FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Reinforcement Learning for Optimizing the Definition of Logistics Distribution Sectors

Tomás Freitas Rodrigues

Mestrado em Engenharia e Ciência de Dados

Supervisor: Tânia Fontes, PhD

Second Supervisor: Zafeiris Kokkinogenis, PhD

October 31, 2024

Reinforcement Learning for Optimizing the Definition of Logistics Distribution Sectors

Tomás Freitas Rodrigues

Mestrado em Engenharia e Ciência de Dados

October 31, 2024

Abstract

This study addresses the complex challenges of last-mile delivery in e-commerce logistics, where high demand for rapid deliveries exacerbates issues such as urban congestion, variable weather, and driver availability. It is investigated the application of Reinforcement Learning (RL) to optimize delivery assignments, proposing an RL agent designed to create and adapt "areas of interest" (AOIs) in response to daily fluctuations in delivery demand. Specifically, we explore the Advantage Actor-Critic (A2C) and Proximal Policy Optimization (PPO) algorithms to assign deliveries to drivers dynamically, ensuring balanced workloads, stable delivery regions, and operational efficiency.

An extensive exploratory data analysis was conducted to model the environment. Some datasets were derived from this analysis and used to model and feed the agent with geographical and logistical information. Clustering methods and boxplots were employed to draw conclusions about the data and drive decisions for modeling. Additionally, a custom reinforcement learning environment was created from scratch, where mathematical reward functions were developed to guide the RL agent's decision-making, aiming to balance workload distribution and minimize travel distance.

Extensive computational experiments compare the performance of A2C and PPO in maintaining flexibility and efficiency in variable environments. Although the model was ultimately unable to fully learn the optimal policies, this project demonstrated effective methods and scenarios for addressing overfitting, specifically in urban transportation and logistics in the last mile. The study also considers ethical and practical factors of RL-driven decision-making in logistics, focusing on scalability, safety, and reliability.

Keywords: Reinforcement Learning, Last-Mile, Logistic-Sectors, Optimization, Assignment

Resumo

Este estudo aborda os desafios complexos da entrega de última milha na logística do comércio eletrónico, onde a elevada procura por entregas rápidas agrava problemas como o congestionamento urbano, condições meteorológicas variáveis e disponibilidade de motoristas. Investigamos a aplicação do Reinforcement Learning (RL) para otimizar as atribuições de entrega, propondo um agente RL concebido para criar e adaptar "áreas de interesse" (AOIs) em resposta às flutuações diárias da procura de entrega. Especificamente, explorámos os algoritmos Advantage Actor-Critic (A2C) e Proximal Policy Optimization (PPO) para atribuir entregas aos motoristas de forma dinâmica, garantindo cargas de trabalho equilibradas, regiões de entrega estáveis e eficiência operacional.

Uma extensa análise exploratória de dados foi conduzida para modelar o ambiente. Alguns conjuntos de dados foram derivados desta análise e utilizados para modelar e alimentar o agente com informação geográfica e logística. Foram empregues métodos de clustering e boxplots para tirar conclusões sobre os dados e informar decisões para a modelação. Além disso, foi criado de raiz um ambiente de aprendizagem por reforço personalizado, onde foram desenvolvidas funções matemáticas de recompensa para orientar a tomada de decisão do agente RL, visando equilibrar a distribuição da carga de trabalho e minimizar a distância percorrida.

Extensas experiências computacionais comparam o desempenho do A2C e do PPO na manutenção da flexibilidade e eficiência em ambientes variáveis. Embora o modelo tenha sido incapaz de aprender completamente a policy deal, este projecto demonstrou métodos e cenários eficazes para abordar o overfitting, especificamente no transporte urbano e na logística de última milha. O estudo considera ainda fatores éticos e práticos da tomada de decisão em logística orientada pela RL, com foco na escalabilidade, segurança e fiabilidade.

Keywords: Reinforcement Learning, Last-Mile, Logistic-Sectors, Optimization, Assignment

Acknowledgements

My education spans over 15 years, during which I have developed strong logical and critical thinking skills. This progress is largely due to the significant investment my parents made in my education.

Therefore, my deepest gratitude goes to them for always believing in my academic potential and understanding that a better education leads to more prepared and capable individuals. Their advice, especially my mom's, brought me peace during moments of uncertainty and doubt. The wisdom in her words did not make the journey easy, but it certainly made it less difficult.

To my supervisor, Tânia Fontes, I extend my gratitude for instilling professionalism in me throughout our collaboration as well as clear guidance regarding the scope of the project. I also want to express appreciation to my co-supervisor, Zaferies, for his guidance and support.

