vehicles' relocation costs and users' waiting times.

To avoid the intractability of a large state-action space, a policy-based actor critic method is used, where the policy is a function of the state. In this approach, the policy and critic function are represented with feed forward dense neural networks that are adapted to enforce that the number of dispatched vehicles is not negative, and the constraint on the number of originating vehicles V_i is fulfilled. We therefore put this approach in the PFA category. Two settings were considered for the objective. In the first one, all users' waiting times are equally considered, while in the second one features "impatient" users with waiting times that exceed a determined threshold. Those are more weighted higher in comparison to "patient" users. The experiments were performed on a dataset of taxi trips in Manhattan. Compared against the REINFORCE algorithm with similar settings and an optimal solution method providing an upper bound, this approach showed to have better results.

Tang et al. (2019) and Tang et al. (2021) propose an approach that embeds both vehicle dispatching and rebalancing in one framework while combining both historical data and data acquired online. The algorithm is decentralized among the vehicles and the state s is modelled as $s = (l, u, v)$ where l refers to the vehicle's location, u denotes the real world time stamp and v is a vector comprising other dynamic information (e.g., current supply and demand) and static information (e.g., day of the week, holiday indicator). The vehicles learn to choose an action $a \in \{d, r\}$ (dispatch and rebalancing). The goal is to maximize the discounted sum of rewards $\sum_{j=1}^{T} \gamma^{j-1} r_j$ with r_j denoting the revenue generated by the trip assigned to the vehicle if $a_j = d$ and 0 otherwise. The dispatching is performed across all selected vehicles in order to maximize the total sum of expected rewards. The rebalancing destination is chosen stochastically for each selected vehicle based on weights derived from the expected value of each alternative.

A key interest of this approach lies in using one single value function for both the dispatch and rebalancing decisions for all the vehicles. The historical data are used to build an offline value function estimation, which is then taken into account alongside online observations to build the usable value function, which is unique

for all the vehicles and updated by all their observations. We consider this method to be a CFA, since the dispatch decision is performed by solving an optimization problem that is built using the value function.

The experiments have been performed in simulation environments built from DiDi's real-world ride-hailing data regarding three different cities over two weekdays and two weekend days. The *dispatching* performance of the proposed approach was compared to a baseline myopic method, a greedy method, the one presented in Tang et al. (2019), and the one that received the first prize from the same task in the KDD Cup 2020 RL track competition. Results show that the approach outperformed the others. On the *rebalancing* side, this approach was compared against the winning method in the rebalancing task of the same competition, a human expert policy extracted from historical data, and a deterministic and greedy version of the proposed approach where the location with the highest value is selected. The performance in this task was studied under different fleet sizes, and it is shown that the proposed method outperforms the others consistently in both versions. The approach was also tested on settings that attempt to reflect temporary and considerable changes on the supply and the demand, like the arrival of new vehicles or an event triggering many unexpected travel requests. The results show that the framework is able to adapt well to such situations.

This work is extended in Eshkevari et al. (2022) where results are presented regarding the implementation of the approach in a real-life ride-hailing service operated by DiDi. A/B testings were first conducted in five major cities where drivers switched between the RL algorithm and the baseline methods for periods of three hours and which demonstrated the relative performance of RL. The algorithm was then adopted to be used in one large unspecified city in China and has been in service since late December 2021.

In Liang et al. (2021), a Deep Reinforcement Learning approach is used to make combined decisions regarding dispatch, rebalancing and recharging. Here the network is divided into a set of hexagonal zones $z \in Z$. At each time t, the sets $R_{z,t}$ of open requests and $I_{z,t}$ of available vehicles in zone z are considered and for each $v \in I_{z,t}$, $soc_{v,t}$ denotes the state of charge of vehicle v at time t. The learning takes place on the zone level and the state of each zone z at time t is $s_{z,t} = (t, soc_{v_1,t}, soc_{v_2,t} \ldots)$ with $I_{z,t} = \{v_1, v_2, \ldots\}$. The zone level action consists of the joint actions of the vehicles $v \in I_{z,t}$. A vehicle's action is either to pick up a request, relocate to an adjacent zone, or recharge at a specific station: $a_{v,t} \in R_{z,t} \cup A_z \cup C_z$. $A_z \subset Z$ denotes the zones adjacent to z and C_z denotes the set of charging stations in z. The reward $r_{a_{v,t}}$ obtained by a vehicle v after performing an action $a_{v,t}$ is defined as follows. If $a_{v,t} \in R_{z,t}$ then $r_{a_{v,t}} = \sum_{\tau=t}^{t+\Delta t_{a_{v,t}}-1} \gamma^{\tau-1} \frac{P_{a_{v,t}}}{\Delta t_{a_{v,t}}}$ where $\Delta t_{a_{v,t}}$ denotes the duration of the selected request's trip and $P_{a_{v,t}}$ the revenue generated by it. γ is a discount factor parameter. If $a_{v,t} \in A_z$ then $r_{a_{v,t}} = 0$. And if $a_{v,t} \in C_z$ then $r_{a_{v,t}} = \sum_{\tau=t}^{t+\Delta t_{a_{v,t}}-1} \gamma^{\tau-1} P_{g,\tau} \Delta(soc_{v,\tau} - soc_{v,\tau-1})$ where $P_{z,\tau}$

refers to the price of electricity effective in zone z at time τ.

The joint decisions of the vehicles in each grid then feed into a binary linear programming algorithm that determines the optimal manner to implement the decisions (making this approach a CFA) while respecting constraints of at most one vehicle per request and one vehicle per charging station. The overall objective is to maximize the revenue of the service (and minimize cost). This algorithm is compared against solving the same binary linear programming problem but with actions determined by heuristics for dispatch, recharging and rebalancing. An interesting feature of this study is that the response of the algorithm to various electricity pricing schemes was studied (fixed prices, prices depending on time and/or location).

Castagna et al. use a decentralized RL algorithm for rebalancing and dispatch of a ride-shared MoD system. The algorithm is decentralized and a vehicle state $s_v = (l_v, d_v, e_v, p_v)$ consists of its location l (latitude and longitude), its next destination d (the closest destination for on-board passengers) and the number of empty seats e. Additionally, $p_v = (s_r^1, s_r^2, s_r^3)$ is the vehicle's perception vector that includes the information related to the 3 closest requests. $s_r^i = (l_r^i, d_r^i, n_r^i)$ with l_r^i, d_r^i and n_r^i, respectively, referring to the request's pickup location, its drop-off location and number of passengers. The action space consists of five alternatives: rebalance, picking up one of three possible requests, and dropping off the passenger(s) with the closest destination. The reward is set to favor ride-sharing by giving a higher reward for picking-up a passenger when the vehicle is already occupied than when it is not. Impossible actions (drop-off when the vehicle is empty or pick-up when the perception vector is empty) are penalized. Like other approaches, the rebalancing procedure considers a set of discrete zones dividing the network. However, here, the set of zones is not fixed beforehand and is computed at each rebalancing decision using an Expectation-Maximization technique based on pending travel requests. Consequently, open requests are dynamically clustered into spatial zones. These clusters are then considered for rebalancing. Each vehicle samples a target zone for rebalancing with a probability equal to the ratio of requests contained in the zone.

In this approach, a good policy is learned by using a proximal policy optimization method (Schulman et al., 2017), making it a PFA method. The approach is tested with the New York Taxi dataset under various configurations characterized by enabling or disabling ride-sharing and by the rebalancing method that is used (no rebalancing, rebalancing with fixed zones, rebalancing with dynamically computed zones). The results show the advantage of using both ride-sharing and dynamic zones computation for enhancing number of served requests and lowering users' wait times.

Reference	Algorithmic aspects					Use case				Evaluation	
	Rebalancing	Dispatch	Ridecharing	Recharging	Policy type	Congestion	PT	Environment	Study area	Simulation	Sensitivity analyses
Fluri et al. (2019)	✓				CFA			urban	San Francisco	MATSim (AMoDeus)	AP
Wen et al. (2017)	✓		✓		VFA			urban	Orpington, London	grid environment	SP
Yoshida et al. (2021)	✓		✓		VFA			urban	Manhattan	grid environment	AP+SP
Kim and Kim (2021)	✓				VFA	✓		urban	Seoul	built-in simulator	
Gammelli et al. (2021)	✓				CFA			urban	New York and Chengdu	built-in simulator	MP
Qin et al. (2021)		✓			CFA			urban	Shanghai	built-in simulator	AP
Enders et al. (2022)		✓	✓		CFA			urban	New York	grid environment	AP+SP
Gueriau et al. (2020)	✓	✓	✓		VFA	✓		urban	Manhattan	Sumo	SP
Al-Abbasi et al. (2019)	✓	✓	✓		VFA			urban	Manhattan	built-in simulator	SP
Haliem et al. (2022)	✓	✓	✓		VFA			urban	Manhattan	built-in simulator	AP
Mao et al. (2020)	✓	✓	✓		PFA	✓		urban	Manhattan	grid environment	
Tang et al. (2021)	✓	✓			CFA	✓		urban	undefined cities in China	DiDi simulation platform	MP
Liang et al. (2021)	✓	✓		✓	CFA	✓		urban	Haikou, China	built-in simulator	AP+SP
Castagna et al.	✓	✓	✓		PFA	✓		urban	Manhattan	built-in simulator	

Table 2.1: Overview of the detailed papers and the features of presented approaches. Policy types: see section 2.3.2. Sensitivity analyses: Approach Parameters (AP), Scenario Parameters (SP), Multiple Scenarios (MP)

2.4.4 . Analysis

In this section, we present our analysis of the literature with regard to our points of interest: The algorithmic basis; the use case aspects; and the methodology that is followed to study the algorithm's performance. In total, 14 papers have been analyzed based on the previous section. Their key characteristics are summarized in Table 2.1.

Algorithmic aspects

We first consider the MoD operation sub-tasks that are addressed in the literature. Empty vehicle rebalancing is the operation task that is studied the most. It is studied exclusively in five of the detailed papers and combined with vehicle assignment in seven other papers. The dispatching task is studied in isolation only in Enders et al. (2022) and Qin et al. (2021).

Regarding service characteristics, ride-sharing is taken into account in eight papers, whereas the necessity of recharging vehicles is considered only in Liang et al. (2021).

Going further into detail, we notice that even for approaches that address the same MoD operation sub-task with similar features, the lower level definitions of the problem can differ and service assumptions can vary strongly. For instance, Wen et al. (2017) and Yoshida et al. (2021) both study the relocation task with ride-sharing. However, the former considers the exact areas that the vehicles will be relocated to while the latter operates on the vehicle's service area that is shrunk, enlarged or moved. Moreover, even when the addressed tasks are the same, the rewards that are maximized can vary widely. For instance, Fluri et al. (2019) uses a reward that reflects the number of waiting users (encouraging the service to serve all requests) while Gammelli et al. (2021) considers operator profit (offsetting revenue and cost) as a reward.

Regarding the framework for sequential decision-making presented in Subsection 2.3.2, VFA and CFA approaches are used in six of the detailed papers, respectively. PFA techniques are underrepresented, with only two papers (Mao et al., 2020; Castagna et al.). However, the analyzed approaches show the potential of such methods and, hence, point towards promising future research. In terms of representing the value function, only two papers use a classical tabular representation (Fluri et al., 2019; Gueriau et al., 2020). All other papers make use of (Deep) Neural Network approximators.

Use cases

All examined papers consider an urban environment with highly dense cities such as San Francisco (Fluri et al., 2019), London (Wen et al., 2017) and Manhattan (six out of fourteen). The potential of MoD in rural setting has been demonstrated in the literature (Sieber et al., 2020) and its operation on such use cases using

RL techniques is worth investigating in future research. A challenge thereby is to obtain relevant benchmarking data, which could be obtained from individual mobility traces or synthetic data (Hörl and Balac, 2021b) based on dynamic demand simulations that integrate traveler behavior (Hörl et al., 2021).

Six studies consider the presence of private vehicles and congestion. The presence of public transports is never considered throughout the analyzed literature. Consistently integrating the surrounding transport system, hence, still poses a challenge in RL-based approaches and constitutes a pathway for future research. Furthermore, apart from differentiating between delayed and non-delayed passengers (Mao et al., 2020), only homogeneous vehicle fleets and passengers are considered in the analyzed papers. However, heterogeneous dispatching, which is relatively rare (Molenbruch et al., 2017), might benefit from the flexible problem formulations in RL.

Evaluation methodology

To simulate vehicle movements, some papers use custom-built simulators.Qin et al. (2021) and Al-Abbasi et al. (2019) make use of a detailed and realistic road network representation, whereas Yoshida et al. (2021) supposes a simple grid network. Two of the analyzed papers use open-source agent-based simulators (MATSim and SUMO) which arguably increase the reproducibility of the presented approaches and allow the integration and coupling of the on-demand systems with the rest of the transport system.

Most of the sensitivity analyses are performed by varying the parameters of the presented algorithms (six out of fourteen). These analyses regard the parameters that are at the core of the algorithms, whereas environmental parameters, in particular regarding the discretization of the study areas and networks, are not considered. However, they could give useful insights on the performance of the approach and its scalability. Regarding use case parameters, most of the analyses concern the MoD fleet size (five of fourteen).

While some of the presented approaches use abstract problem formations that, hypothetically, can be transferred from one use case to another, only one example (Gammelli et al., 2021) is presented where transferability is demonstrated and quantified. Given that transferability is frequently cited as a potential major advantage of RL-based methods, it is surprising to not see analyses of the excess learning effort (transfer costs) from one case to another more often. We, hence, encourage further research in that direction.

In terms of benchmarking the performance of the approaches, we observe that new approaches are still rarely tested against other RL-based methods and that authors fall back to comparison with classic methods. This is partly a result of lacking openness in the approaches as only few of them are easily reproducible by other researchers for comparison. Among the papers reviewed in this work, only Gammelli et al. (2021) provides access to the code allowing to reproduce the results

through an open repository. The rarity of comparisons between RL approaches can also be explained by a lack of comprehensive benchmarks in the literature and the early stage of RL-based fleet control methods, but should be considered in the future. The only structured benchmark of RL based algorithms for MoD operation that we have encountered in our review was the KDD RL cup competition, to which Tang et al. (2021) has been submitted.

2.5 . Conclusion

In this chapter were introduced the key concept that constitute the background for this thesis work. First the concept of a MoD system and the decisions involved in its design on the operational level were introduced in Section 2.1. These will be of key importance in chapters 4 and 5 that address the topic of MoD. Section 2.2 presented the interest of agent-based modeling and simulation approaches for mobility studies and detailed the platform used in this work, MATSim. Finally, Section 2.3 introduced the basis of RL and sequential decision making and Section 2.4 presented a detailed literature review on the usage of RL-based algorithms for the operation of MoD systems.

3 - Developing the Île-de-France/Paris-Saclay use case

In this chapter, the methodology that is followed to build our main simulation use case is presented. The chapter opens in section 3.1 with an introduction on our study area, Île-de-France and Paris-Saclay, the mobility challenges that are faced and the future public transport projects that are planned by 2030. The methodology followed in this work is then presented under three aspects: (i) the generation of the synthetic population of the study area in Subsection 3.2.1, (ii) the evaluation methodology in Subsection 3.2.2 and (iii) the scenario building in Subsection 3.2.3. Simulation results are presented in 3.3 and discussed in 3.4. This chapter draws from a conference paper presented at the 102^{nd} annual meeting of the Transportation Research Board (TRB 2023) Chouaki et al. (2023).

3.1 . Île-de-France, Paris-Saclay and future public transports

In our work, we focus essentially on the Île-de-France region, one of the twelve regions of metropolitan France, and located in the north-central part. Encompassing the capital city of Paris and its suburbs, this region is the most populous of the country, with over 12 million inhabitants for an area of 12012 square kilometers.

On the public transportation side, the Île-de-France region is home to one of the most complex and extensive transportation systems in the world, with a dense network of public transport connecting the capital city, the suburbs and the outer areas of the region.

The Paris Métro, which first opened in 1900, is the fifth oldest subway network in the world. Today it comprises 14 lines that span over 220 kilometers and 303 stations with an average of 550 meters between two consecutive stops. The five RER lines (Réseau Express Regional for express regional network, a hybrid subway-train system) traverse the region through Paris and offer means to travel from and to the suburbs. These lines cover 600 kilometers and serve 249 stations. The other eight train lines, depart from Paris and link the outer areas of the region, covering 903 kilometers. The rail-based offer also includes 11 tramway lines that cover 126 kilometers and 235 stations. The extensive bus offer comprises around 1500 lines throughout the territory[1].

However, the transportation system in Île-de-France also faces several challenges, such as overcrowding, aging infrastructure, and congestion. Moreover, the public transport network does not cover all areas of the region, leaving some suburbs and rural areas underserved or isolated.

To address these issues, several major transportation projects are underway

[1]https://www.iledefrance-mobilites.fr/le-reseau/services-de-mobilite

in Île-de-France, such as the Grand Paris Express, which aims to create a new rapid transit system connecting the suburbs of Paris and reducing travel time and congestion. The project includes the construction of four new metro lines (15 to 18) and the extension of two existing ones (11 and 14), covering a distance of 200 kilometers and serving 68 stations[2]. Other tramway projects are also planned throughout the area and intend to further reinforce the public transport offer.

In this thesis, We focus our simulations and analysis on a subset of the Île-de-France region which is of particular interest to us. This consists in the urban community of Paris-Saclay (Communauté Paris-Saclay - CPS) which is a gathering of 27 municipalities at the south of Paris[3]. Subject to major ongoing developments, the CPS area is growing into a major scientific and technological hub with already 65000 students and 15000 researchers across 280 research labs. The CPS area is already connected to the rest of Île-de-France through two RER lines, RER B and RER C, across 18 stations. The high-speed train station in Massy-Palaiseau provides direct connection to various other regions in France. Among the future rail-based lines planned for Île-de-France, the CPS area is concerned by two, line Métro 18 from the Grand Paris Express and the line Tram 12. Figure 3.1 shows the new future lines taken into account in our work as well as the geographical location of the CPS area relatively to Île-de-France.

The focus on the CPS area is motivated by three reasons. First is that reducing the geographical scope allows to go deeper in the analyses of the results. The changes in number of travelers in public transport can be assessed on the level of lines including buses. Second is the presence of the local authorities of the Paris-Saclay area among the partners of the Anthropolis Chair which hosts this thesis. The ability to discuss with the local authorities and validate hypothesis and retrieve insights has proven to be greatly valuable for this work. Finally, for the work on MoD presented in chapters 4 and 5, various settings of the MoD system are tested (fleet size and algorithm parameters). This results in a large number of simulations making the overall time required for the work highly dependent on the execution time of a single simulation thus motivating reducing the study area.

3.2 . Methodology

3.2.1 . Synthetic population generation

As mentioned in 2.2, we use the MATSim simulation framework. The tool allows assessing the impact of different mobility demand and offer configurations on users' choices and service quality. An important part of an agent-based mobility simulation methodology is the synthetic population that is given as input to the simulation. Since it considers individual agents in a microscopic manner, it relies on

[2]https://www.societedugrandparis.fr

[3]http://www.paris-saclay.com/l-agglo/grands-projets/cluster-paris-saclay-270.html

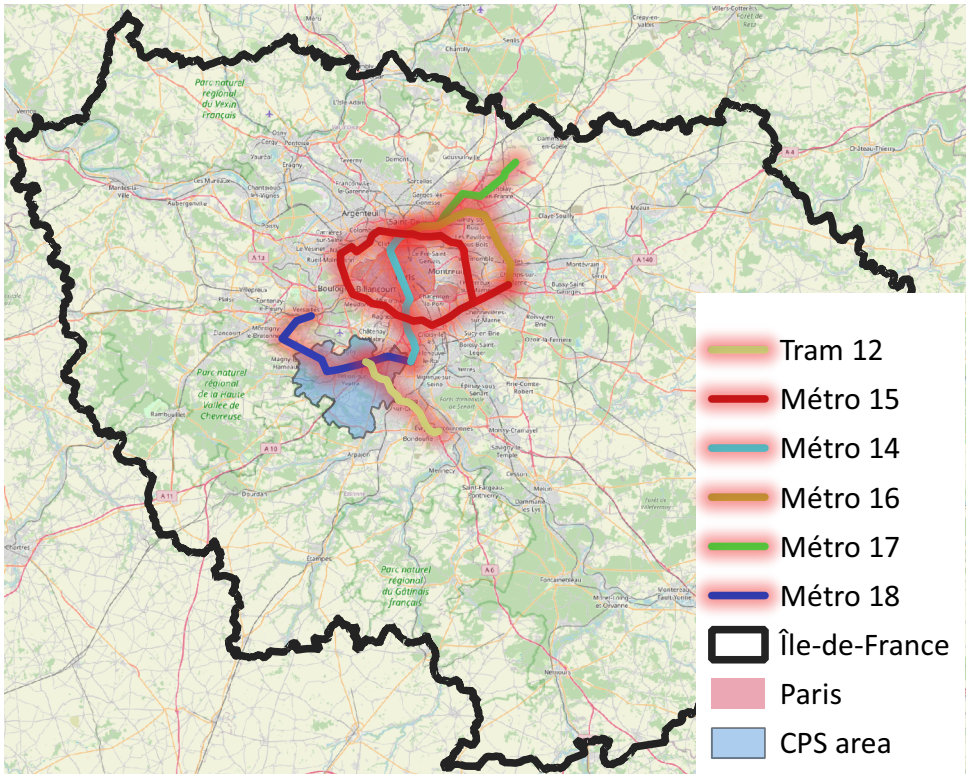

Figure 3.1: Overview of the future rail-based public transport lines included in our study. Background map provided by OpenStreetMap

fine-grained data related to individuals. In MATSim, the required data consist of a set of agents with a list of activities, for which the locations in space and in time are given (see 2.2.2). Various methodologies have been employed in the literature to generate synthetic populations. They differ according to the geographical use case and the data that are available. Consequently, the openness of the data that are used in a particular method plays a major role in its reproducibility.

For our study, we rely on a well established methodology for the generation of a synthetic population for the Île-de-France. First introduced in Hörl and Balac (2021b), the tool is openly available online[4] and relies on various socio-economical data. Table 3.1 summarizes the data that were used to generate the base synthetic population for Île-de-France. The openness of the data and the approach make the simulations fully reproducible and extensible by the research community. The resulting population consists of more than 5 million households with 11 million persons and 46 million individual daily activities.

MATSim performs simulations in iterations, changing users' choices (typically used transportation modes) between iterations. Built-in functionality allows simulating various transportation modes such as cars, regular public transport, and also shared MoD. Each transportation mode in MATSim is associated to a routing procedure. The modes that are considered during a simulation are available alternatives for the agents to chose from for their trips at the beginning of each iteration. the *eqasim* extension is applied to simulate mode choices Hörl and Balac (2021a). This extension makes use of finely configurable discrete choice models. More specifically, we use discrete mode choice model for the individual choices of transportation modes to use for each trip. This choice relies on the computation of the expected utility of the considered alternative, in our case, a chain of modes of a home-based tour. We use the mode choice parameters and utility computation methods documented by Hörl et al. (2019).

In order to do so, we apply a process to cut the simulated Île-de-France scenarios to keep only the relevant area (Hörl, 2020). From the population side, we keep those agents who perform at least an activity inside the area or which cross it at any point during the day. Agents enter and exit the cut simulation at the exact times and location where they enter it in a simulation of the full region. This allows to not only consider the inhabitants of the particular area but also, for instance, people who live elsewhere in Île-de-France but work in the study area.

The cutting process has an impact on the mode choice in the resulting simulations making some mode changes inconsistent to the original simulation. We will detail the allowed mode transitions that were allowed in the feeder simulation below.

[4]https://github.com/eqasim-org/ile-de-france

Title	Format	Source/Provider	Used version
Individus localisés au canton-ou-ville - Zone A	dbase	INSEE	2015
Mobilités professionnelles des individus : déplacements commune de résidence / commune de travail	dbase	INSEE	2015
Mobilités scolaires des individus: déplacements commune de résidence / commune de scolarisation	dbase	INSEE	2015
Population en 2015 - IRIS - France hors Mayotte	dbase	INSEE	2015
Base niveau communes	xls	INSEE	2015
Base niveau administratif	xls	INSEE	2015
Équipements géolocalisés (commerce, services, santé...)	csv	INSEE	2020
Enquête nationale transports et déplacements (ENTD)	csv	Ministère de la transition écologique et de la cohésion des territoires	2008
Contours IRIS		IGN/INSEE	2017
Découpage infracommunal	xls	INSEE	2017
Base Sirene des entreprises et de leurs établissements	csv		2021
La modélisation 2D et 3D du territoire et de ses infrastructures sur l'ensemble du territoire français			2021
Cartographie OpenStreetMap pour la région Île-de-France		OpenStreetMap	2021
Horaires prévues sur les lignes de transport en commun d'Île-de-France	GTFS	IDFM	2022

Table 3.1: Summary of data that were used for the generation of the synthetic population of Île-de-France that was used in this work

3.2.2 . Evaluation

In the evaluation process of our simulation results, we adopt a transport planning oriented approach. The emphasis is put on the usage (number of trips) of the future lines as well as the impact on the usage of existing rail-based lines. We also consider the impact on the usage of transit stations (number of travelers passing by). The focus is on an analysis of CPS area, taking into account only the stops that are located in the area and the public transport lines go through it and counting only the trips with either the origin or the destination station inside the CPS. However, since the simulations are first performed on the Île-de-France scale, the overall impact of the new lines are well taken into account. Restricting the area for our analyses allows us to also consider the impacts on the ridership of bus lines.

Regarding the changes in agent behaviors, the Discrete Mode Choice (DMC) model implemented in MATSim was used (described in 2.2.2). The model is parameterized using reference values calibrated on the observed mode shares in Île-de-France. At each iteration, 5% of the agents consider replanning their trips. Alternatives for each home-based tour are considered. Rather than considering all possible mode combination for the tour, a filtering is performed to avoid illogical options. For example, choosing car then public transports for the two trips of a home→leisure→home tour would mean that the agent's car would be left at the leisure location. For each considered tour alternative, the trips are routed using the mode-specific routing procedures. This allows to estimate trip quality indicators such as travel time, wait time, monetary cost and number of transfers. These indicators are used to compute the trip utilities which are then summed to retrieve the utility of the overall tour alternative. A Multinomial Logit selection is performed on the basis of the computed utilities in order to chose the alternative that will replace the previous trip modes.

3.2.3 . Scenarios

The synthetic Île-de-France population generated as presented in the previous subsection can directly be simulated. This default simulation scenario generated through this process is our baseline Île-de-France scenario, it comprises the currently existing public transport lines with their schedules generated using public GTFS data.

Upon the baseline Île-de-France scenario, we build our second scenario (2030 mobility offer) by adding the subway lines that are planned as part of the Grand Paris Express. These are the new lines 15 to 18 and the extension of the existing line 14. We gathered openly available datasets regarding the positions of stations for these lines[5], the travel times between the stations[6], and the frequency of sub-

[5]https://www.data.gouv.fr/fr/datasets/point-de-localisation-des-gares-du-grand-paris-express/

[6]https://www.data.gouv.fr/fr/datasets/temps-de-parcours-intergares-

ways[7]. We then used more recent information regarding the lines[8] to build the stop sequences. We also add an upcoming tram line (T12) to this scenario by using the stations' locations[9] and taking as a hypothesis an average speed of 40km/h. This results in our GPE+T12 scenario on the Île-de-France level.

Both scenarios are simulated on the Île-de-France level for 80 iterations. The resulting populations of both simulations are then cut around the CPS area. More precisely, the road network is cut along the borders of the CPS area, generating entry and exit points. Public transport lines and routes are processed the same way. Regarding the agents, they are filtered to keep only those that are inside the CPS area at some point during the simulation. The plan of an agent that has been selected for CPS scenario is also processed to only model explicitly the parts inside CPS. The part of the plans that lay outside of the study area are replaced by fixed teleportation between fake *outside* activities. For example, an agent in a Île-de-France level simulation that has the activity sequence home→work→home with the home and work activities taking place respectively outside and inside of CPS and a plan specifying the car mode for each of the two trips, the corresponding agent in the CPS level simulation will have a outside→work→outside activity sequence with car trips and where the outside activities are located at the border of CPS from which the initial car trip enters or leaves the area. Figure 3.3 summarizes this workflow of scenario-building and simulations.

The cutting procedure described above has implications on the mode choice model in the CPS level scenarios. Tours starting or ending with an outside activity cannot be replanned while ensuring the overall logic of the original trip in the Île-de-France level still holds. Taking the example of the outside→work→outside with car trips mentioned above, replacing the initial car-based alternative with another fully based on public transports (because the latter is evaluated much better than the former) would mean one of two things: either the agent's car is parked at the borders of CPS or the totality of the car trip on the Île-de-France level is replaced by public transports. In both cases, the impact of these decisions on the utilities on the Île-de-France level is not taken into account and the relative advantage of the new alternative is no longer certain.

3.3 . Results

With access to a computer cluster with 1TB of random-access memory (RAM) and 4 CPUs (Intel Xeon Gold 6230 20C @ 2.1GHz) with a total of 80 cores, each of the two simulations on the the Île-de-France level takes 21 days. Simulations

previsionnels/

[7]https://www.data.gouv.fr/fr/datasets/frequences-previsionnelles-des-trains-aux-heures-de-pointe-du-matin/

[8]https://www.societedugrandparis.fr

[9]https://tram12-express.iledefrance-mobilites.fr/decouvrir-le-projet/

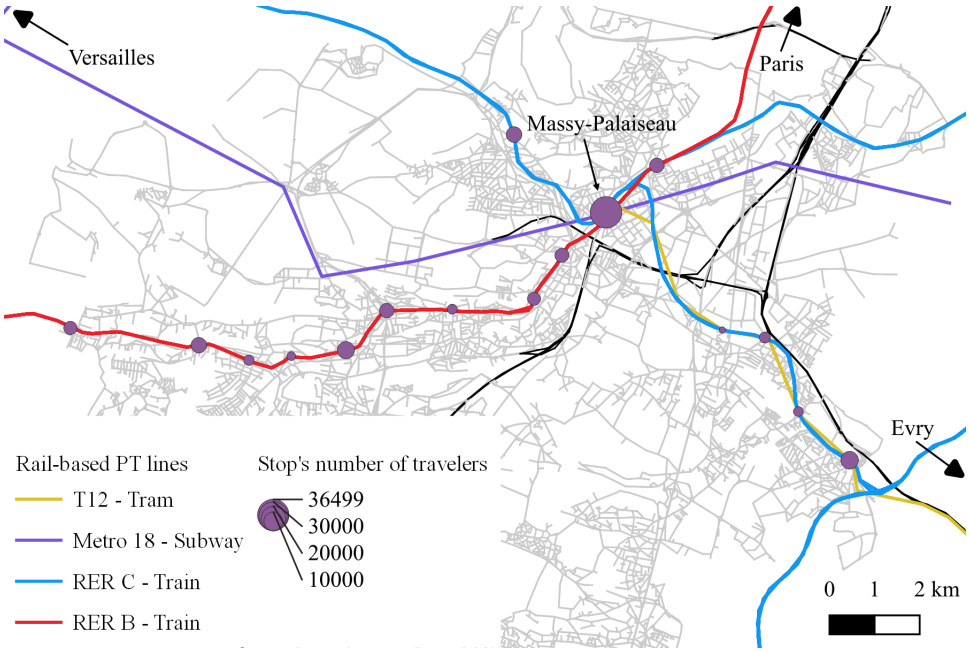

Figure 3.2: Existing and future rail-based transit lines that are considered in our simulations - Focus on the CPS area

on the CPS level take less than one day.

In this section, we detail and discuss the simulation results of the two presented scenarios on the CPS level.

3.3.1 . Baseline

Table 3.2 shows the distribution of activities preceding and following the agents trips. The *outside* activity refers to times of the day when the agent is outside of the CPS area. The results of the Baseline simulations highlight the strong connection of the CPS territory with the rest of Île-de-France, 47% and 30% of the trips taking place on the CPS territory respectively have both ends or only one end located outside the area as shown in Figure 3.5. However, this distribution is not uniform throughout trip modes as shown in Figure 3.4. 74% of the trips are performed by car, 60% by the drivers themselves and 14% by passengers. Public transports hold a 10% share of the overall trips.

Currently, most of the trips (84%) are performed in buses as the two existing rail-based lines do not cover the whole area (Figure 3.6). Note that a trip with mode public transports does not necessarily correspond to only one trip with one of the sub-modes as it can comprise multiple chained sub-trips. Table 3.3 shows the number of trips performed using the different transit lines inside the CPS in the Baseline scenario as well as the gains or losses that are observed for these lines in the GPE+T12 simulation. Regarding the currently existing rail-based lines, RER B is witnessing 74% more trips than RER C. The former being the most direct link

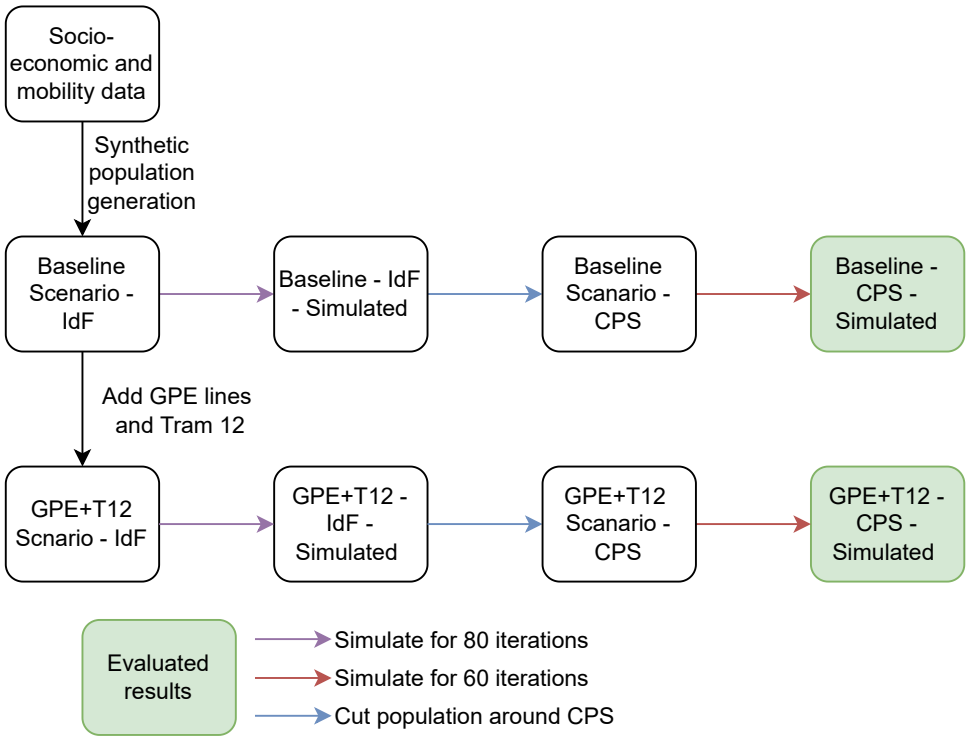

Figure 3.3: Summary of our workflow of scenario-building and simulation

following activity preceding activity	education	home	leisure	other	outside	shop	work	total
education	0.08%	1.97%	0.21%	0.05%	0.58%	0.04%	0.00%	2.93%
home	2.11%	0.14%	1.79%	1.91%	5.93%	2.02%	1.62%	15.53%
leisure	0.08%	1.93%	0.27%	0.17%	2.19%	0.13%	0.25%	5.02%
other	0.03%	1.77%	0.19%	0.26%	1.85%	0.21%	0.16%	4.48%
outside	0.61%	5.92%	2.22%	1.86%	47.02%	1.56%	2.94%	62.12%
shop	0.02%	2.31%	0.10%	0.08%	1.55%	0.16%	0.10%	4.30%
work	0.00%	1.47%	0.31%	0.17%	2.86%	0.20%	0.61%	5.63%
total	2.94%	15.51%	5.08%	4.49%	61.99%	4.32%	5.67%	100.00%

Table 3.2: Distribution of trip purposes, i.e. preceding and following activities, in the Baseline CPS simulation

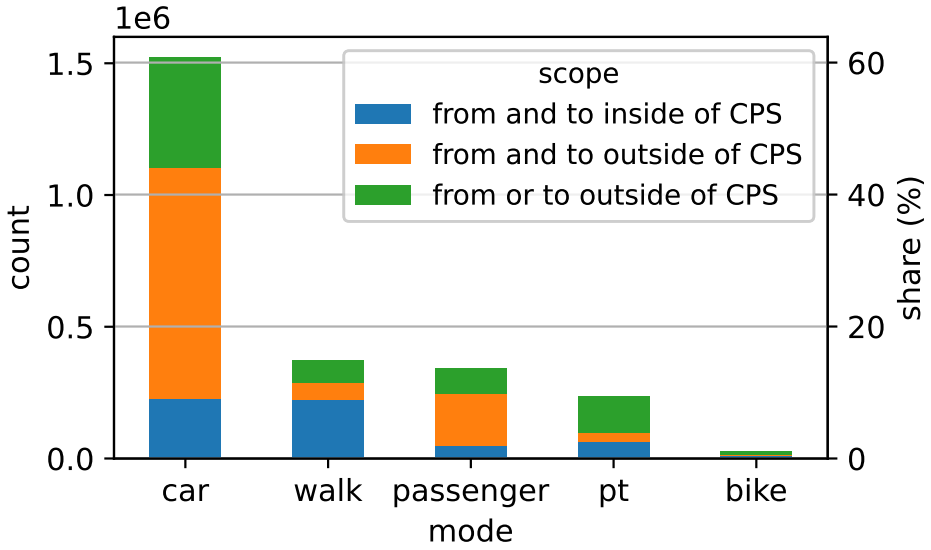

Figure 3.4: Distribution of trip modes and scopes relatively to the CPS area in the baseline simulation

to Paris. In our simulations, a total of 59 bus lines is simulated. In table 3.3, we focus on the lines with at least 1000 passengers per day and a gain or loss higher than 15% is observed in the GPE+T12 simulation. Currently, the three bus lines with most number of trips are 107, 297 and 91-06 with 8.6%, 5.5% and 4.8% of bus trip segments respectively (although the latter one is not shown in the table). All of these lines are linked to at least one rail station.

As shown Figure in 3.2, the Massy-Palaiseau station is the most crowded one in the CPS area with more than 36000 daily visitors while all of the other stations do not exceed 10000 visitors per day. This station being the meeting point between RER B and RER C lines.