I want to extend my gratitude to all my colleagues in the master's degree program. Our shared journey over the past two years has been incredibly enriching, filled with learning and fond memories. I have learned a great deal from each of you, and I cherish the diverse perspectives and experiences we have exchanged.

Indeed, learning from each other is as crucial as learning from books, and the blend of backgrounds and cultures among us has undoubtedly shaped me into a better student and person. Thank you all for the camaraderie and growth we have experienced together.

Lastly, I would like to pay tribute to the individuals who, though no longer with us, played a significant role in shaping my journey and instilling important values in me.

Tomás Rodrigues

"Estudar com dúvida,
realizar com fé."

Unknown

Contents

List of Figures

List of Tables

Abbreviations

AGD	Average Grab Duration
AMP	Área Metropolitana do Porto
AMD	Average Moving Distance
AWT	Average Waiting Time
ARTG	Average Refresh Times between Grabs
AOI	Area of Interest
DRL	Deep Reinforcement Learning
DTV	Distance Travelled per Vehicle
DQN	Deep Qeue Networks
DTE	Delivery time efficiency
EDA	Exploratory Data Analysis
EV	Electric Vehicles
ETA	Estimated Time Arrival
FCSs	Fast Charging stations
FEUP	Faculdade de Engenharia da Universidade do Porto
GIS	Geographic Information Systems
KPI's	Key Performance Indicators
MOD	Mobility On Demand
NCT	Number of Confidently assigned Tasks
NFT	Number of Fully assigned Tasks
OFD	Online Food Delivery
PMA	Porto Metropolitan Area
PSR	Percentage of served requests
PCDT	Percent of Completed Delivery Tasks
PCPT	Percent of Completed Pick-up Tasks
RL	Reinforcement Learning
TCR	Task Completion Rate
UAV	Unmanned aerial vehicle
USV	Unmanned surface vehicle

Chapter 1

Introduction

This introductory chapter 1 provides the foundational context and motivation for the research, focusing on the complexities and challenges of last mile delivery in the e-commerce sector. It begins by outlining the key factors that exacerbate these challenges, such as urban traffic congestion, unpredictable weather conditions, and environmental impacts like deteriorating air quality, which affect both consumers and distributors 1.1. The chapter then highlights the increasing demand for same-day and next-day deliveries, driven by the rapid growth of e-commerce, emphasizing the pressing need for innovative, adaptive solutions to address these issues.

Following this, the chapter presents the specific research objectives 1.3, emphasizing the potential of Reinforcement Learning (RL) to tackle the unique challenges of last mile delivery. The discussion includes the formulation of primary research questions and underscores the significance of developing an RL agent capable of maintaining an adaptive Area of Interest (AOI) for couriers. The ultimate goal is to bridge the gap between theoretical advancements in machine learning and practical applications in logistics, setting the stage for the detailed exploration in the subsequent chapters.

1.1 Context

The last mile represents the last stage of the supply chain from a distribution center to the end user and according to Samet (2023) it is a challenging and expensive phase of the entire delivery process. Factors such as traffic congestion, unpredictable weather conditions, and the increasing preference for same-day or next-day deliveries have intensified the complexities associated with last mile logistics.

The rise of e-commerce has intensified last mile complexities, particularly impacting urban air quality and traffic congestion. The increased number of delivery vehicles leads to higher emissions of pollutants, worsening air quality and potentially harming public health, especially in densely populated areas. Moreover, traffic congestion caused by these vehicles results in longer commute times for consumers and increased operational inefficiencies for distributors. This congestion can

also escalate delivery costs, affect customer satisfaction, and contribute to the deterioration of urban infrastructure. According to AMBIENTE (2024), the transportation sector was responsible for most of the CO2 emissions, approximately 17,062 kt of CO2eq., in 2022, representing a reduction of only 14.5% compared to 2005, while the target for 2030 is a 40% reduction.

In recent years, the rapid growth of e-commerce has revolutionized the retail industry, leading to an extraordinary surge in demand for efficient and timely last mile delivery solutions. In Portugal, and according to INE Instituto Nacional de Estatística (2021), from 2010 to 2021, the purchases made online, in the past 3 months prior to when the people where interviewed, went from 9.5% to 40.4%, as it can be see in 1.1.