Line	Baseline Ridership	Δ (abs) GPE+T12	Δ (%) GPE+T12
108	1483	-663	-44.71
107	23159	-8145	-35.17
197	1032	-283	-27.42
199	8066	-1872	-23.21
3	3942	-825	-20.93
91-02	1691	-308	-18.21
91-05	5806	-890	-15.33
399	11961	-1325	-11.08
297	14896	-1548	-10.39
299	1861	-192	-10.32
1	10261	-949	-9.25
18	1171	-105	-8.97
RER C	18611	-1629	-8.75
2	11725	-968	-8.26
17	10247	-791	-7.72
23	6336	-457	-7.21
DM153	11965	-664	-5.55
DM11E	2550	-103	-4.04
DM11 A	5865	-128	-2.18
22	3620	-73	-2.02
DM12	6002	-120	-2.00
DM10 A	2930	-40	-1.37
DM11 C	3631	-49	-1.35
DM13	1404	-13	-0.93
RER B	32349	-148	-0.46
39-07	3394	71	2.09
11	6492	379	5.84
91-11	4336	886	20.43
119	9908	2981	30.09
Tram 12	-	18478	-
Metro 18	-	15062	-

Table 3.3: Observed riderships of public transport lines in the CPS area in the Baseline and GPE+T12 scenarios

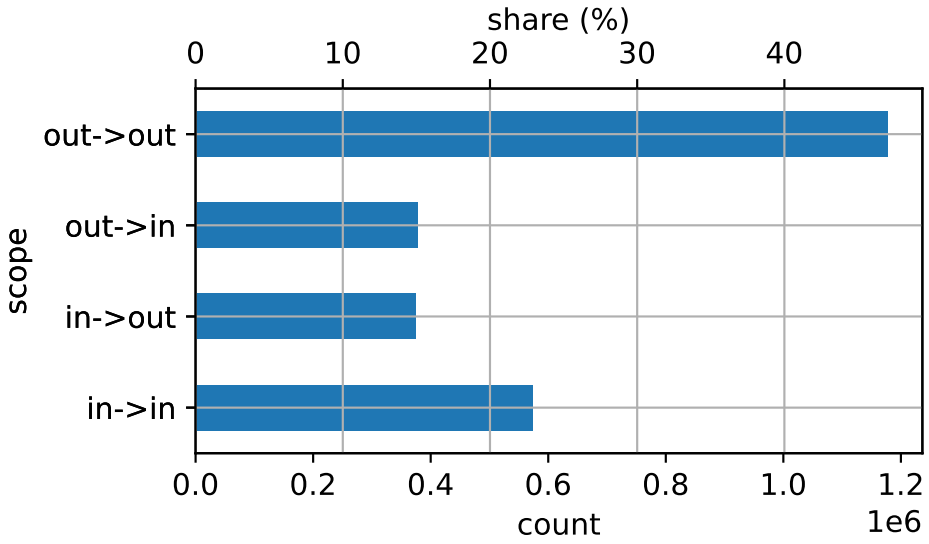

Figure 3.5: Distribution of trip scopes relatively to the CPS area in the baseline simulation. **in** for locations inside the CPS area and **out** for locations outside the CPS area.

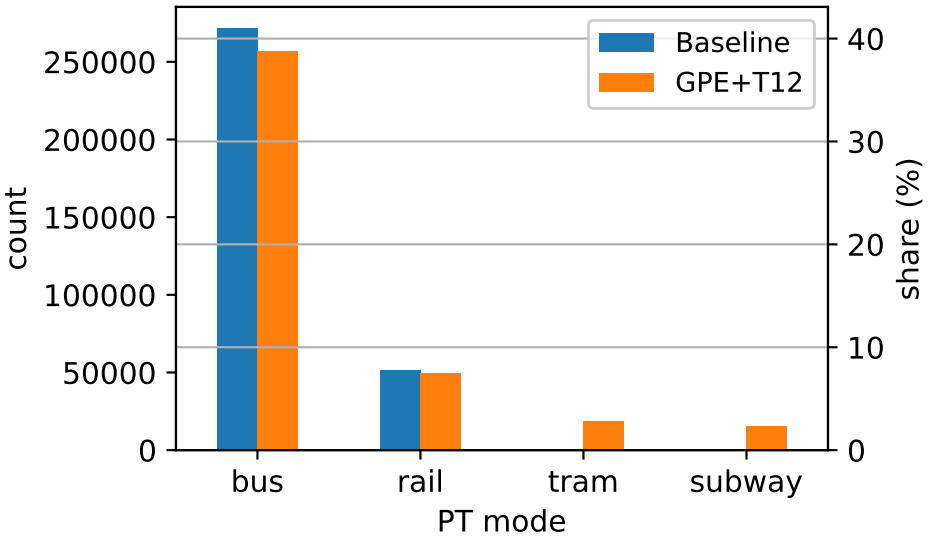

Figure 3.6: Trip counts per public transport sub-mode

3.3.2 . GPE+T12

The GPE+T12 simulation results show that the addition of the two lines Tram 12 and Metro 18 will not reduce car use on the CPS area but rather introduce changes in the use of public transports. These new lines would not impact the ridership of the existing lines equally, whereas trips performed with RER C decrease by 8%, the number of trips related to the RER B do not substantially vary. However, most of the trips on the two new lines (18000 and 15000 trips respectively), come from bus lines that are rendered less attractive. This is especially the case for lines 107, 199 and 299 (-35.17, -23.21% and -10.39% respectively). Nonetheless, even if the general numbers of bus trips decreases, a few lines benefit from their proximity with stations of the new lines. Lines 10, 119, 6 respectively see their number of trips increase by 38.69%, 30.09% and 29.63%. A special case is the line 7 with an increase of 622.76% as it went from 496 to 3089 passengers per day.

3.4 . Discussion

The simulation results allow us to gain insight on the future impacts of upcoming transit lines on the CPS. In general, the introduction of Tram 12 and Metro 18 reinforces the rail-based mobility offer and provides more attractive route alternative than existing bus lines. In this section, we discuss our methodology and the assumptions that were made while building the simulation scenarios.

The future public transit lines that are taken into account in this work are planned to begin operating progressively between 2024 and 2030. However, the synthetic population generation is performed using existing data reflecting a relatively recent population. For a more comprehensive analysis, a prospective synthetic population for 2030 needs to be generated. This can be performed with different methods with varying levels of details according to the available data. This spans from just scaling the population according to demographic evolution forecasts to taking currently planned housing projects in order to locate the new households and future activity locations (new workplaces, shopping centers, schools and universities, ...). Also, different alternative socio-economical assumptions can be taken into account to produce different future synthetic populations.

Regarding our scenario building process, we have analyzed the impact of future lines on the CPS level while still taking into account their impact on the Île-de-France level by simulating first on the latter then cutting the resulting population around the former. Besides the GPE, we have only considered the future line Tram 12 as a major part of it lays within the CPS area. Other currently planned tram lines for which similar data are available will be considered in the future.

A missing point in this work is the integration of bus lines to serve stations of the future rail-based ones. Rather than being due to a lack of data per se, future bus lines are planned in much shorter terms than rail-based systems which

require more complex work on the infrastructure. Through exchanging with the local authorities of the CPS area, it has come to our knowledge that bus lines were still in early design and discussion process at the time our simulation results became available. The full potential of the future offer is consequently underestimated as it can only be more attractive with buses providing access and egress. The work presented in Chapter 4 on intermodal MoD systems mitigates this issue without requiring additional data while providing useful insights on the demand for the future lines.

3.5 . Conclusion

In this chapter, future rail-based transportation lines that are planned for the Île-de-France region (Grand Paris Express and Tram 12) were considered. These lines were studied prospectively in an agent-based simulation and offers a first assessment of the impact of these lines on the CPS area. To our knowledge, this work is the first to specifically consider the GPE with a microscopic level of details. Relying solely on open tools, data and methodologies, this research as well as the simulation results are fully reproducible. Many development points have been identified in Section 3.4 and will be further investigated in future works: generate a prospective synthetic population, include all the changes that are planned on rail-based transit network in Île-de-France[10] and consolidate the related assumption, use a better calibrated mode choice model. This work shows the feasibility and potential of using and extending existing tools and methodologies in order to build a tool that allows to perform comprehensive and localized studies related to future transport systems planning. In the present use case, such a tool would allow to evaluate different strategies for the Grand Paris Express taking into account economic, ecological and social metrics.

[10] https://www.iledefrance-mobilites.fr/le-reseau/projets

4 - Intermodal MoD feeder system

This chapter presents the work performed in this thesis that addresses challenges in the simulation of MoD systems. Section 4.1 introduces the chapter with an overview of the current practices in agent-based simulations of MoD systems and highlights the gaps that are addressed in this work.

Section 4.2 details the developments performed to achieve simulations of intermodal MoD systems that act as feeder service for rail-based public transports. Building upon the simulation scenarios presented in Chapter 3, such a service is evaluated on the area of Paris-Saclay along the future rail-based offer. These developments and results have been presented alongside the ones detailed in Chapter 3 at the 102^{nd} annual meeting of the Transportation Research Board (TRB 2023) Chouaki et al. (2023).

Section 4.2 attempts to fill a gap identified in the literature regarding rejection rates of MoD systems and their integration in the mode choice model. The proposed novel approach to address this issue yields promising and insightful results while not requiring extensive parameter tuning and more importantly without increasing required simulation time. This work has been the subject of a conference paper that is accepted for presentation at the 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS) between 14 and 16 June 2023.

4.1 . Introduction: current state of the art on the simulation of MoD systems

Mobility-on-Demand (MoD) has been introduced in Chapter 2 alongside the current state of the literature in this topic. In this section, only concepts that are necessary to grasp the subjects addressed by the work at hand are recalled and elaborated.

MoD refers to a type of mobility system that consists in a fleet of on-demand vehicles that are available to the users as needed, offering more flexibility and an alternative to owning private cars. The development of MoD is closely related to the development of autonomous vehicles.

Various agent-based simulation tools and frameworks with different capabilities and levels of details have been proposed, many of which are open-source (Saidallah et al., 2016). Consequently, several works based on agent-based simulations and considering MoD systems can be found in the literature (Hörl et al., 2021; Narayanan et al., 2020). Different aspects of MoD systems are addressed in the literature from fleet sizing (Balac et al., 2020; Vosooghi et al., 2019a) to operation strategies (Hörl et al.; Zardini et al., 2021). Another difference between performed studies lies in the features external to the MoD system that are taken or not taken

into account. Such as congestion and the availability of other transport modes to the users (private cars, public transport). When the latter is considered, mode choice mechanisms are used to allow users to switch between modes in reaction to the perceived interest of each alternative.

This is the case in the MATSim simulation framework, used throughout this thesis, in which MoD systems implementation are offered by the DRT module (Bischoff et al., 2017). Trips are scheduled on the fleet side only when the request is made (typically at the trip's departure). The module considers parameterized constraints of maximum waiting times and detour factors that should not be violated. This can result in requests being rejected if no nearby idle vehicle is available. The maximum wait time constraint means that a request can be rejected if no idle vehicle is near enough to the request departure point. Consequently, the initial locations of the vehicle as well as the rebalancing strategy for idle vehicles play a major role in the service performance.

The state of the tool at the beginning of this PhD work allows MoD systems to be considered only as a completely separate mode that is available to users as a choice alternative evaluated by the mode choice model in the replanning step. Thus preventing the community from easily studying MoD systems in intermodal settings. A rare example can be found in Bürstlein et al. (2021), where a MoD system for first and last mile transit towards and from regional train stations is studied using macroscopic simulation. Other public transport alternatives in the study area are ignored and trips from/to the concerned train stations are set to take the MoD system. The MoD system considers constraints of maximum wait times and detour factor which results in potentially rejected requests. The service was tested with different fleet sizes, vehicle capacities and values for the maximum wait time and detour factor parameters. The impact of each configuration on the percentage of rejected trips is discussed. The number of rejected requests is differently considered in Chouaki et al. (2022) where it is considered as an objective function to minimize through proper rebalancing of idle vehicles (good strategies are learned using reinforcement learning).

Regarding the integration of MoD systems in the DMC models, the existing methodologies and software functionalities take into account only incomplete and imprecise components of the system's performance. More particularly, the possibility of a request being rejected is not at all considered when evaluating a MoD trip. Previous works performed with DMC models and featuring MoD rely mainly on estimated travel and wait times for computing the utilities of MoD trip (Horl et al., 2019). However, the estimations are performed regardless of the state of the fleet at the time the trip would actually take place and thus do not take into account the possibility of the request being rejected. This results in a situation where simulating two services, with different rejection rates and similar performance of non rejected requests, resulting in the same mode share of MoD. The equilibrium reached by the simulation can then feature agents choosing MoD for a trip while

the request is highly likely to be rejected. Consequently, the percentage of requests rejected by a MoD system is at most taken into account as an indicator of the performance of tested settings.

In this chapter, we extend the literature around agent-based simulations of MoD systems, alongside the MATSim software architecture, in the aspects identified above. First, we showcase how agent-based simulations can be used to investigate the use of a MoD system as an intermodal service that acts as a feeder to rail-based public transports. This is especially helpful for the assessment of the potential of planned future rail-based lines in the absence of data regarding future complementary buses. In Chapter 3, the results of prospective simulations of the future mobility offer of the area of Paris-Saclay (CPS) were presented (Grand Paris Express and Tramway T12). One identified missing point in Section 3.4 is the complementary mobility systems that will serve as feeders for the future rail-based lines. This issue will be addressed here using an intermodal MoD system.

The other major contribution is related to the integration of the MoD service performance into the mode choice models. We propose a novel approach to mitigate the lack of consideration of rejection rates in a way that is flexible without too much parameter tuning while providing results that can be interpreted in various dimensions.

Consequently, this chapter is structured as follows: Section 4.2 presents the integration of a MoD system as a feeder for rail-based lines, in the GPE+T12 scenario on the CPS level presented in Subsection 3.2.3, including details on intermodal routing and mode choice. Simulation results are presented with the updated impact of future rail-based lines in combination with MoD systems. Section 4.3 details our novel approach for taking into account MoD request rejection rate in the DMC model. Section 4.4 concludes and outlines research perspectives.

4.2 . Integrating an intermodal MoD feeder service in the Paris-Saclay simulation

4.2.1 . Technical implementation of intermodal MoD

Extending the open-source MATSim software architecture used in this thesis to reach the research goals presented in the previous section poses several technical challenges. With the objective of implementing the intermodal MoD system as a new mode available to travellers, two components are required: (i) The routing procedure that computes routes for the new mode. Such a procedure essentially develops an intermodal MoD trip into its base components (public transport and pure MoD) that can be carried on in the simulation by their respective modules. (ii) The integration of the new mode in the mode choice model and the utility calculation procedure allowing to evaluate an intermodal MoD trip and compare it to other alternatives. The work performed on both aspects is detailed below.

Routing

In MATSim, each mode available in the simulation is associated with a routing procedure. Car trips are routed using a shortest path algorithm that use the travel time as a cost. The travel time being dependent on the mode choices of other agents, the values observed in the previous iterations are used. Active modes such as walking and biking are simply teleported with a fixed time dependent on the distance and travel speed. Public transport trip routes are computed using the RAPTOR algorithm described in Delling et al. (2015). This algorithm is provided by the SwissRailRaptor extension made available by the Swiss Federal Railways and offers the possibility to use other modes as access and egress for public transport trips. However this feature is not used in our work due to technical reasons.

Routes using our intermodal MoD mode between an origin and a destination are computed as follows: (1) The closest rail stations (alongside subway, tramway and train lines) are identified and selected as access and egress stations. To optimize execution time, the set or rail stations is computed only once at the beginning of the simulation and put in a Quadtree structure to increase the speed with which closest stations are retrieved. (2) MoD trip segments are planned from the trip's origin to the access station and from the egress station to the trip's destination. (3) The segment of the route between the access and egress stations are computed with MATSim's standard router using the RAPTOR algorithm. (4) All segments (access, transit, egress) are chained in order to obtain the intermodal route. Note that, in some cases, the access or egress segments are reduced to walk-only trips if the origin and destination are near enough from each other. However, we require that at least one of these segments concretely uses the MoD fleet. Consequently, some of our intermodal trips have either an access MoD part or an egress one instead of both.

Demand estimation and mode choice

As for the demand for MoD, we rely, in this work, on the synthetic population generation process previously used in Chapter 3. The output population of this step includes a demand for car, bike, public transport and walking that is calibrated to match real-world observation. This means that the input demand of simulations do not include MoD. Our intermodal MoD system is integrated in the simulation as a new mode, completely separate from regular public transport. Demand for the system arises through the iterations by effect of the replanning step where MoD is considered as a possible alternative and evaluated and chosen when interesting as depicted in Figure 2.2.

The utility of an intermodal feeder trip alternative $\mathbf{x}$ is estimated by first dividing $\mathbf{x}$ into its components: The two access and egress MoD parts $\mathbf{x}_{MoD1}$ and $\mathbf{x}_{MoD2}$ and the public transport part $\mathbf{x}_{pt}$ and then summing their utilities.

$$\tilde{v}_{\text{MoD*}}(\mathbf{x}) = \tilde{v}_{\text{MoD}}(\mathbf{x}_{MoD1}) + \tilde{v}_{\text{MoD}}(\mathbf{x}_{MoD2}) + \tilde{v}_{\text{MoD}}(\mathbf{x}_{pt}) \tag{4.1}$$

The utilities for each components are estimated using mode specific formulas presented in Hörl et al. (2019):

$$\begin{aligned}\tilde{v}_{\text{MoD}}(\mathbf{x}) = \beta_{\text{ASC,MoD}} \\ + \beta_{\text{travelTime,MoD}} \cdot x_{\text{travelTime,MoD}} \\ + \beta_{\text{waitingTime,MoD}} \cdot x_{\text{waitingTime,MoD}} \\ + \beta_{\text{cost}} \cdot p \cdot x_{\text{distance,MoD}}\end{aligned} \tag{4.2}$$

$$\begin{aligned}\tilde{v}_{\text{pt}}(\mathbf{x}) = \beta_{\text{ASC,pt}} \\ + \beta_{\text{numberOfTransfers}} \cdot x_{\text{numberOfTransfers}} \\ + \beta_{\text{inVehicleTime}} \cdot x_{\text{inVehicleTime}} \\ + \beta_{\text{transferTime}} \cdot x_{\text{transferTime}} \\ + \beta_{\text{accessEgressTime}} \cdot x_{\text{accessEgressTime}} \\ + \beta_{\text{cost}} \cdot \left(\frac{x_{\text{crowflyDistance}}}{\theta_{\text{averageDistance}}}\right)^{\lambda} \cdot x_{\text{cost}}\end{aligned} \tag{4.3}$$

Public transport trips can be exactly estimated in advance given that our simulations do not consider delays of public transport and suppose enough vehicle capacity. The computation of $\tilde{v}_{\text{MoD}}(\mathbf{x})$, on the other hand, relies on estimations for travel and waiting time. In this stage of the study, the utility computation of the MoD sub-trips estimates the waiting and travel times to be the maximum values allowed under the constraints. Performed MoD trips are then of better utility than what is estimated. However, this method does not take into account the possibility of the request being rejected during the actual simulation. Consequently, we focus our evaluation of the feeder fleet in configurations with enough vehicles as to have very few rejected requests. In section 4.3 we present pathways to take into account MoD request rejections more realistically.

4.2.2 . Scenario building

The scenario building methodology presented in 3.2.3 is extended here by adding a scenario comprising an intermodal MoD system. We call this scenario *Feeder* as depicted in Figure 4.2.

The fleet of MoD vehicles that is considered here consists of 400 4-seated vehicles that are scattered throughout the network. Choice was made in this work to evaluate a MoD system under dimensions that achieve the minimum rejection rates. This is mainly motivated by the fact that the DMC model for MoD used in this study does not take into account the possibility of a request being rejected and thus supposes a service that is always available. Various fleet sizes have been tested as shown in Figure 4.1. The rejection rate observed with 400 vehicles is less than 5% and does not greatly decrease with greater fleet sizes.

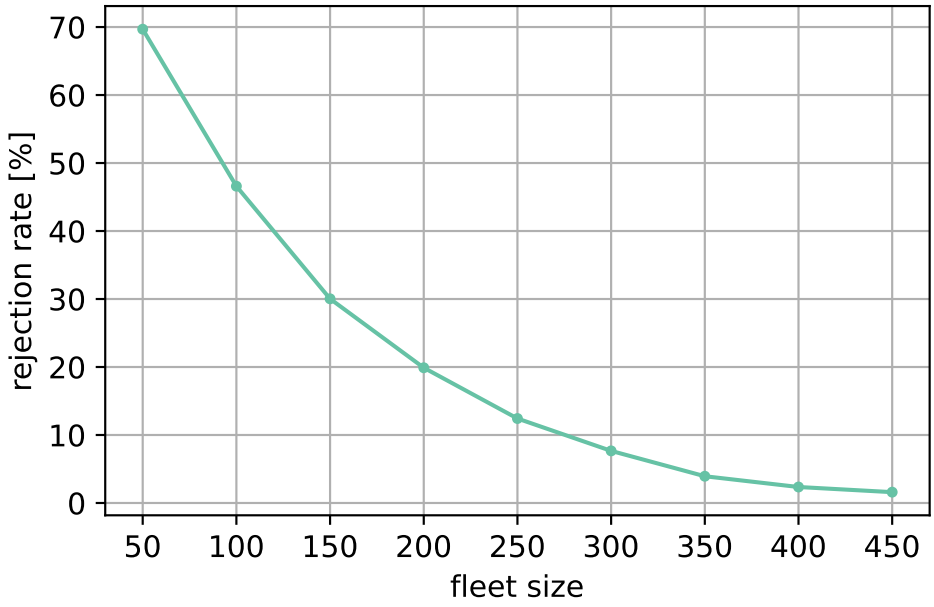

Figure 4.1: Observed MoD rejection rates with fleet sizes between 50 and 450 vehicles

Vehicles are assigned to requests using the insertion algorithm included in the DRT module of MATSim. Empty vehicle relocation is managed by the algorithm presented in Ruch et al. (2020). In the beginning of the simulation, MoD vehicles are initially positioned near rail-based stations which are concerned by the intermodality of the system. This is done while ensuring that the number of vehicles besides each station is proportional to the number of travelers that are observed in the station, while ensuring at least one vehicle per station, during the CPS GPE+T12 simulation. This constitutes the reason for adding the fleet only after the GPE+T12 scenario is simulated as passenger traffic per station is needed. Moreover, we ensure that the vehicle locations are reinitialized at the start of each iteration.