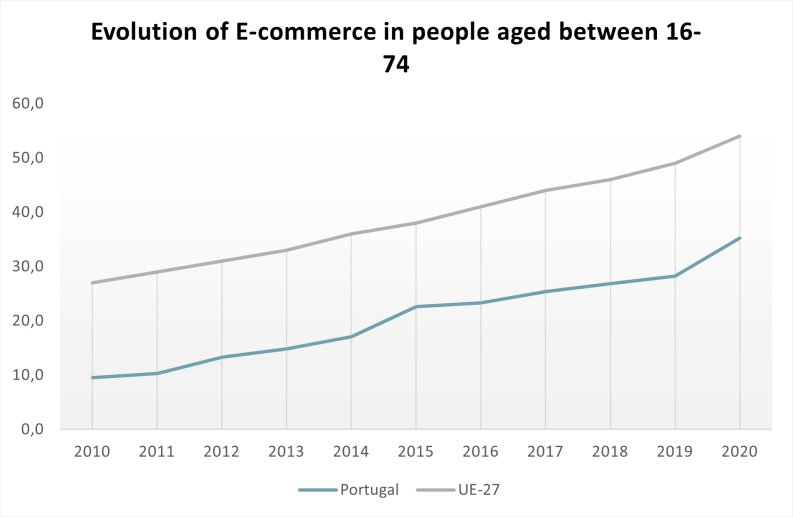

Figure 1.1: Evolution of e-commerce in Portuguese people aged between 16-74 (Source: Instituto Nacional de Estatística (2021)).

1.2 Motivation

Reinforcement Learning applied to last-mile delivery solutions is a relatively recent development in the research literature. Historically, heuristics such as Hill Climbing, Tabu Search, and Bee Colony Optimization have been more commonly employed. A simple query search on Scopus conducted on February, 2024, using the terms *('Last-Mile')* and *('Heuristics')* returns 354 papers. In contrast, the search query *('Last-Mile')* and *('Reinforcement-Learning')* yields only 68 articles. This discrepancy suggests that the application of Reinforcement Learning to last-mile delivery remains underexplored and presents significant opportunities for further investigation.

Since Reinforcement Learning is a machine learning field that has grown a lot in the last years, along with the fact that there are only a few papers in this problem domain and the fact that there is a need for innovative and adaptive approaches that can address the unique challenges posed by this critical phase of the supply chain. Reinforcement Learning, a subfield of machine learning,

emerges as a promising solution capable of learning optimal strategies through interaction with the environment.

An efficient last mile delivery solution can bring multiple benefits for a company, its employees and even the customers. Firstly, the employees by knowing their area of distribution better become more productive, and are capable of delivering the orders faster. By providing a good service, the customer satisfaction increases leading to increased sales. Lastly, all the other points converge to an increase in the companies profits as well as its reputation.

To sum up, by addressing the last mile delivery problem through the lens of Reinforcement Learning, this research attempts to bridge the gap between theoretical advancements in machine learning and the practical challenges faced by the logistics industry, ultimately paving the way for more effective and resilient last mile delivery systems.

1.3 Objectives

The central focus of this dissertation is on the ability of our Reinforcement Learning Agent to adapt to an uneven distribution of deliveries, while also maintaining a flexible Area of Interest (AOI). Ideally, a courier should stay within a consistent AOI to become more familiar with that area. However, the AOI should be adjustable, expanding or contracting as needed based on daily variations in delivery demand. This adaptability ensures that the agent can respond efficiently to changes, providing better service in areas with fluctuating delivery needs.

The research aims to explore some algorithms compatible with the action and observation space to perform comprehensive computational experiments and enable a fair comparison of their performances. Furthermore, visualizing the decision-making process through graphical representations is identified as a crucial aspect of this work.

An essential contribution of this study is the investigation of overfitting and its mitigation if verified. Overfitting, a well-documented concern in both deep learning and machine learning models, will be addressed by comparing performances on training and validation datasets, as well as by assessing the reinforcement learning agent's performance in diverse environments. Moreover, the implications of deploying a reinforcement learning agent will be thoroughly examined. This exploration will encompass ensuring the agent's reliability in real-world deployment scenarios and assessing the ethical, practical, and other implications associated with its deployment.

The main research questions of this work are:

- *RQ1* Is it possible to maintain a consistent Area of Interest (AOI) for couriers that adjust to daily changes in order distributions?
- *RQ2* Is the reinforcement learning agent prone to overfitting?
- *RQ3* Can the reinforcement learning agent be deployed in a near real-world scenario and what would be the implications of doing so?

1.4 Structure

This master dissertation follows the following structure and organization:

- **Chapter 1: Introduction**
  This chapter provides a comprehensive overview of the dissertation's theme, focusing on e-commerce and the Last-mile problem. It discusses the significance of the topic within the broader context, emphasizing its relevance and potential impact. The chapter also outlines the specific research questions addressed in this study and summarizes the proposed contributions.

- **Chapter 2: Background**
  This chapter introduces the foundational concepts of reinforcement learning, detailing key terminology and principles. It presents an in-depth overview of commonly used deep reinforcement learning algorithms, discussing their theoretical underpinnings and practical applications.

- **Chapter 3: Systematic Literature Review**
  In this chapter, a thorough review of the relevant literature is conducted. Key papers in the field are identified, and recent advancements are examined. The chapter organizes these studies to facilitate clarity and comparison, analyzing different approaches to modeling reinforcement learning environments. Methodologies and outcomes proposed by various authors are synthesized using tables for efficient comparison.

- **Chapter 4: Methodology**
  This chapter outlines the research methodology adopted to address the research problem. It details the exploratory data analysis (EDA) conducted to gain insights into the data and problem space. The chapter also describes the modeling process using Python, employing the most current and relevant libraries.

- **Chapter 5: Results**
  This chapter presents the results obtained from the models, comparing their performance based on various key performance indicators (KPIs). It includes an analysis of the model's rewards, and discusses issues related to overfitting and generalization of the agent.

- **Chapter 6: Conclusion**
  The concluding chapter summarizes the dissertation's findings, highlighting the key results and addressing how the research questions were approached. It reflects on the limitations of the study and suggests areas for future research. The chapter also discusses the potential for further refinement and evolution of the work, providing a clear linkage between the research questions and the results obtained.

Chapter 2

BackGround

This chapter introduces RL, covering its fundamental concepts, important terminology, and notable algorithms in deep reinforcement learning. The aim is to provide a comprehensive overview of the field to familiarize the reader with key terms and nomenclature used throughout this work.

2.1 Introduction

Reinforcement Learning (RL) is a subfield of machine learning concerned with how agents should take actions in an environment to maximize cumulative rewards. Unlike supervised learning, where a model is trained on a fixed dataset with labeled examples, RL involves learning from the consequences of actions, allowing agents to adapt their behavior based on the environment's feedback. This approach is particularly powerful in scenarios where the best action is not immediately clear and must be discovered through trial and error.

RL is inspired by behavioral psychology, where learning is driven by rewards and punishments. It has been successfully applied to various domains, such as robotics, game playing, and resource management, demonstrating the versatility and robustness of this learning paradigm Sutton and Barto (1998); Mnih et al. (2015).

2.2 Reinforcement Learning

Machine learning can be broadly classified into three categories: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning deals with predicting outcomes based on labeled data, while unsupervised learning involves exploring unlabeled data to discover hidden patterns. Reinforcement learning, distinct from the other two, involves learning optimal strategies through direct interaction with the environment.

Formally, reinforcement learning is a computational approach where an agent seeks to maximize some notion of cumulative reward by making a sequence of decisions. The agent interacts

with an environment, makes observations, and receives rewards. Over time, the agent learns to take actions that maximize the long-term return, often defined as the discounted sum of future rewards Kaelbling et al. (1996); Sutton and Barto (1998).

2.2.1 Basic concepts and terminology

Terminology

Understanding the core concepts and terminology is crucial for grasping the mechanics of various algorithms. Below are the key terms:

- **Agent (A)**: The entity that interacts with the environment to achieve a goal. The agent's objective is to maximize cumulative reward over time.
- **Action (a)**: The set of all possible moves or decisions the agent can make. The agent selects actions based on a policy to influence the environment.
- **Environment (e)**: The external system with which the agent interacts. It provides the state to the agent and reacts to the agent's actions.
- **State (s)**: A representation of the current situation in the environment. The state provides the context needed for the agent to make decisions.
- **Reward (R)**: A scalar feedback signal received from the environment in response to an action. It reflects the immediate benefit of an action taken by the agent.
- **Policy (π)**: A strategy or a mapping from states to actions. It defines the agent's behavior and can be deterministic or stochastic.
- **Value (V)**: The expected cumulative reward from a given state, following a particular policy. It reflects the long-term desirability of states.
- **Q-value or action-value (Q)**: The expected cumulative reward of taking a particular action in a given state, under a specific policy. It helps the agent decide the best action to take.

These concepts are foundational to understanding how reinforcement learning algorithms work and how they can be applied to solve complex problems Sutton and Barto (1998).

2.3 Algorithms

Reinforcement learning algorithms are generally classified into two categories: model-free and model-based. Each category has its strengths and suitable application scenarios.