As detailed in 3.2.3, cutting the Île-de-France level simulation to generate the CPS level one has implications on the mode choice. Trips from or to the outside of CPS are represented but cannot be subject to mode choice to keep the consistency of the overall trip. However, with the introduction of the intermodal MoD system, one exception was added to this restriction. That is the switching between intermodal MoD and pure public transport. By allowing public transport trips from or to outside of CPS to be replaced by an intermodal MoD trip, while ensuring that the latter enters or leaves CPS with a public transport part, does not break any constraint on the global level.

A common practice in the MATSim literature is to downscale the synthetic population to represent a fraction of the real one. This allows to save computation

Line	Baseline Ridership	Δ (abs) GPE+T12	Δ (%) GPE+T12	Δ (abs) Feeder	Δ (%) Feeder
108	1483	-663	-44.71	-798	-53.81
107	23159	-8145	-35.17	-9503	-41.03
197	1032	-283	-27.42	-96	-9.30
199	8066	-1872	-23.21	-3712	-46.02
3	3942	-825	-20.93	-895	-22.70
91-02	1691	-308	-18.21	-495	-29.27
91-05	5806	-890	-15.33	-1792	-30.86
399	11961	-1325	-11.08	-2648	-22.14
297	14896	-1548	-10.39	-2934	-19.70
299	1861	-192	-10.32	-376	-20.20
1	10261	-949	-9.25	-1685	-16.42
18	1171	-105	-8.97	-253	-21.61
RER C	18611	-1629	-8.75	605	3.25
2	11725	-968	-8.26	-4005	-34.16
17	10247	-791	-7.72	-2593	-25.30
23	6336	-457	-7.21	-2170	-34.25
DM153	11965	-664	-5.55	-2720	-22.73
DM11E	2550	-103	-4.04	-613	-24.04
DM11 A	5865	-128	-2.18	-1450	-24.72
22	3620	-73	-2.02	-1079	-29.81
DM12	6002	-120	-2.00	-1219	-20.31
DM10 A	2930	-40	-1.37	-675	-23.04
DM11 C	3631	-49	-1.35	-1254	-34.54
DM13	1404	-13	-0.93	-415	-29.56
RER B	32349	-148	-0.46	9737	30.10
39-07	3394	71	2.09	-576	-16.97
11	6492	379	5.84	-1050	-16.17
91-11	4336	886	20.43	305	7.03
119	9908	2981	30.09	2139	21.59
Tram 12	-	18478	-	20814	-
Metro 18	-	15062	-	15757	-

Table 4.1: Observed riderships of public transport lines in the CPS area in the Baseline, GPE+T12 and Feeder scenarios

resources and execution time. However, we choose not scale down the synthetic population. This is mainly motivated by the impact such a scaling has on the assessment of the system's performance and the fleet sizing. Whereas the road network and public transport's capacities can be reduced by the same factor with which the population is sampled, the capacity of MoD vehicles be reduced since it is already relatively low. On the other hand, research has shown that population downscaling does not allow to fairly assess a shared MoD service even if the fleet size is downscaled accordingly Kaddoura and Schlenther (2021); Kagho et al. (2022). Consequently, and considering our research objective, we have conducted our experiments with a non-scaled population as mentioned in Subsection 3.2.3.

4.2.3 . Results

As with Metro 18 and Tram 12 introduced in the GPE+T12 scenario, introducing the intermodal MoD system in our setting does not have a substantial impact on the mode shares. However, it greatly improves the attractiveness of its related rail-based public transport lines. Figure 4.3 shows number of trips per type of pub-

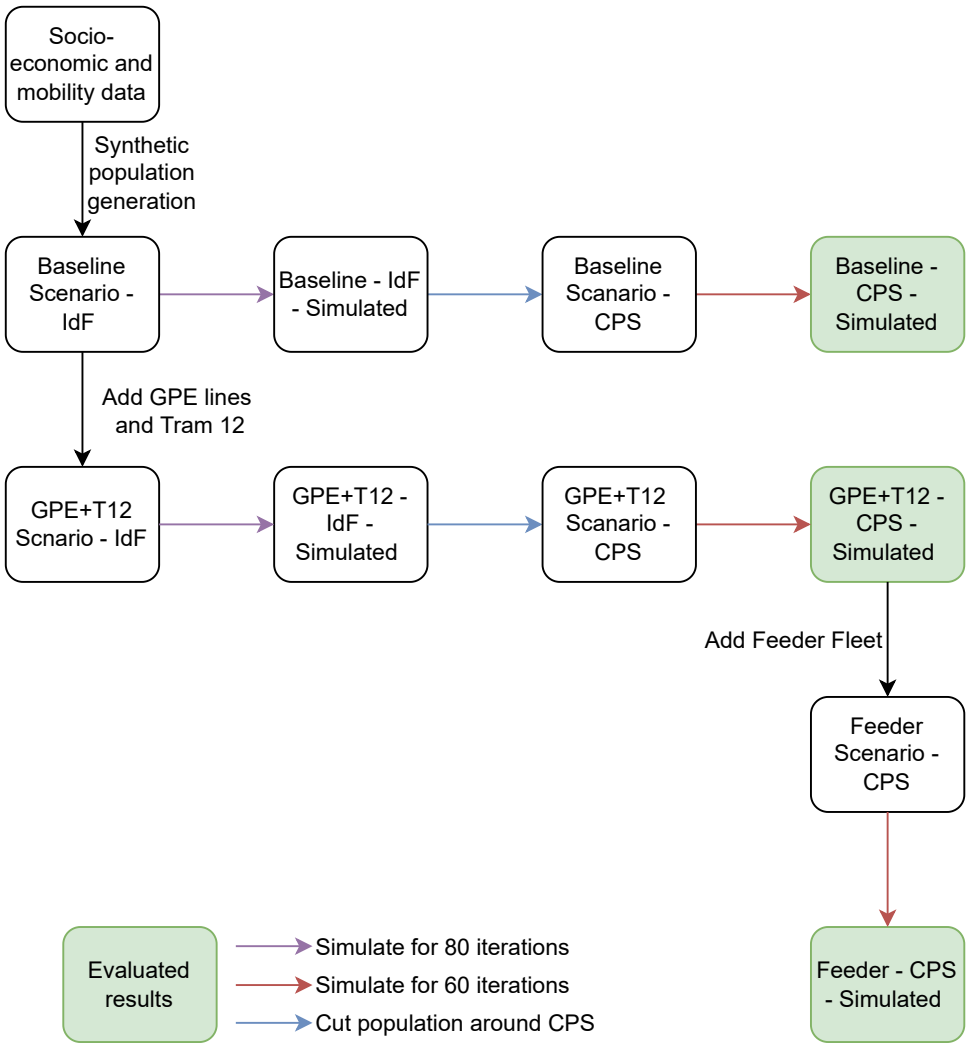

Figure 4.2: Overview of the methodology of scenario building extending the one presented in 3.3 to integrate the intermodal MoD service

lic transport observed on Baseline, GPE+T12 and Feeder scenario simulations on the CPS level. Overall, the introduction of the intermodal MoD service increases the attractiveness of the new offer (tram and metro) as well as the existing rail-based modes (RER B and RER C). This increase is more noticeable in the latter one. The interest of buses in general is further decreased in comparison to the GPE+T12 scenario.

Table 4.1 dives deeper in this analysis by presenting the updated number of public transport users per line for the same scenarios on the CPS level. RER C's travelers increase by 3.25% after losing 8.75% in the GPE+T12 simulation while RER B sees an increase of 30.10% in number of trips compared to the Baseline simulation. This can be explained by the fact that the RER B is the main link between the CPS area and Paris. As for the ridership of bus lines, even though they decrease in general, some bus lines witness more travellers after the introduction of the MoD service (bus 119 and 91-11).

Figure 4.4 shows the distribution of the origins of MoD trips that are performed as part of routes computed using the feeder mode. The vast majority of these trips originate near the public transport stations that are linked to the feeder system but a portion of those trips also originate from other areas throughout the CPS.

Regarding the usage of the MoD service, 42203 MoD trips are requested in the Feeder simulations. Among them, 1016 are rejected because no vehicle is available to perform the trip while satisfying the quality constraints of waiting time and detour factor. Figure 4.5 shows the load of MoD vehicles during the day. Vehicles can be in 3 states: (i) STAY if the vehicle is idle waiting to be assigned to a trip. (ii) RELOCATE if the vehicle is currently moving to another position, suggested by the rebalancing algorithm, to then wait to be assigned. (iii) SERVING if the vehicle is performing one more user trips. The vehicle is transporting no passenger when on the first two states. On the third one, the number of passengers can vary between 0 and the maximum vehicle passenger capacity (4 in this case). Our results show a relatively high use of the fleet with the system almost reaching capacity during peak time. The level of ride-sharing is also high with a large proportion of vehicles transporting more than one passenger. This can be explained by the strictly intermodal setting of the service for which the demand is strongly related to the public transport schedule. The likelihood of having multiple requests originating from the same place at the same time is much larger since the service is accessible to travellers who leave rail-based transports.

4.2.4 . Discussion

Our simulation results clearly show the potential of using an intermodal MoD service that acts as a feeder for rail-based public transport. We discuss below our methodology and the assumptions that were made while building the simulation scenarios.

In this work, for the simulations on the CPS level, the mode choice constraints related to trips between CPS and the rest of Île-de-France are configured to allow

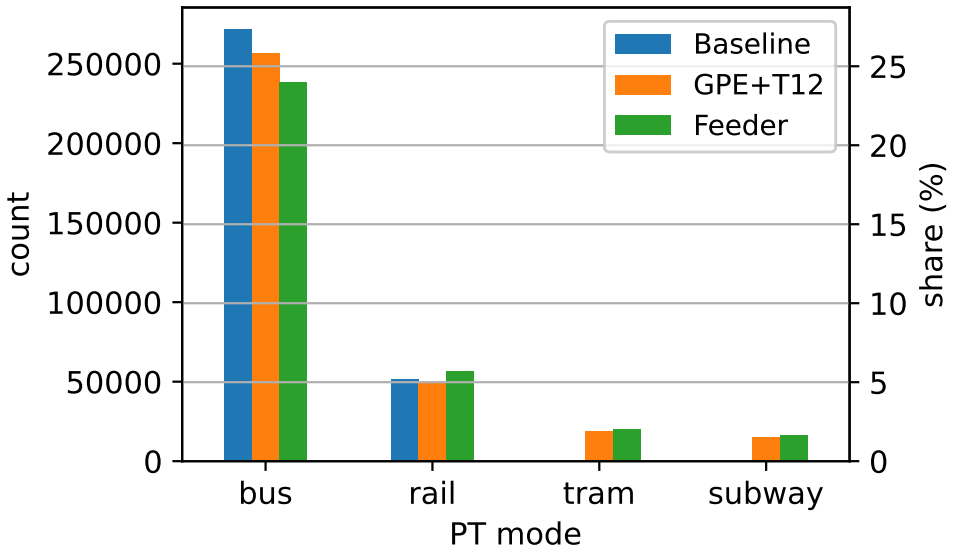

Figure 4.3: Trip counts per PT sub-mode compared over the Baseline, GPE+T12 and Feeder scenario simulations

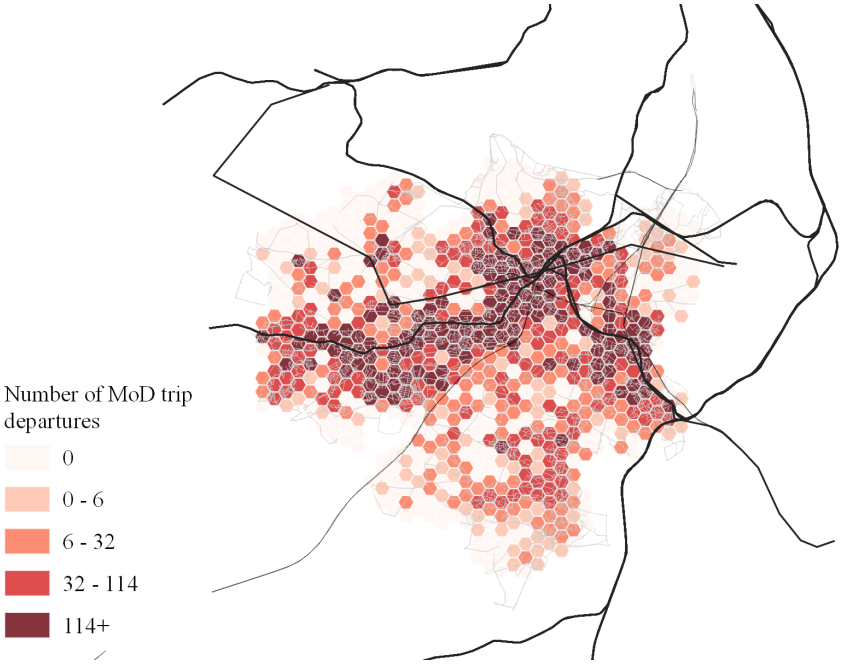

Figure 4.4: Distribution of MoD trip departures in the Feeder scenario simulation

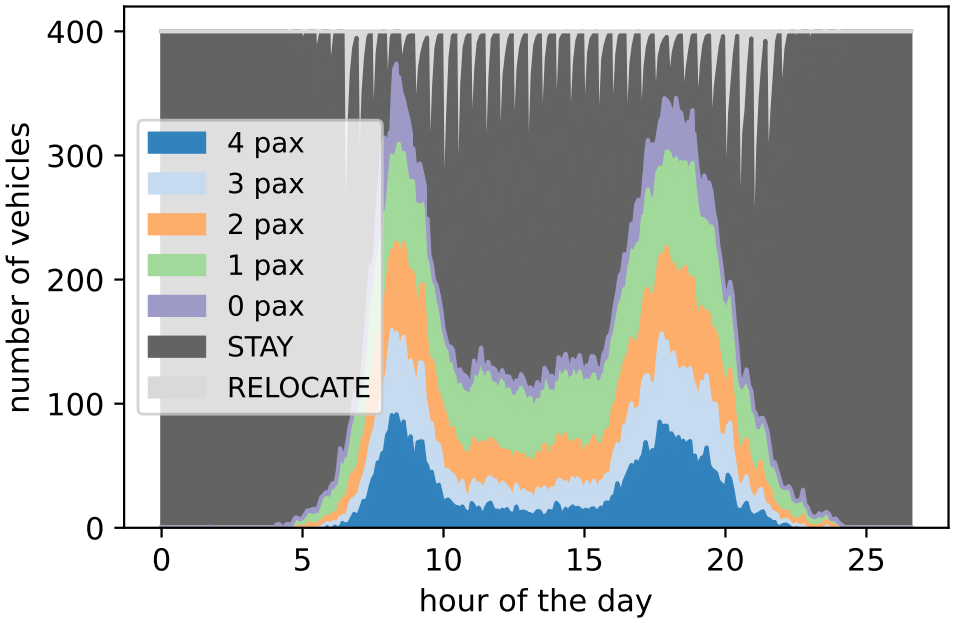

Figure 4.5: Overview of usage of the MoD fleet vehicles

transitions only between using regular public transport or the feeder system which combines the latter with using the MoD fleet. Switching from car to feeder is not allowed in order to preserve car availability constraints. This restrains the potential impact of the system due to the fact that agents that come to the area by car cannot decide to use MoD in this setting. This aspect will be taken into account in future works.

Also, the utility computation of the MoD segments of a trip does not take the pricing of this service that it is included in the public transport fare. Moreover, public transport and MoD are valued in the same manner. This hypothesis can be challenged and adjusted, for instance also with recent survey data Berrada et al. (2020).

In addition, the probability of rejections or the actual travel and wait times that are observed for the service are not taken into account (the utility is optimistic in regards to the former point and pessimistic in regards to the latter). Consequently, the performance of the system and underlying operation strategies do not impact its attractiveness. A better method for computing these utilities is explored in 4.3.

In general, more effort can be put in the calibration of the mode choice parameters. As for the routing of trips with the feeder mode, we currently chose the stations belonging to the relevant lines that are the closest to the trips' origins and destinations respectively as access and egress stations. Finding the globally fastest route would be greatly more time consuming as all couples of access and egress stations need to be evaluated. Moreover, the durations of the MoD segments of

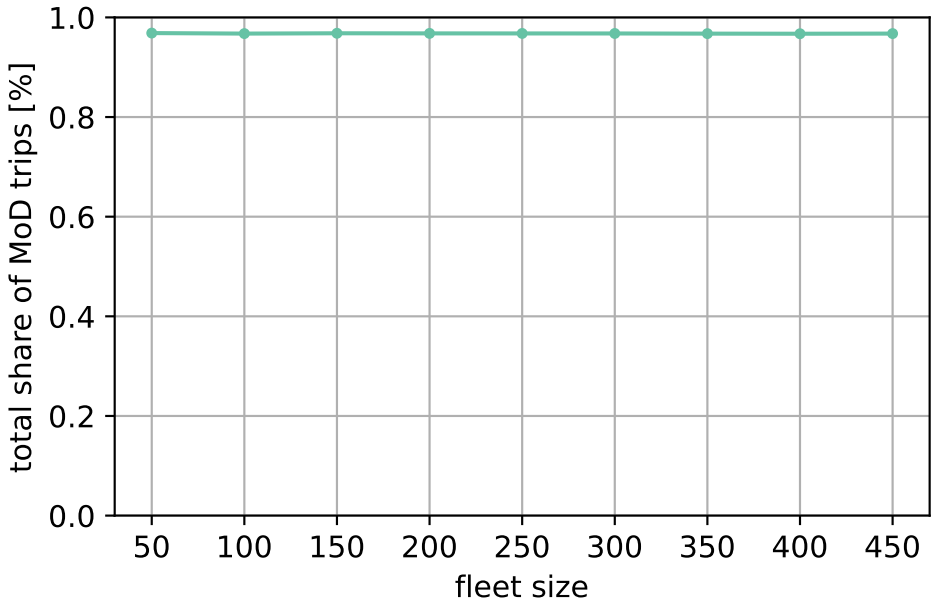

Figure 4.6: Total shares of MoD trips in function of fleet size

the trips can only be estimated before they are effectively performed. One interesting pathway of research is to consider the use of reinforcement learning based techniques for access and egress station selection. Finally, the evaluation of the MoD system can be more extensive and include the analysis of vehicles' occupancy and empty-to-total mileage. Also, the use of the MoD fleet can provide insight for future bus lines planning and a basis on which these lines can be implemented and tested in simulation.

4.3 . Integrating MoD trip rejections in the mode choice

In Section 4.2, an intermodal MoD system acting as a feeder for rail-based public transport was introduced in a simulation of the area of CPS that includes existing public transport and future rail-based lines. Results show the interest of the MoD system in increasing the attractiveness of concerned public transport lines. The fleet was sized appropriately to avoid the impact of high rejection rates could have on the demand. The current state of the art around mode choice models for MoD systems greatly lacks the integration of MoD trip rejections. In practice this results in simulations with largely different fleet sizes (and thus largely different rejection rates) to stabilize at the same demand for MoD. This is illustrated in experiments with different fleet sizes for the intermodal MoD system studied in Section 4.2 as shown in Figure 4.6.

In this section, we propose a method to integrate rejection rates into an endogenous demand simulation model. Subsection 4.3.1 formally introduces the type of

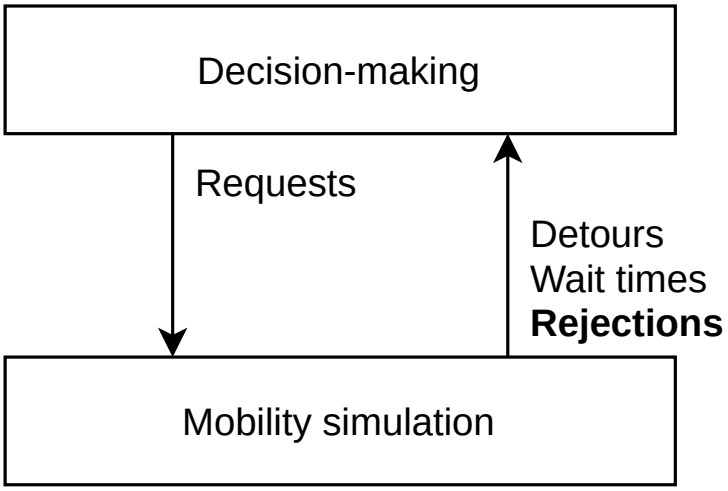

Figure 4.7: Endogenous demand loop

simulations our approach can apply to. Potential solutions for integrating rejection rates are discussed, and their individual shortcomings are identified. Subsection 4.3.2 presents our method based on a linear proportional controller to address the identified issues. In Subsection 4.3.3, results of our approach on our use case are shown. Subsection 4.3.4 discusses our approach, edge cases, and proposes policy-relevant interpretations.

4.3.1 . Problem

Formally, we consider simulation set-ups as shown in Figure 4.7. In an iterative way, a set of requests is generated using a demand model, which is then processed in a mobility simulation. In a closed loop, the obtained information on waiting times, detours, and also the rejection rate, are fed back to the decision process in order to generate a new adapted set of requests. Generally, low levels of service will lead to lower demand while reduced wait times, travel times and rejection rates should yield higher demand.

In most endogenous demand studies, discrete mode choice models are used. In those models, users can choose for each of their trips between a set of modes, for instance, using the car, public transport, using the bicycle or walking. For each mode, a utility function is defined, with higher utilities indicating more interesting alternatives. This is the case in the work presented in section 4.2 where the formula in Equation 4.3 is used to evaluate MoD trips.

In this equation, variables denoted as x describe estimated choice dimensions for a specific trip, and behavioral parameters (usually negative) denoted as β quantify their share in the overall generalized costs for the trip. Furthermore, a price p is defined, and similar utility functions are defined for the other modes of transport

that compete with the on-demand service.

A common discrete choice model type that is also used in Hörl et al. (2019) is the multinomial logit model which yields a probability to choose each alternative based on the calculated utilities for each mode Train (2009). Additionally, an availability $a_k \in \{0, 1\}$ can be defined which defines whether an alternative is available for a specific trip k or whether the choice probability is zero.

While some choice dimensions (like the distance) can be obtained directly through routing, others, such as the waiting time can be estimated from the mobility simulation, for instance in time bins and on a hexagonal grid such as in Hörl et al. (2019) or Hörl et al. (2021).

Additionally, we now assume that a fleet control algorithm with rejections is used in the simulation, which allows us to calculate a global rejection rate $\rho \in [0, 1]$ as the quotient of the rejected number of requests and the total amount.

The resulting question is how to feed back this rejection rate like the other choice dimensions. In the following, let $i \in \mathbb{N}$ denote the currently executed iteration from which information has been obtained. Intuitively, by extending the current implementation, observed rejection rates can be taken into account in various ways in order to achieve r^* in relatively straightforward manners. We present in the following a few alternatives:

(i) The rejection rate can be integrated into the utility function as another **choice dimension**, weighted by the parameter $\beta_{\text{rejection}}$:

$$v_{i+1} = \tilde{v}(\mathbf{x}_i) + \beta_{\text{rejection}} \cdot \rho_i \tag{4.4}$$

Economically, the formulation is not realistic, as a rejection rate of 100% should lead users to never consider the service. However, then $\beta_{\text{rejection}}$ should be quite high, and it is not intuitively clear which value to put for the parameter. There is also a conceptual contradiction: The idea of the parameter is that $\beta_{\text{rejection}} < 0$ would penalize rejections. However, if a state is reached at which rejections are avoided, ρ_i will be zero and the additional term will have no impact.

(ii) The **availability** of the MoD mode can be sampled according to

$$a_{k,i+1} \sim \text{Bernoulli}(1 - \rho_i) \tag{4.5}$$

indicating that the number of trips for which MoD is available follows the acceptance rate. However, applying this approach will not yield the desired rejection rate in the following iteration, since choices should be discarded *after* being chosen based on the utilities. This could be a viable approach, but would imply complex structural changes to the choice process.

(iii) Similar to the first option, a new **penalty** term $\pi \in \mathbb{R}$ can be introduced to the utility function:

$$v_{i+1} = \tilde{v}(\mathbf{x}_i) + \pi_{i+1} \tag{4.6}$$

Being a simple constant (but iteration-dependent) term to the utility function, it can be interpreted easily in a multinomial logit context and even converted to monetary units. If π_{i+1} is chosen close to zero, there will be little impact on the overall demand and the rejection rate. If a strongly negative value is chosen, demand will be suppressed and (assuming same fleet size) the rejection rate will decrease. We assume that a value π^* can be found that pushes the rejection rate below a predefined threshold ρ^*.

Due to its flexibility and interpretability, we choose the last approach. The correct value of π^* to reach a rejection rate of ρ^* depends on external configuration inputs, such as the fleet size. One would, hence, need to run multiple simulations for each new configuration to find a near-optimal π^* and even repeat the process for different target rates ρ^*.

Therefore, we propose an iterative approach to auto-calibrate the parameter π^* over the course of one iterative simulation.

4.3.2 . Approach

The approach proposed in this paper draws from the field of control theory and concretely makes use of a Linear Proportional Controller Johnson and Moradi (2005) in order to dynamically find a penalty that drives the rejection rate below ρ^*.

The control process is formalized by defining an update rule for the penalty:

$$\pi_{i+1} = \pi_i + \Delta\pi_i \tag{4.7}$$

In a purely linear control application, a proportional controller can be defined through the update rule

$$\Delta\pi_i = K \cdot \Delta\rho_i \quad \text{with} \quad \Delta\rho_i = \rho^* - \rho_i \tag{4.8}$$

leading to a positive update if the current rejection rate ρ is too low and to a negative update otherwise, assuming a positive proportional gain $K > 0$, which specifies how fast the penalty is adjusted.

For stability reasons (see below), we propose an extended update rule as

$$\Delta\pi_i = \begin{cases} K \cdot \Delta\rho_i & \text{if} \quad \rho_i \geq \rho^* \\ 0 & \text{else} \end{cases} \tag{4.9}$$

Accordingly, the penalty is only updated if the current value is larger than the target value. The initial penalty π_0 is set to zero.

This approach is implemented in the Eqasim architecture as part of the mode choice model. The controller is configurable through the XML configuration of MATSim as shown in Figure 4.8. This allows to quickly test with various parameter values.

```
<module name="IDFDrtModule" >
        <param name="useFeeder" value="true" />
        <parameterset type="drtRejectionPenaltyProvider" >
                <parameterset type="drtRejectionsLinearPenaltyProvider" >
                        <param name="alpha"
                                value="0.3" />
                        <param name="enableBackwardAdjustment"
                                value="false" />
                        <param name="initialRejectionPenalty"
                                value="0.0" />
                        <param name="targetRejectionProbability"
                                value="0.15" />
                </parameterset>
        </parameterset>
</module>
```

Figure 4.8: XML code example for configuring the linear proportional controller

4.3.3 . Results

In the Feeder scenario presented in Section 4.2, only requests which have origin or destination at a rail-based public transport stop can be submitted. Furthermore, decisions are based on a tour-based (rather than a trip-based) model, which means that accumulated utilities over multiple individual trips, for instance, along a home-based round-trip, are considered. The control-based approach proposed in this study easily generalizes to such complex scenarios without any specific adjustment (as would be the case for the second approach described above).

In contrast to the results shown in the previous section, where rejection rates were merely reported, not fed back into the agent decisions, here we perform simulations making use of the proposed control approach. The simulations are performed with 400 iterations, requiring about 3 days of runtime on 10 cores and using 40GB of RAM for one full simulation. Therefore, an approach that is integrated into the simulation dynamics, rather than an offline calibration of π^*, is beneficial.

Figure 4.9 shows observed rejection rates through simulations of a fleet of 200 vehicles and different values of the target rate $\rho^* \in \{0\%, 2\%, 5\%, 15\%, 20\%\}$. The trajectory of the observed rejection rate is compared to the base simulation where no controller is used and where a demand is generated that leads to about $20^{\%}$ of the requests being rejected. With $\rho^* = 5\%$, Figure 4.9 shows that our approach is able to lower the demand as to obtain rejection rates below the value of r^*.

However, when the target rejection rate is very low ($\rho^* \in \{0\%, 2\%\}$), the observed rejection rates do not drop consistently below ρ^*. This is an artifact of the demand model, which is probabilistic and, even when the MoD mode is strongly penalized, may choose it for a specific trip. Those trips are usually characterized by

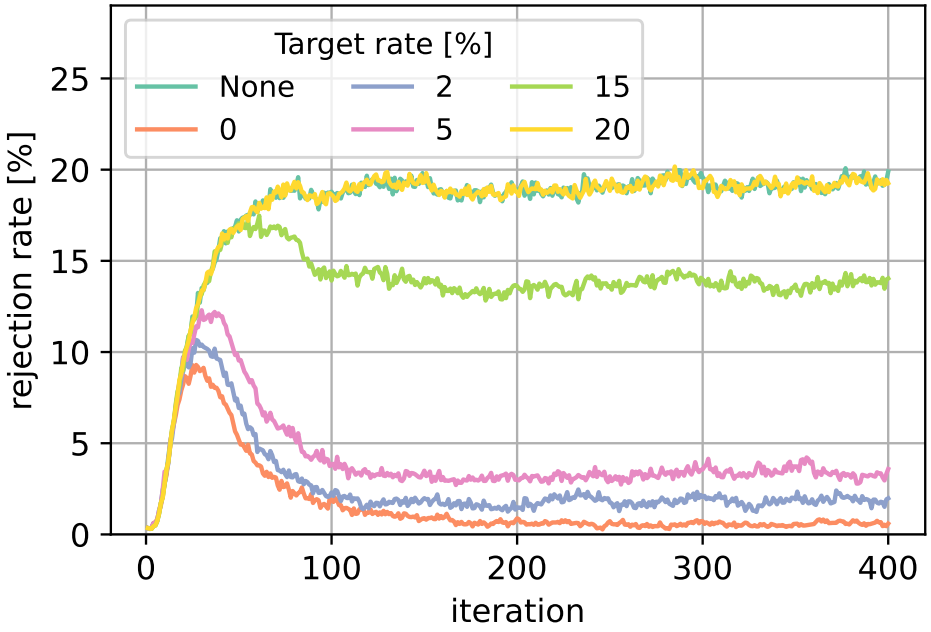

Figure 4.9: Observed rejection rates during simulations with 200 vehicles varying target rejection rates compared to the base simulation

poor other options, for instance, without car access, but also not having a viable public transport alternative. Based on the particular set-up of the case study, these trips are also those which are likely to be rejected by the operator because operational constraints (delay, wait time) cannot be fulfilled.

Moreover, we note that when the target rejection rate is above the one already observed in the base case ($r^* = 20\%$), our controller does not reduce the observed rejection rate.

These results are mirrored in Figure 4.10 where the evolution of the penalty in those simulations is shown. The penalty remains at zero when $\rho^* = 20\%$. As expected, the penalty decreases with lower values of ρ^*. For $\rho^* \in \{5\%, 15\%\}$, the converged value of the penalty is achieved after about 100 iterations. As the required rejection rate cannot be reached, for $\rho^* \in \{2\%, 0\%\}$, the penalty decreases continuously.

For properly chosen target rejection rates, these results show that our approach is able to find a penalty value that lowers the demand just enough to achieve an acceptable rejection rate for a fixed fleet size. From another perspective, it is interesting to analyze what penalty is needed to reach the same target rejection rate at different fleet sizes. In order to do so, we perform simulations with fleet sizes between 50 and 450 vehicles and a target rejection rate of $\rho^* = 5\%$.

Figure 4.11 shows the obtained penalty values after 400 iterations. With higher fleet sizes, the penalty approaches zero, indicating that after 350 vehicles the fleet

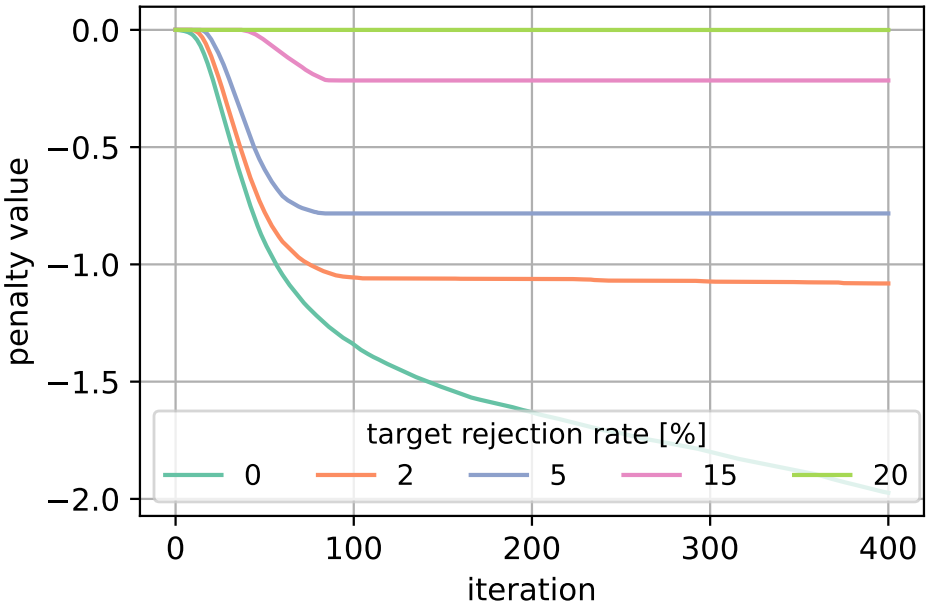

Figure 4.10: Evolution of the penalty during simulations with 200 vehicles varying target rejection rates

is large enough to provide rejection rates below 5%. Figure 4.12 shows the number of trips performed in these simulations. We notice that the when using the penalty, less trips are performed. This can be explained by the fact that the penalty applies equally to all MoD alternatives and is dependent on the global rejection probability, not the probability of a particular trip being rejected. This results in discarding some MoD trips that would not have been rejected. When using the penalty, the number of trips increases almost linearly with the number of vehicles below the fleet size of 350 vehicles, with a clear saturation of demand.

4.3.4 . Discussion

The results shown in the previous subsection demonstrate the effectiveness of our approach and its ability to lower the observed rejection rates below a certain value. However, the obtained rejection rate can be somewhat lower than the target (Figure 4.9). This is due to an overshooting by our method that reduces the penalty more than necessary. The overshooting behavior is directly related to the value of K that is used. Lower values for this parameter will overshoot less, but take longer to converge. The problem can be tackled by making use of the default update rule introduced in Equation 4.8.

Figure 4.13 shows the evolution of the penalty when using the standard controller. As is common for purely proportional controllers (Johnson and Moradi, 2005), the penalty value shows oscillations. Its amplitude seems to be decreasing, but the penalty does not reach convergence within 400 iterations. This issue could

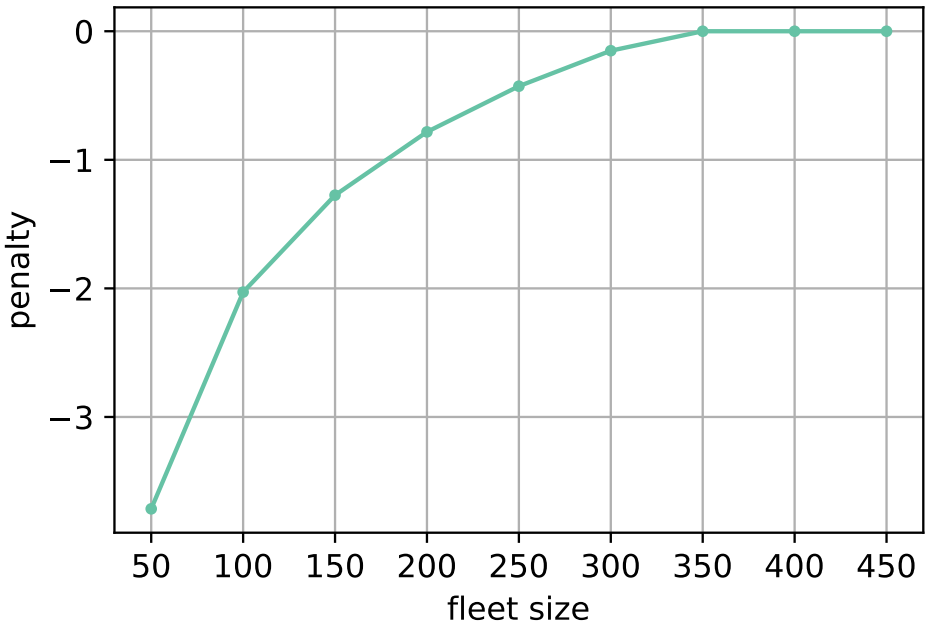

Figure 4.11: Obtained penalty value to ensure a rejection rate less than 5% with various fleet sizes

further be mitigated by making use of a PI (proportional integral) controller using the update rule:

$$\Delta\pi_i = K_P \cdot \Delta\rho_i + K_I \cdot \sum_{j=0}^{i} \Delta\rho_j \tag{4.10}$$

However, this approach would require additional (and computationally costly) fine-tuning of the two gain parameters K_P and K_I.

Our results also allow observations related to edge cases: When the fleet size is large enough to handle the demand without a penalty, the service does not become more attractive and the rejection rate always stays below the target. On the other hand, when the fleet size or the target rejection rate are too low, the penalty can decrease infinitely, and the demand is driven close to zero.

Regarding the interpretation of the obtained penalty values, a few alternatives are possible:

- The penalty can be understood as an addition to the alternative-specific constant (β_{ASC}) that is defined for the mode and captures preference effects that are not explained explicitly by the choice dimensions. However, this is a bare conceptual interpretation that does not allow any economic interpretation.

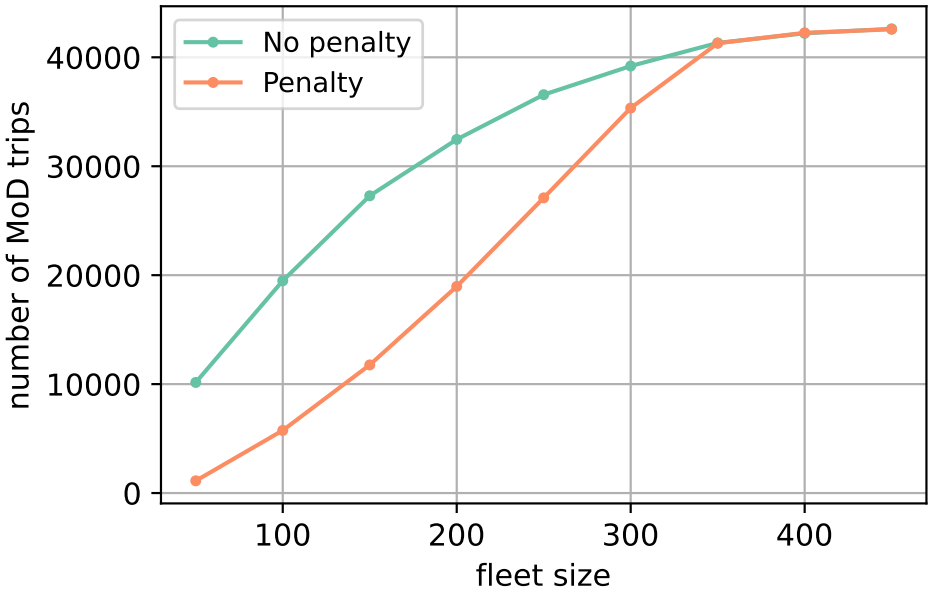

Figure 4.12: Number of MoD trips performed with various fleet sizes with and without using a penalty to ensure less than 5% of rejected requests

- In a multinomial logit model, the marginal effects β can be converted into marginal (monetary) costs by dividing the whole utility function by β_{cost} (Hörl et al., 2021). Hence, $\pi^*/\beta_{\text{cost}}$ represents a price. In this case study, a penalty of $\pi^* = -1$ would, according to the values of the choice model, represent an additional cost of about 4.85 EUR per trip. This is the price that an operator would need to ask to avoid excessive rejection rates in the system (for a given fleet size).
- Analogously, π^* can be translated into any other choice dimension in a multinomial logit model with a linear utility function. For instance, the penalty of $\pi^* = -1$ represents 20 minutes of additional wait time per trip in Chouaki et al. (2023). This is the delay that an operator can allow its vehicles if rejection rates should be capped.

An interesting development pathway for this method is to adjust penalties based on user characteristics, zones, or time of day. This would allow differentiating between situations with varying demand. When interpreting the penalty as a price, this idea links to the topic of dynamic pricing for MoD systems that has been addressed in literature (Saharan et al., 2020).

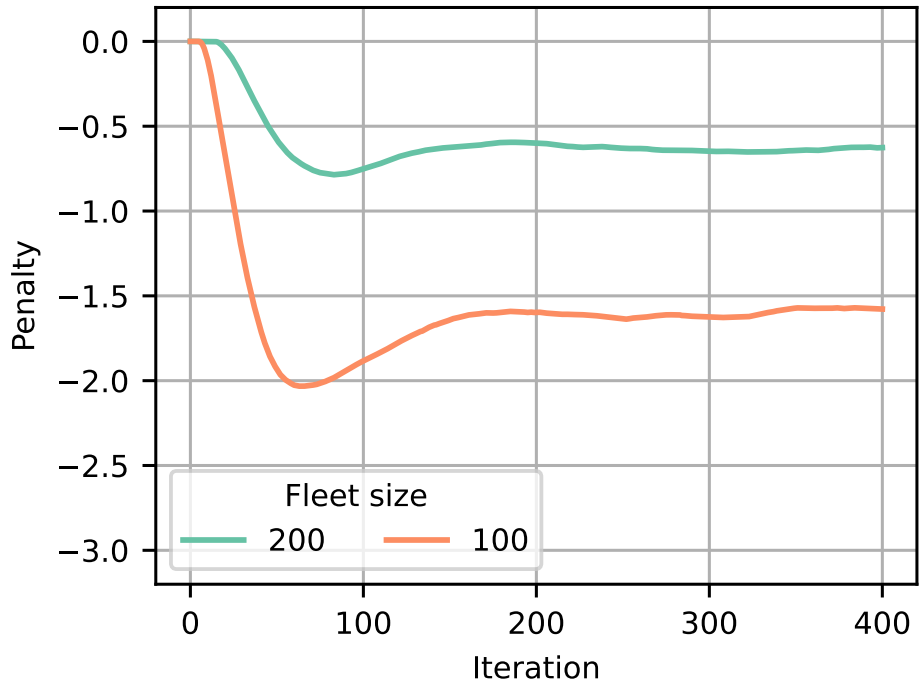

Figure 4.13: Evolution of the penalty when enabling backward adjustment during simulations with 200 vehicles and $r^* = 0.05$

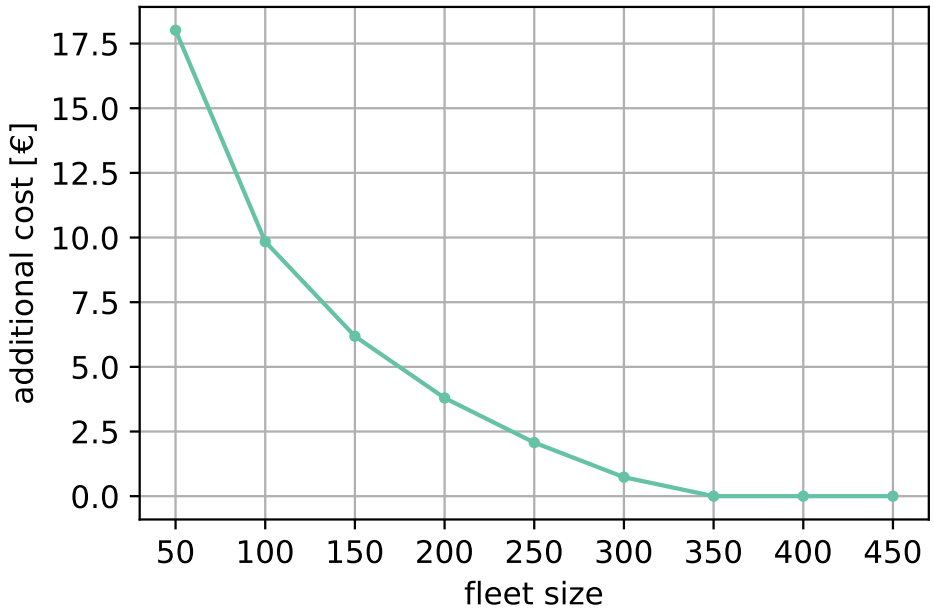

Figure 4.14: Penalty values interpreted as additional cost of service in EUR for various fleet sizes

4.4 . Conclusion

In this chapter, two challenges related to agent-based simulations of MoD systems are addressed. First, the simulation of an intermodal MoD system for access to and egress from rail-based public transports is explored in Section 4.2. This type of service is assessed on our CPS use case on top of the GPE+T12 scenario introduced in Chapter 3. Results show the interest of such a system in increasing the attractiveness of rail-based public transport lines as well as identifying where the demand for those lines emerges. Even if a MoD system is not actually adopted as a solution, this methodology can then be used to guide the design of more classical bus lines.

In Section 4.3, a novel approach for taking into account the rejection rate of a MoD service in the DMC model is proposed. This approach is evaluated on our CPS simulations and results show that it is able to achieve the feedback from rejections to users' choice without increasing the simulation time. Moreover, results of this approach are highly interpretable, notably as a necessary cost of service to lower the demand to a certain level.

Several development points and perspectives are identified for each aspect. On the first one, a DMC model for MoD using parameters calibrated for the context of Île-de-France should be in future works. Fellow researchers working on this topic were identified and future collaboration is to be expected. Moreover, as already identified in Chapter 3 for future public transport lines, simulations should be assessed on the Île-de-France level in order to better assess the impact of MoD on users' mode choices. However, unlike the impact of new public transports which can be assessed with a downscaled population, MoD systems are better assessed with the full population. This presents a challenge as simulations on the Île-de-France level take 3 weeks per simulation, making the evaluation of various fleet sizes, service costs and operational strategies a challenge.

On the technical side, large efforts were put in addressing the computational and software related challenges to achieve the integration of an intermodal MoD system. On the one hand, an open-source framework like MATSim makes the integration of a new feature non-trivial. On the other hand, the impact of new developments is accordingly amplified since are be usable in various studies. Today, the technical infrastructure for performing realistic simulations of intermodal MoD systems is available and already being used in various researches in the Anthropolis Chair. For the most part, these studies are investigations and assessment of policy scenarios of MoD (e.g. sharing or non-sharing the bus infrastructure with MoD vehicles). These studies, beyond the scope of this thesis, have just started as they are dependent on external data which were provided only recently by project partners.

5 - Implementing a RL operated MoD system in MATSim

The highly dynamic nature of mobility systems and the fact that mobility environments and the demand for on-demand mobility systems can evolve in uncertain ways present a potential for the use of Reinforcement Learning (RL) algorithms in the operation of on-demand mobility systems.

As detailed in Chapter 2, techniques to operate a fleet of on-demand vehicles based on reinforcement learning algorithms have been studied and tested in agent-based simulation frameworks in recent works. The literature review performed in Section 2.4 shows the variety of approaches that have been employed and analyzes the differences between them in the algorithmic aspects, addressed use cases and evaluation methodologies. Many gaps and development pathways have been identified. Mainly, the literature around reinforcement learning for MoD lacks reproducibility of the approaches as most of the works are performed using non-open simulation tools on use cases that do not take into account the presence of other mobility systems and congestion.

In this chapter, we focus on the rebalancing task. We detail our work of implementing a reinforcement learning algorithm for on-demand vehicle rebalancing within the MATSim simulation framework. Given the complexity of the existing MATSim code base and more particularly the DRT module which implements MoD systems, this tasks involves various software and development challenges. These challenges are addressed in a way that offers a highly extendable architecture in which reinforcement learning-based algorithms can be implemented and tested in any MATSim simulation scenario involving MoD with minimum coupling. The details of this architecture are presented in Section 5.1

The capabilities of our architecture for reinforcement learning in MATSim are illustrated in a simulation use case. A decentralized Q-Learning for empty vehicle rebalancing is proposed in Section 5.2 and tested on a simple scenario in Section 5.3. This work is the subject of a conference paper submitted to The 13th International Conference on Ambient Systems, Networks and Technologies (ANT 2022) Chouaki et al. (2022)

5.1 . Implementing A RL rebalancing server for MATSim

MATSim is built in a modular manner that allowed it to be extended in various ways. The software architecture with which MATSim is made of bricks called modules. Each module implements a feature of the tool. For instance, the QSim module for the network simulation of vehicles, the transit module for the simulation of public transports or the DRT module for the simulation of MoD systems. The

```
<parameterset type="rebalancing">
        <param name="interval" value="1800"/>
        <param name="maxTimeBeforeIdle" value="900.0"/>
        <param name"minServiceTime" value="3600"/>
        <parameterset type="PlusOneRebalancingStrategy">
                <param name="RelocationCalculatorType"
                        value="FastHeuristic"/>
        </parameterset>
</parameterset>
```

Figure 5.1: XML code example for configuring the DRT module to use the MFAR rebalancing strategy

```
<parameterset type="rebalancing">
        <param name="interval" value="1800"/>
        <param name="maxTimeBeforeIdle" value="900.0"/>
        <param name"minServiceTime" value="3600"/>
        <parameterset type="minCostFlowRebalancingStrategy">
                <param name="demandEstimationPeriod" value="1800"/>
                <param name="rebalancingTargetCalculatorType"
                        value="EstimateDemand" />
                <param name="targetAlpha" value="0.5"/>
                <param name="targetBeta" value="0.0"/>
                <param name="zonalDemandEstimatorType"
                        value="PreviousIterationDemand" />
        </parameterset>
</parameterset>
```

Figure 5.2: XML code example for configuring the DRT module to use the MCF rebalancing strategy

```
<parameterset type="rebalancing">
        <param name="interval" value="1800"/>
        <param name="maxTimeBeforeIdle" value="900.0"/>
        <param name"minServiceTime" value="3600"/>
        <parameterset type="remoteRebalancingStrategyParams">
                <param name="address" value="tcp://localhost:5555"/>
                <parameterset type="SimpleQLearning">
                        <param name="alpha" value="0.01"/>
                        <param name="gamma" value="0.5"/>
                        <param name="epsilon" value="0.05"/>
                        <param name="discreteTimeIntervalLength"
                                value="1800"/>
                </parameterset>
        </parameterset>
</parameterset>
```

Figure 5.3: XML code example for configuring the DRT module to use the our Q-Learning rebalancing

basis of the software architecture is written in Java, thus new MATSim modules need to be written in the same programming language.

A MATSim module interacts directly with another module in two main ways: either by explicitly requiring a certain functionality made available by the other module or by listening to events fired (i.e. emitted) by it. In most cases, the first method allows to order the module to perform certain actions while the second is used to track what is happening in the simulation. Moreover, a functionality can be required and an event can be listened to without making an assumption on which module provides the functionality or fires the event. This allows modules to be swapped and replaced by others that can greatly differ in the inner working and keep MATSim functioning as long as the interfaces are the same.

The DRT module handles all behaviors related to MoD systems. This involves receiving trip requests, performing the vehicle assignment algorithm to match open requests with vehicles (if possible) and perform idle vehicle relocations by following a rebalancing algorithm. The DRT module itself is built in a modular manner. For instance, each rebalancing strategy is provided by a rebalancing module and the one to be used can be specified in the configuration of the DRT module. The rebalancing strategy to be used can be specified and configured in XML as part of the configuration of the DRT module. Figures 5.1 and 5.2 show an example of configuration for two of the implemented rebalancing algorithms. Consequently, to properly add a new rebalancing strategy into the DRT module, a new rebalancing module needs to be written and integrated.

The most straightforward approach for investigating RL-based rebalancing algorithms in MATSim would be to implement a new rebalancing module and use it within the DRT module. However and with our ambition of providing a frame-

work for benchmarking various RL algorithms for MoD in MATSim, the current architecture of the simulation framework and more particularly the DRT module is quite constraining for various reasons:

- The JAVA programming language is used for the MATSim framework. While it allowed a high modularity and portability, this language is far from being the go-to solution when it comes to implementing RL algorithms. Most of the research in the literature around RL, including the RL for MoD literature presented in Section 2.4, is performed using Python in which many libraries related to RL are available.

- Implementing a new MATSim module requires to be well familiar with the MATSim code base. This steep learning curve has been much experienced within this thesis project. Implementing a RL based algorithm for MoD in MATSim as a new rebalancing module then requires researchers and developers that are both experts in MATSim and familiar with RL. This greatly reduces the scope and interest of following this approach.

- Implementing a large number of RL-based rebalancing algorithms is not the goal of DRT module and would add unnecessary coupling and complexity to the overall code-base.

- Many MATSim simulations already require a large amount of computer memory and run on machines that tightly fit the hardware requirements. Adding an RL-based algorithm can render the overall simulation too resource consuming to be performed. Especially if deep-RL approaches that use deep neural networks are used, computer hardware requirements can include the presence of GPUs in addition to a large memory. With our experience in using computer clusters in Chapters 3 and 4, both are almost never present on a same computing node. Consequently, a solution restricted to using a single machine would not allow to perform large scale studies.

For these reasons, we chose not to implement our RL algorithms inside the DRT module itself but as an external tool following a client-server architecture where the external tool constitutes a *rebalancing server*. This server is coded in Python and built in an extensible manner that allows to quickly implement and test new RL algorithms. On the MATSim side, the DRT module is extended with a rebalancing module called *Remote Rebalancing Module*. This extension handles the communication with the server (building and sending rebalancing requests and interpreting responses) and the configuration of the algorithms on the server side. Communications between MATSim and the rebalancing server are performed following network TCP requests. This allows the rebalancing server and the simulation to run on separate, appropriately equipped, computers. The location of the server is known to MATSim through the configuration. This communication between

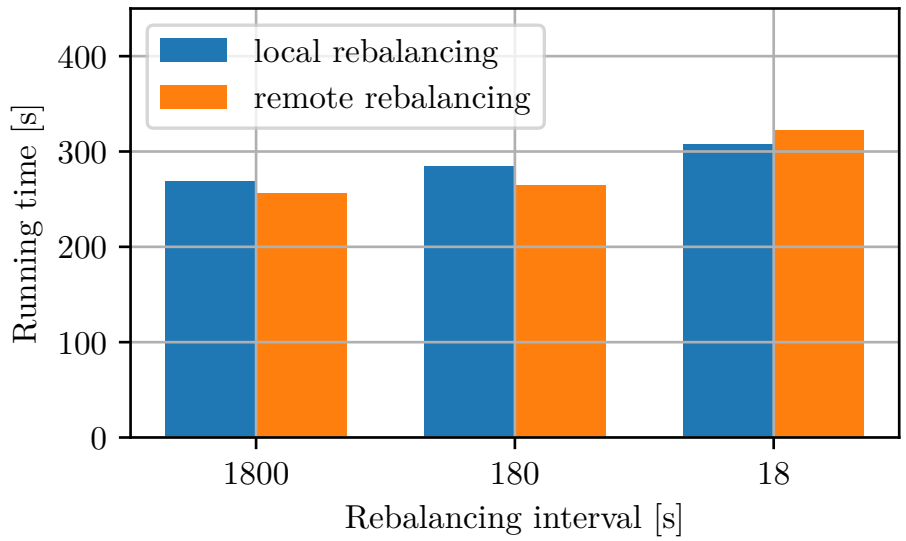

Figure 5.4: Communication overhead benchmark

MATSim and the rebalancing server is enabled by the ZeroMQ library Hintjens (2013) which is available in both Java and Python (alongside other languages).

Figure 5.5 details the communications between MATSim and the rebalancing server. At the start of a MATSim simulation, all required modules are installed using their respective configurations. During the installation of the DRT module, our extension contacts the rebalancing server to initialize the requested rebalancing algorithm with relevant parameters. The algorithm and the parameters can be easily specified in XML in the simulation's configuration file as illustrated in Figure 5.3. Later, at each rebalancing step, a request with relevant information for the rebalancing algorithm is built and sent to the server. The request contains sub-requests for each vehicle to rebalance, including all necessary information to make the decision. The latter performs the algorithm computation and replies with a relocation command to be carried on for each vehicle. Moreover, the rebalancing server is informed of each iteration's end to avoid linking the first encountered states of the next iteration with the last states encountered in the previous one.

Using an external rebalancing server means adding a communication overhead between the two pieces of software. We ran tests with a simple rebalancing strategy that exists in the DRT module and that we re-implemented (Send vehicles to their start link) while increasing the frequency of rebalancing requests (1800 seconds of simulated time then 180 then 18) to increase the number of requests. The results of this benchmark show that the overhead of using a rebalancing server can be neglected, as depicted in figure 5.4.

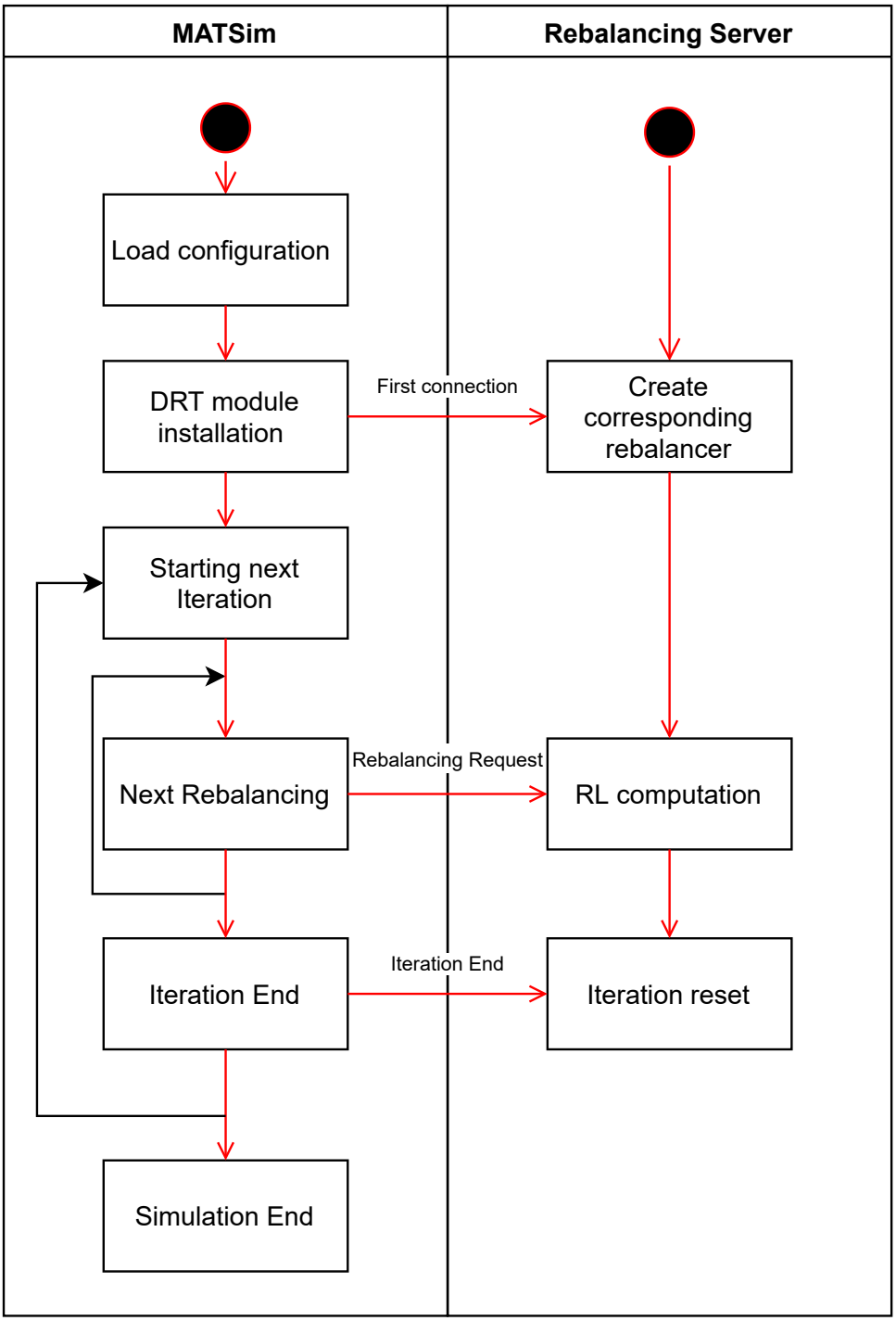

Figure 5.5: MATSim-Rebalancing server communication

5.2 . A first RL algorithm for MoD vehicle rebalancing in MATSim

Based on the presented infrastructure, we have implemented a Q-learning algorithm. Its main objective is that the vehicles of an on-demand fleet learn to relocate in the network to be able to serve forthcoming demand more effectively. Our approach is decentralized, meaning that each vehicle learns its own policy separately. The components of the learning agents' state in this algorithm are the vehicle location and the time of the day, which are both discretized. An agent's decision corresponds to choosing one of the discrete locations available. This means that a network divided in 500 zones and a day divided in 48 slots of 30 minutes each will produce a Q-table (state-action-expected reward) of 12 million entries. The size of Q-tables has to be kept in mind both for convergence time and for memory consumption. For the reward signal, at each rebalancing decision, the vehicle receives the number of passengers that have been transported since the last decision. The Q-values are then updated following the standard Q-learning rule:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{s+1}, a) - Q(S_t, A_t)]. \quad (5.1)$$

In this equation, the old value of a state-action pair is shifted towards the most recent reward observation while taking into account the long-term reward that can be obtained from the new state if the Q-values are used greedily afterwards.

All vehicles are checked periodically, and a rebalancing decision (and thus an update of its Q-table and action selection) for each empty vehicle is made after every rebalancing request. We use an ϵ-greedy policy for action selection, meaning that a vehicle selects the action with the best Q-value with probability $1 - \epsilon$ and a random action with probability ϵ.

The learning process spans across multiple iterations of the MATSim simulation, such that the same vehicle-time-place combination in different iterations corresponds to the same state entry in the Q-table. The update of the Q-table described in equation 5.1 is not performed for the first rebalancing action of each iteration, since the first state encountered at iteration $t + 1$ do not result from the last action performed at iteration t

5.3 . A simple monomodal test case

To test the algorithm that we describe above, we use a MATSim scenario for the city of Cottbus, Germany, that was first described in Grether et al. (2011) and adapted for mobility on-demand in the MATSim-MaaS example[1].

We have disabled MATSim agents' replanning behavior (shifting activity and trip times as well as choosing better modes) to ensure that the improvements ob-

[1] https://github.com/matsim-org/matsim-maas

served over multiple iterations come only from the vehicles learning better policies over time and not the users learning better departure times. Alongside the rebalancing algorithms that are tested, the vehicles are assigned to requests using an insertion algorithm that takes both maximum waiting times and maximum detour factors for shared rides into account. Consequently, requests can be rejected if no vehicle is close enough to the origin and the number of rejected requests has to be taken into account in assessing the performance of the rebalancing strategy.

We compare the performance of our algorithm with two other rebalancing algorithms that are implemented in MATSim's DRT module:

- Model-free adaptive relocation policy (MFAR) (Ruch et al., 2020): This algorithm considers that requests tend to appear in the same areas and relocates idle vehicles to the locations of previously matched requests without using any data on prior demand.
- MCF (MCF): This algorithm, described in Bischoff and Maciejewski (2020) computes target vehicle counts for each zone of the network based on the demand of the previous iteration, and then solves a transportation problem that optimally sends extra vehicles to zones where the target vehicle count is not met. This algorithm is greatly favored by our test scenario since MATSim agents' replanning is disabled and thus the demand is exactly the same through all iterations.

The Cottbus scenario in MATSim-MaaS contains a fleet of 200 vehicles by default. Both MFAR and MCF algorithms are able to satisfy all the requests with such a fleet. To be able to test our Q-learning rebalancing algorithm on a more constrained setting, we lower the size of the fleet (each time by 10 vehicles) until reaching the limit below which the algorithms show rejected requests. This threshold is 140 vehicles for both algorithms. Consequently, we test our algorithm with this fleet size.

Our algorithm is tested with various combinations of values for learning rate, discount factor and exploration probability, $(\alpha, \gamma, \epsilon) \in \{0.1, 0.5, 0.9\} \times \{0.1, 0.5, 0.9\} \times \{0.01, 0.05, 0.1\}$. This sensitivity analysis allows us to study how these parameters affect the performance of Q-learning algorithms for the rebalancing task.

Figures 5.6.A and 5.6.B compare the number of rejected requests and the average passengers' waiting times for our Q-learning algorithm with MCF and MFAR across the simulation's iterations. The MFAR rebalancing strategy's performance is stable across all iterations, while being able to satisfy the whole demand starting right from the beginning. In contrast, a first iteration is needed for MCF to learn the demand that will be exactly the same for the rest of the simulation. The algorithm satisfies all the requests with waiting times higher in average than MFAR.

Our Q-learning algorithm starts with a very low performance (especially in terms of rejected requests), but then rapidly learns to satisfy the majority of the

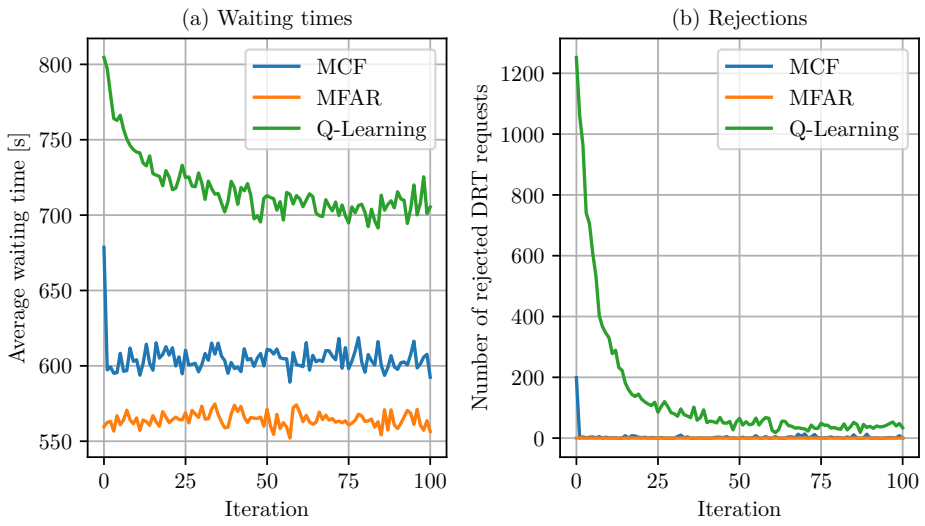

Figure 5.6: Comparison of between the implemented Q-learning algorithm and the MCF and MFAR algorithms. Figure 5.6.a shows the evolution of the average waiting time of performed DRT requests across the iterations of the simulation. Figure 5.6.b shows the evolution of the number of rejected DRT requests across the iterations of the simulation

requests with only one request remaining unsatisfied in the best case, but with waiting times on average higher than those of MCF and MFAR. These results show that an RL-based algorithm for rebalancing can effectively learn better operation policies over MATSim iterations, and the implemented architecture allows to easily run the same scenarios with various operation algorithms and compare them. However, more intelligent RL algorithms may be necessary to compete with the performance of the existing algorithms.

5.4 . Conclusion

In this work, we have designed and implemented a software architecture extending MATSim, and more specifically its DRT module with a remote rebalancing server, to allow rebalancing algorithms for shared on-demand vehicles to be implemented in Python. This is done with very low coupling with MATSim's code base and overhead due to the communication with the rebalancing server. This allows for more flexibility and will enable researchers to test different, more elaborate approaches on the various ready-to-use scenarios that are available in MATSim. Our first simple Q-learning algorithm shows interesting results that illustrate the potential of using such approaches in the MATSim framework, as well as the ability to be compared with other algorithms. However, we report that the chosen approach is not competitive with existing implemented relocation algorithms in MATSim with respect to waiting times. More advanced reinforcement learning approaches should, hence, be explored. Moreover, the test case used in this chapter is idealized as there is no uncertainty (this is where RL algorithms usually have an added value over classical methods). Cases with more dynamic demands and reflecting evolution on longer periods need to be considered.

Due to time constraints, particularly related to the necessary computation time, a full investigation of the use of RL approaches on the intermodal MoD test case developed in Chapter 4 could not be performed. In the results presented in Section 5.3, a large sensitivity analysis on the algorithm parameters is performed to identify the values that yield the best results. The same algorithm was tested with the same values in the intermodal MoD setting on the Paris-Saclay area but the results are not conclusive. A full in-depth study is necessary to assess the interest of this algorithm. This lies in the scope of future research.

In further work, the architecture described in this article could be extended to include vehicle assignment alongside rebalancing. Algorithms that consider electric vehicles (including the decisions of whether and where they should go to recharge) can be studied, and the operation of MoD systems with taking into account real time information about regular public transport present an interesting pathway of research.

6 - Conclusion

In this chapter, a summary of the contributions of this thesis is given and the perspectives for pursuing this work even further are presented.

Section 6.1 gives an overview on the scientific contributions and methodological contribution of this PhD projects. These consist of publications, presentations, participations to workshops, research collaborations and other academic activities. Section 6.2 emphasizes the technical contributions related to the simulation of MoD systems and their operation using RL-based algorithms in MATSim. Section 6.3 outlines the many perspectives that are envisioned to go further in various aspects addressed by this PhD work. Some of which have already started progressing in the scope of the Anthropolis Chair.

6.1 . Scientific contributions and methodological advances

This thesis started with a period of becoming familiar with the relevant concepts and software tools, especially the MATSim simulation framework, after which the rhythm of the thesis has been alternating between periods of technical developments and simulations, result analyses, paper writing and various communications. The related work presented Chapter 5 is based on the first conference publication that has been accepted during the course of this PhD. The work was presented at the 13th International Conference on Ambient Systems, Networks and Technologies (ANT 2022) Chouaki et al. (2022). To our knowledge, this conference paper is the first work to show how RL for MoD can be introduced and used effectively in MATSim. Moreover, it shows the lack of elaborate use cases with MoD where RL approaches can present an increased value.

Addressing this lack has been the goal of the following part of the thesis. Intermodality between MoD and public transport with realistic demand has been identified as a focus. Moreover, without being constrained by them, the interest of pairing the study of an intermodal MoD system with a prospective assessment of future planned rail-based systems perfectly aligns with the goals of the Anthropolis Chair whose project partners have been able to provide insights and data. After a first presentation of the work on this subject in the MATSim User Meeting of March 2022 at KU Leuven, developments continued and led to the submission of a conference paper to the 102^{nd} annual meeting of the Transportation Research Board (TRB 2023) (Chouaki et al., 2023) that has been presented in a poster format to fellow researchers interested in the same or related topics. A major selling point of our work consists in the reproducibility of our simulations as they rely on open data and open source tools. Moreover, our ability to find ways to study MoD systems in simulations with a non-scaled population sample allows to obtain directly interpretable results.

Later, the emphasis was put on better integration of a MoD system's performance in the mode choice model. A novel-approach for taking into account the rejection of trip requests to MoD systems in the discrete mode choice model responsible of traveler behavior is proposed and tested on our intermodal MoD setting with promising results. This represents an important step towards better comparison of operational strategies of MoD systems. In most cases, the main difference between operational strategies tested in MATSim lies in the resulting rejection rate. Our approach then allows to go one step further by comparing operational strategies (e.g. potentially RL-based rebalancing algorithms) through the level of demand that each solution attracts. The work on this aspect has been the subject of a paper that is accepted for presentation at the 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS) between in June 2023. Additionally, this work was presented at the workshop 'Open challenges in flexible mobility' that was held at Telecom Paris on February 24th 2023.

A part of the academic effort in this thesis project has been dedicated to a literature review on the use of RL based algorithms for operating MoD systems. This review, presented in Chapter 2, gives a comprehensive overview of the state of the art. A set of RL approaches for MoD are presented in detail, with a notation that is as unified as possible. Their characteristics in terms of algorithmic aspects, use cases and evaluation methodologies are compared. Moreover, the algorithms are located in a novel framework for sequential decision making to further show the trends of the literature and the under-explored areas. This work is materialized by a journal paper that is, at the time these lines were written, under review at the Transportation Research Part C: Emerging Technologies.

The work on the assessment of MoD systems is extended to include a socio-economic analysis in a collaboration with another PhD project, performed by Felix Carreyre at the LVMT laboratory and Vedecom. A methodology for cost-benefit analysis of a MoD service powered by agent-based simulations is proposed and tested on a use case of the area of Berlin. This work has been presented at the TRB 2023 conference Carreyre et al. (2023) and an extended version is under review at the Transportation Research Record with Felix Carreyre as the lead author.

Results of our simulation use case of the Paris-Saclay area with MoD developed in this thesis have been used in the scope of a research conducted at Telecom Paris by Severin Diepolder and Andrea Araldo. In this work, an approach for measuring the accessibility of MoD services is proposed and tested in a realistic setting. This study is the subject of a conference paper that has been submitted to the 11^{th} Symposium of the European Association for Research in Transportation (hEART 2023).

During this thesis project, our work had the chance to be communicated in various occasions, mainly events of the Anthropolis Chair. Our work on the future of mobility in the Paris-Saclay area and the impact of future rail-based lines has

been presented during the Anthropolis 2022 Colloquium held on September 16th 2022[1]. The thesis project was pitched during the ThesisDay, an event centered on PhD students organized at IRT SystemX, on June 24th 2022[2]. The overall work performed in this PhD project has been presented at the SMART seminar session of March 2023, organized jointly between 5 french laboratories working on mobility topics[3].

6.2 . Technical contributions

Alongside the academic contributions made in this thesis, various technical developments have been performed that extend the set of tools that are available for mobility studies in MATSim. References to the developed code are given throughout this section.

As detailed in Chapter 5, a software architecture for integrating MATSim with an external server for the MoD rebalancing task is proposed[4]. In such a server, RL-based algorithms can be implemented and easily tested in MATSim simulations. In the future, our ambition is to push this platform further to include various algorithms to be compared through a set of scenarios.

The Eqasim extension, allowing to use discrete choice models in MATSim, has been extensively used throughout our work. The intermodal MoD system as well as its integration in the discrete mode choice model with appropriate constraints have been integrated in this extension. Additionally, our linear control approach for integrating MoD rejection rates into the mode choice is integrated in the extension. Moreover, features allowing to log the steps of the mode choice, considered alternatives and computed utilities have been added to allow better tracking and troubleshooting [5].

In the scope of our collaboration between this PhD project and the one of Felix Carreyre, aiming at performing cost-benefit analyses of MoD systems using agent-based simulations, a MATSim module called 'MATSim-cba' has been proposed[6]. This module allows to generate simulation outputs containing information that are relevant for such analyses in a format that is directly usable by the tools that are used in this field. Highly configurable through the standard XML configuration of MATSim simulations, this extension makes it easier for non technical experts to perform socio-economic analyses of simulated scenarios.

6.3 . Perspectives

[1] https://www.youtube.com/watch?v=sGVD5-ZZ2OQ
[2] https://www.youtube.com/watch?v=RG_zKh4Eg4I
[3] https://www.lvmt.fr/evenements/seminaire-smart/
[4] https://github.com/tkchouaki/matsim-drt-rl
[5] https://github.com/tkchouaki/eqasim-java
[6] https://github.com/MATSimToCBA/matsim-cba

As the title of this thesis indicates, various aspects related to agent-based simulations of intermodal MoD systems operated by reinforcement learning are addressed in this work. All the advances made on all of these aspects offer new pathways of research along which they can be pushed even further.

Regarding agent-based simulations of mobility in general, we show in Chapter 3 how the impact of planned future public transport lines can be assessed using prospective simulations. A realistic demand, consisting in a synthetic population, is paired with the current offer on top of which the planned developments are added. While this work provides the first simulation assessment of future rail-based lines the Paris-Saclay area, the approaches can be extended both on the side of the synthetic population and the mobility offer. On the first, forecasts regarding population developments (in terms of number and geographical distribution) can be taken into account to build a prospective synthetic population. Moreover, to avoid assessing over one highly unlikely future, multiple prospective scenarios reflecting different development hypothesis can be considered in order to obtain more nuanced conclusions. This is already being addressed in the Anthropolis Chair in close interaction with the PhD project of Tjark Gall Vallet et al. (2022). On the second side, our integration of future rail-based systems in Chapter 3 is missing data regarding future bus lines. This was mitigated in 4 by introducing an intermodal MoD system. Nonetheless, we see a potential for exploring methods for automatic design of public transport lines using agent-based simulations. Such an approach can propose new lines that integrate well with the future rail-based offer which can then serve as inputs for planning stakeholders to better understand the demand and actually implement new lines with knowledge of the real-world territory.

The work on simulation of intermodal MoD services presented in Chapter 4 can also be extended in various directions. The current implementation of the intermodal MoD uses the closest stations from the origin and destination for switching between MoD and public transport. Whereas this simplification greatly reduces computation time, it can yield sub-optimal results. Approaches with better compromise between execution time and route quality should be investigated. We see the possibility here for using reinforcement learning approaches in order to learn the suitable decisions over time. As for the integration of MoD performance in the discrete mode choice model, the immediate next step after rejection rates will be to use realistic estimations of waiting and travel times rather than the maximum ones allowed by the constraints. A research currently performed at ETH Zurich aiming to achieve just that has been identified. This offers a great opportunity for a collaboration where an approach for realistically estimating waiting and travel times will be combined with ours for taking into account rejection rates. This collaboration is in progress and will lead in the near-future to the submission of a journal article where the approaches are tested on two use cases. Moreover, the work presented in Section 4.2 is now being used as a basis for a study that aims to

investigate the impact of cost of the MoD service. In collaboration with the PhD project of Felix Carreyre, various service dimensions and cost strategies are tested and socio-economically evaluated using a cost-benefit analysis taking into account the operator costs.

The work on Reinforcement Learning can be greatly extended in various ways. More RL approaches for empty vehicle rebalancing can be implemented in the software architecture proposed in Chapter 5. In future works, an emphasis will be put on approaches based on policy function approximations as they have been less explored in the literature. Moreover, deep reinforcement learning techniques ought to be investigated in intermodal settings. Using neural networks for state representation offers the ability to include a wide range of features and real-time information about the state of the public transport system can be integrated to achieve a better performance with an intermodal MoD. Cross-scenario testing of algorithms will also be an interesting contribution to this field of research.

Finally, all the aspects addressed in this thesis can be combined for more holistic studies. The technical developments performed in this project can now serve as off-the-shelf components that can be easily integrated together. One immediately possible application is the use of RL based approaches for the operation of intermodal MoD systems combined with future public transport and comparing with other operational strategies under same level of rejection rates ensured by our penalty-based approach.

Bibliography

Abar, S., Theodoropoulos, G.K., Lemarinier, P., O'Hare, G.M., 2017. Agent based modelling and simulation tools: A review of the state-of-art software. Computer Science Review 24, 13–33.

Al-Abbasi, A.O., Ghosh, A., Aggarwal, V., 2019. DeepPool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning 20, 4714–4727. URL: https://ieeexplore.ieee.org/document/8793143/, doi:10.1109/TITS.2019.2931830.

Balac, M., Hörl, S., Axhausen, K.W., 2020. Fleet Sizing for Pooled (Automated) Vehicle Fleets. Transportation Research Record: Journal of the Transportation Research Board doi:10.1177/0361198120927388.

Balmer, M., Cetin, N., Kai Nagel, Raney, B., 2004. Towards truly agent-based traffic and mobility simulations, in: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004., pp. 60–67.

Berrada, J., Mouhoubi, I., Christoforou, Z., 2020. Factors of successful implementation and diffusion of services based on autonomous vehicles: Users' acceptance and operators' profitability. Research in Transportation Economics 83, 100902. doi:10.1016/j.retrec.2020.100902.

Bischoff, J., Maciejewski, M., 2020. Proactive empty vehicle rebalancing for Demand Responsive Transport services. Procedia Computer Science 170, 739–744. doi:10.1016/j.procs.2020.03.162.

Bischoff, J., Maciejewski, M., Nagel, K., 2017. City-wide shared taxis: A simulation study in Berlin, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 275–280. doi:10.1109/ITSC.2017.8317926. iSSN: 2153-0017.

Burmeister, B., Haddadi, A., Matylis, G., 1997. Application of multi-agent systems in traffic and transportation. IEE Proceedings - Software Engineering 144, 51–60. doi:10.1049/ip-sen:19971023.

Bürstlein, J., López, D., Farooq, B., 2021. Exploring first-mile on-demand transit solutions for North American suburbia: A case study of Markham, Canada. Transportation Research Part A: Policy and Practice 153, 261–283. doi:10.1016/j.tra.2021.08.018.

Carreyre, F., Chouaki, T., Coulombel, N., Berrada, J., Bouillaut, L., Hörl, S., 2023. On-demand autonomous vehicles in berlin: A cost benefit analysis, in: 102nd Annual Meeting of the Transportation Research Board (TRB 2023), Washington D.C, United States.

Castagna, A., Guériau, M., Vizzari, G., Dusparic, I., . Demand-responsive rebalancing zone generation for reinforcement learning-based on-demand mobility Preprint, 1–16. URL: https://content.iospress.com/articles/ai-communications/aic201575, doi:10.3233/AIC-201575. publisher: IOS Press.

Chouaki, T., Hörl, S., Puchinger, J., 2022. Implementing reinforcement learning for on-demand vehicle rebalancing in MATSim. Procedia Computer Science 201, 134–141. doi:10.1016/j.procs.2022.03.020.

Chouaki, T., Hörl, S., Puchinger, J., 2023. Towards Reproducible Simulations of the Grand Paris Express and On-Demand Feeder Services, in: 102nd Annual Meeting of the Transportation Research Board (TRB 2023), Washington D.C, United States.

Cugurullo, F., Acheampong, R.A., Gueriau, M., Dusparic, I., 2020. The transition to autonomous cars, the redesign of cities and the future of urban sustainability. Urban Geography , 1–27URL: https://www.tandfonline.com/doi/full/10.1080/02723638.2020.1746096, doi:10.1080/02723638.2020.1746096.

Delling, D., Pajor, T., Werneck, R.F., 2015. Round-Based Public Transit Routing. Transportation Science 49, 591–604. doi:10.1287/trsc.2014.0534.

Enders, T., Harrison, J., Pavone, M., Schiffer, M., 2022. Hybrid multi-agent deep reinforcement learning for autonomous mobility on demand systems. arXiv e-prints , arXiv–2212.

Eshkevari, S.S., Tang, X., Qin, Z., Mei, J., Zhang, C., Meng, Q., Xu, J., 2022. Reinforcement Learning in the Wild: Scalable RL Dispatching Algorithm Deployed in Ridehailing Marketplace. Technical Report arXiv:2202.05118. arXiv. URL: http://arxiv.org/abs/2202.05118, doi:10.48550/arXiv.2202.05118. arXiv:2202.05118 [cs] type: article.

Fluri, C., Ruch, C., Zilly, J., Hakenberg, J., Frazzoli, E., 2019. Learning to operate a fleet of cars, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE. pp. 2292–2298. URL: https://ieeexplore.ieee.org/document/8917533/, doi:10.1109/ITSC.2019.8917533.

Gammelli, D., Yang, K., Harrison, J., Rodrigues, F., Pereira, F.C., Pavone, M., 2021. Graph neural network reinforcement learning for autonomous mobility-

on-demand systems, in: 2021 60th IEEE Conference on Decision and Control (CDC), IEEE. pp. 2996–3003.

Geist, M., Pietquin, O., 2013. Algorithmic Survey of Parametric Value Function Approximation. IEEE Transactions on Neural Networks and Learning Systems 24, 845–867. doi:10.1109/TNNLS.2013.2247418. conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Golpayegani, F., Guériau, M., Laharotte, P.A., Ghanadbashi, S., Guo, J., Geraghty, J., Wang, S., 2022. Intelligent Shared Mobility Systems: A Survey on Whole System Design Requirements, Challenges and Future Direction. IEEE Access 10, 35302–35320. doi:10.1109/ACCESS.2022.3162848.

Grether, D., Bischoff, J., Nagel, K., 2011. Traffic-actuated Signal Control: Simulation of the User Benefits in a Big Event Real-World Scenario .

Gueriau, M., Cugurullo, F., Acheampong, R.A., Dusparic, I., 2020. Shared autonomous mobility on demand: A learning-based approach and its performance in the presence of traffic congestion 12, 208–218. doi:10.1109/MITS.2020.3014417. conference Name: IEEE Intelligent Transportation Systems Magazine.

Gurumurthy, K.M., Kockelman, K.M., Loeb, B.J., 2019. Sharing vehicles and sharing rides in real-time: Opportunities for self-driving fleets, in: Advances in Transport Policy and Planning. Elsevier. volume 4, pp. 59–85. doi:10.1016/bs.atpp.2019.09.001.

Haliem, M., Aggarwal, V., Bhargava, B., 2022. AdaPool: A Diurnal-Adaptive Fleet Management Framework Using Model-Free Deep Reinforcement Learning and Change Point Detection. IEEE Transactions on Intelligent Transportation Systems 23, 2471–2481. doi:10.1109/TITS.2021.3109611. conference Name: IEEE Transactions on Intelligent Transportation Systems.

Hamadneh, J., Esztergar-Kiss, D., 2019. Impacts of Shared Autonomous Vehicles on the Travelers' Mobility, in: 2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), IEEE, Cracow, Poland. doi:10.1109/MTITS.2019.8883392.

Hancock, P.A., Nourbakhsh, I., Stewart, J., 2019. On the future of transportation in an era of automated and autonomous vehicles. Proceedings of the National Academy of Sciences 116, 7684–7691. doi:10.1073/pnas.1805770115.

Harper, C.D., Hendrickson, C.T., Samaras, C., 2018. Exploring the Economic, Environmental, and Travel Implications of Changes in Parking Choices due to Driverless Vehicles: An Agent-Based Simulation Approach. Journal of Urban Planning and Development 144. doi:10.1061/(ASCE)UP.1943-5444.0000488.

Heilig, M., Hilgert, T., Mallig, N., Kagerbauer, M., Vortisch, P., 2017. Potentials of Autonomous Vehicles in a Changing Private Transportation System – a Case Study in the Stuttgart Region. Transportation Research Procedia 26, 13–21. doi:10.1016/j.trpro.2017.07.004.

Hintjens, P., 2013. ZeroMQ: messaging for many applications.

Horl, S., Balac, M., Axhausen, K.W., 2019. Dynamic demand estimation for an AMoD system in Paris, in: 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE, Paris, France. pp. 260–266. doi:10.1109/IVS.2019.8814051.

Hörl, S., Balać, M., Axhausen, K.W., 2019. Pairing discrete mode choice models and agent-based transport simulation with MATSim, in: 2019 TRB Annual Meeting Online, Transportation Research Board. p. 19. doi:10.3929/ethz-b-000303667.

Hörl, S., Becker, F., Axhausen, K.W., 2021. Simulation of price, customer behaviour and system impact for a cost-covering automated taxi system in Zurich. Transportation Research Part C: Emerging Technologies 123, 102974. doi:10.1016/j.trc.2021.102974.

Horni, A., Nagel, K., Axhausen, K., 2016. The Multi-Agent Transport Simulation MATSim. doi:10.5334/baw.

Hörl, S., 2020. Dynamic Demand Simulation for Automated Mobility on Demand. Doctoral Thesis. ETH Zurich. URL: https://www.research-collection.ethz.ch/handle/20.500.11850/419837, doi:10.3929/ethz-b-000419837. accepted: 2020-06-12T12:28:41Z.

Hörl, S., Balac, M., 2021a. Introducing the eqasim pipeline: From raw data to agent-based transport simulation. Procedia Computer Science 184, 712–719. URL: https://www.sciencedirect.com/science/article/pii/S1877050921007274, doi:10.1016/j.procs.2021.03.089.

Hörl, S., Balac, M., 2021b. Synthetic population and travel demand for Paris and Île-de-France based on open and publicly available data. Transportation Research Part C: Emerging Technologies 130, 103291. URL: https://www.sciencedirect.com/science/article/pii/S0968090X21003016, doi:10.1016/j.trc.2021.103291.

Hörl, S., Balac, M., Axhausen, K.W., 2019. Dynamic demand estimation for an AMoD system in Paris, in: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 260–266. doi:10.1109/IVS.2019.8814051. iSSN: 2642-7214.

Hörl, S., Ruch, C., Becker, F., Frazzoli, E., Axhausen, K.W., . Fleet control algorithms for automated mobility: A simulation assessment for

zurich. URL: https://www.research-collection.ethz.ch/handle/20.500.11850/175260, doi:10.3929/ethz-b-000175260. accepted: 2019-03-28T11:40:17Z Publication Title: Arbeitsberichte Verkehrs- und Raumplanung Volume: 1270.

Johnson, M.A., Moradi, M.H. (Eds.), 2005. PID Control. Springer-Verlag, London. URL: http://link.springer.com/10.1007/1-84628-148-2, doi:10.1007/1-84628-148-2.

Kaddoura, I., Schlenther, T., 2021. The impact of trip density on the fleet size and pooling rate of ride-hailing services: A simulation study. Procedia Computer Science 184, 674–679. URL: https://www.sciencedirect.com/science/article/pii/S1877050921007213, doi:10.1016/j.procs.2021.03.084.

Kagho, G.O., Meli, J., Walser, D., Balac, M., 2022. Effects of population sampling on agent-based transport simulation of on-demand services. Procedia Computer Science 201, 305–312. URL: https://www.sciencedirect.com/science/article/pii/S1877050922004549, doi:10.1016/j.procs.2022.03.041.

Kemker, R., McClure, M., Abitino, A., Hayes, T., Kanan, C., 2018. Measuring catastrophic forgetting in neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence.

Kim, J., Kim, K., 2021. Optimizing Large-Scale Fleet Management on a Road Network using Multi-Agent Deep Reinforcement Learning with Graph Neural Network, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), pp. 990–995. doi:10.1109/ITSC48978.2021.9565029.

KJ, P., Singh, N., Dayama, P., Agarwal, A., Pandit, V., 2022. Change point detection for compositional multivariate data. Applied Intelligence 52, 1930–1955.

Konda, V., Tsitsiklis, J., 1999. Actor-Critic Algorithms, in: Advances in Neural Information Processing Systems, MIT Press. URL: https://proceedings.neurips.cc/paper/1999/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

Liang, Y., Ding, Z., Ding, T., Lee, W.J., 2021. Mobility-Aware Charging Scheduling for Shared On-Demand Electric Vehicle Fleet Using Deep Reinforcement Learning. IEEE Transactions on Smart Grid 12, 1380–1393. doi:10.1109/TSG.2020.3025082. conference Name: IEEE Transactions on Smart Grid.

Littman, M.L., 1994. Markov games as a framework for multi-agent reinforcement learning, in: In Proceedings of the Eleventh International Conference on Machine Learning, Morgan Kaufmann. pp. 157–163.

Lloyd, S., 1982. Least squares quantization in pcm. IEEE transactions on information theory 28, 129–137.

Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., 2018. Microscopic traffic simulation using sumo, in: 2018 21st international conference on intelligent transportation systems (ITSC), IEEE. pp. 2575–2582.

Mao, C., Liu, Y., Shen, Z.J.M., 2020. Dispatch of autonomous vehicles for taxi services: A deep reinforcement learning approach. Transportation Research Part C: Emerging Technologies 115, 102626. URL: https://www.sciencedirect.com/science/article/pii/S0968090X19312227, doi:10.1016/j.trc.2020.102626.

Martinho, A., Herber, N., Kroesen, M., Chorus, C., 2021. Ethical issues in focus by the autonomous vehicles industry. Transport reviews 41, 556–577.

Miao, F., Han, S., Lin, S., Stankovic, J.A., Zhang, D., Munir, S., Huang, H., He, T., Pappas, G.J., 2016. Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach. IEEE Transactions on Automation Science and Engineering 13, 463–478.

Molenbruch, Y., Braekers, K., Caris, A., 2017. Typology and literature review for dial-a-ride problems. Annals of Operations Research 259, 295–325. doi:10.1007/s10479-017-2525-0.

Narayanan, S., Chaniotakis, E., Antoniou, C., 2020. Shared autonomous vehicle services: A comprehensive review. Transportation Research Part C: Emerging Technologies 111, 255–293. doi:10.1016/j.trc.2019.12.008.

Pavone, M., Smith, S.L., Frazzoli, E., Rus, D., 2012. Robotic load balancing for mobility-on-demand systems. The International Journal of Robotics Research 31, 839–854.

Pernestål, A., Kristoffersson, I., 2019. Effects of driverless vehicles. European Journal of Transport and Infrastructure Research 19. doi:10.18757/EJTIR.2019.19.1.4079.

Pinto, H.K., Hyland, M.F., Mahmassani, H.S., Ömer Verbas, I., 2020. Joint design of multimodal transit networks and shared autonomous mobility fleets. Transportation Research Part C: Emerging Technologies 113, 2–20. URL: https://www.sciencedirect.com/science/article/pii/S0968090X18317728, doi:https://doi.org/10.1016/j.trc.2019.06.010. 23rd International Symposium on Transportation and Traffic Theory (ISTTT 23).

Powell, W.B., 2019. A unified framework for stochastic optimization 275, 795–821. URL: http://www.sciencedirect.com/science/article/pii/S0377221718306192, doi:10.1016/j.ejor.2018.07.014.

Qin, G., Luo, Q., Yin, Y., Sun, J., Ye, J., 2021. Optimizing matching time intervals for ride-hailing services using reinforcement learning. Transportation Research Part C: Emerging Technologies 129, 103239. URL: https://www.sciencedirect.com/science/article/pii/S0968090X21002527, doi:https://doi.org/10.1016/j.trc.2021.103239.

Ruch, C., Gächter, J., Hakenberg, J., Frazzoli, E., 2020. The +1 Method: Model-Free Adaptive Repositioning Policies for Robotic Multi-Agent Systems. IEEE Transactions on Network Science and Engineering 7, 3171–3184. doi:10.1109/TNSE.2020.3017526. conference Name: IEEE Transactions on Network Science and Engineering.

Ruch, C., Hörl, S., Frazzoli, E., 2018. Amodeus, a simulation-based testbed for autonomous mobility-on-demand systems, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE. pp. 3639–3644.

Saharan, S., Bawa, S., Kumar, N., 2020. Dynamic pricing techniques for Intelligent Transportation System in smart cities: A systematic review. Computer Communications 150, 603–625. URL: https://www.sciencedirect.com/science/article/pii/S0140366419310990, doi:10.1016/j.comcom.2019.12.003.

Saidallah, M., Fergougui, A.E., Elalaoui, A.E., 2016. A Comparative Study of Urban Road Traffic Simulators. doi:10.1051/matecconf/20168105002.

Schaller, B., 2018. The New Automobility: Lyft, Uber and the Future of American Cities .

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 .

Sieber, L., Ruch, C., Hörl, S., Axhausen, K.W., Frazzoli, E., 2020. Improved public transportation in rural areas with self-driving cars: A study on the operation of swiss train lines. Transportation research part A: policy and practice 134, 35–51.

Storsæter, A.D., Pitera, K., McCormack, E.D., 2021. The automated driver as a new road user. Transport reviews 41, 533–555.

Sutton, R.S., Barto, A.G., 1998. Reinforcement Learning: An Introduction. The MIT Press. URL: https://muse.jhu.edu/book/60836.

Tang, X., Qin, Z.T., Zhang, F., Wang, Z., Xu, Z., Ma, Y., Zhu, H., Ye, J., 2019. A Deep Value-network Based Approach for Multi-Driver Order Dispatching, in:

Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, New York, NY, USA. pp. 1780–1790. URL: https://doi.org/10.1145/3292500.3330724, doi:10.1145/3292500.3330724.

Tang, X., Zhang, F., Qin, Z., Wang, Y., Shi, D., Song, B., Tong, Y., Zhu, H., Ye, J., 2021. 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, Association for Computing Machinery, New York, NY, USA. pp. 3605–3615. URL: https://doi.org/10.1145/3447548.3467096, doi:10.1145/3447548.3467096.

Tlc, N., 2020. Nyc taxi and limousine commission (tlc) trip record data .

Train, K., 2009. Discrete choice methods with simulation. 2nd ed ed., Cambridge University Press, Cambridge ; New York. OCLC: ocn349248337.

Uhrmacher, A.M., Weyns, D., 2009. Multi-Agent Systems: Simulation and Applications. CRC Press.

Vallet, F., Hörl, S., Gall, T., 2022. Matching Synthetic Populations with Personas: A Test Application for Urban Mobility. Proceedings of the Design Society 2, 1795–1804. doi:10.1017/pds.2022.182.

Vosooghi, R., Kamel, J., Puchinger, J., Leblond, V., Jankovic, M., 2019a. Robo-Taxi service fleet sizing: Assessing the impact of user trust and willingness-to-use. Transportation doi:10.1007/s11116-019-10013-x.

Vosooghi, R., Puchinger, J., Jankovic, M., Vouillon, A., 2019b. Shared autonomous vehicle simulation and service design. Transportation Research Part C: Emerging Technologies 107, 15–33. doi:10.1016/j.trc.2019.08.006.

Wen, J., Chen, Y.X., Nassir, N., Zhao, J., 2018. Transit-oriented autonomous vehicle operation with integrated demand-supply interaction. Transportation Research Part C: Emerging Technologies 97, 216–234. URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X18300378, doi:10.1016/j.trc.2018.10.018.

Wen, J., Zhao, J., Jaillet, P., 2017. Rebalancing shared mobility-on-demand systems: A reinforcement learning approach, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 220–225. doi:10.1109/ITSC.2017.8317908. iSSN: 2153-0017.

Williams, E., Das, V., Fisher, A., 2020. Assessing the Sustainability Implications of Autonomous Vehicles: Recommendations for Research Community Practice. Sustainability 12. doi:10.3390/su12051902.

Williams, R.J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning 8, 229–256. URL: https://doi.org/10.1007/BF00992696, doi:10.1007/BF00992696.

Yoshida, N., Noda, I., Sugawara, T., 2020. Multi-agent service area adaptation for ride-sharing using deep reinforcement learning, in: International Conference on Practical Applications of Agents and Multi-Agent Systems, Springer. pp. 363–375.

Yoshida, N., Noda, I., Sugawara, T., 2021. Distributed service area control for ride sharing by using multi-agent deep reinforcement learning, pp. 101–112. URL: https://www.scitepress.org/Link.aspx?doi=10.5220/0010310901010112.

Zardini, G., Lanzetti, N., Pavone, M., Frazzoli, E., 2021. Analysis and Control of Autonomous Mobility-on-Demand Systems: A Review.