2.3.1 Model free

Model-free algorithms learn to make decisions without a model of the environment's dynamics. They rely on experience, often using trial-and-error methods to improve decision-making. These algorithms are particularly useful when the environment is complex or unknown, as they do not require explicit modeling of the system.

Key techniques in model-free RL include:

- Q-Learning: A popular value-based method where the agent learns a Q-function that estimates the expected utility of taking a particular action in a given state **?**.
- Deep Q-Networks: An extension of Q-learning that uses deep neural networks to approximate the Q-function, allowing it to handle high-dimensional state spaces, as demonstrated in playing Atari games Policy Gradient Methods: These directly optimize the policy by maximizing the expected reward, often using techniques like REINFORCE or Proximal Policy Optimization (PPO) Schulman et al. (2017).

2.3.2 Model Based

Model-based algorithms, in contrast, involve building a model of the environment's dynamics and using this model to plan actions. These algorithms can be more sample efficient than model-free methods since they can leverage the model to simulate and evaluate the outcomes of actions.

Examples include:

- **Dyna-Q:** An approach that integrates learning, planning, and acting by using a model to simulate experiences Sutton and Barto (1998).
- **World Models:**A recent approach that uses generative models to create a compact representation of the environment, enabling efficient planning and exploration Ha and Schmidhuber (2018).

Model-based methods are advantageous in scenarios where accurate modeling of the environment is feasible and can lead to more efficient learning.

2.4 Summary

In summary, reinforcement learning is a powerful framework for developing intelligent agents capable of autonomous decision-making. The field has evolved significantly, with deep reinforcement learning techniques like DQN and PPO setting new benchmarks in various applications. The choice between model-free and model-based methods depends on the problem at hand and the available resources for modeling the environment. As the field continues to grow, it holds promise for tackling increasingly complex and diverse challenges in artificial intelligence and beyond.

Chapter 3

Systematic literature review

In the following section, we delve into the reasoning behind the decision criteria for the literature review. Each criterion was carefully considered to reach a selection of articles and works that are significant to the problem domain. Through this process, we have examined a collection that encapsulates the essence of relevance and importance, crucial for the advancement of the research.Therefore, the chapter presents the reasoning to reach the final pool of articles.

3.1 Identification of key articles

The key papers were identified using the Web of Science platform, a well-recognized database of research papers, with the following query: 'reinforcement learning' AND 'assignment' AND ('last-mile' OR 'e-commerce' OR 'delivery' OR 'demand' OR 'logistics' OR 'traffic network' OR 'mobility' OR 'vehicles' OR 'transport').

The search focused on the title, abstract, and author keywords fields, yielding 269 results as of 22/02/2024. Many of these articles were not directly relevant to the research, as they covered a wide range of topics outside of the central investigation. To narrow down the list and identify valuable articles, two key steps were defined. In these steps, an article was considered important if it addressed 'Last-Mile' delivery with trucks (Section 3.1.1) or involved geographical area partitioning in other relevant topic domains (Section 3.1.2). These steps can be broken down to two core steps:

1. **Main Theme:** The title and abstract of each article were analyzed to determine its relevance to the study. Themes such as credit or coupon assignment, military applications of UAVs, wireless networks, internet connections, traffic issues, scheduling in various domains, and others were discarded.

2. **Content of the article:** After identifying the main topics covered by the proposed papers, it became necessary to read the articles more thoroughly to verify their relevance to the problem domain. Articles focused exclusively on routing scheduling, cost minimization, food order recommendations, or traffic were discarded, as they lacked any connection to

urban assignment or geographical partitioning. After applying the second and last step a final list composed by 17 articles was reached.

From the initial list of 269 articles, the first step narrowed it down to 32 promising articles. After applying the second and final step, the list was further refined to a final selection of 17 articles.

It is worth mentioning that the quartile, impact factor of journals and conferences, and the number of citations were not considered. This decision was made because the domain problem is relatively recent and the number of total articles is short. Some relevant papers, though closely aligned with our study, were published in a variety of journals and conferences with limited citations. As a result, all articles that aligned with our domain problem were given consideration.

3.1.1 Criteria for defining an article as focused on last-mile delivery

Last-mile delivery, defined as the final leg of the delivery journey from a transportation hub to the end consumer, is a critical component of logistics and supply chain management. Therefore, an article is considered focused on last-mile delivery if it meets the following criteria:

1. **Direct involvement in final delivery phase**: The article must address processes or challenges specifically related to the final step of delivering goods from a transportation hub or distribution center directly to the end consumer. This phase is often characterized by short distances and time-sensitive operations.

2. **Application in urban or localized settings**: Last-mile delivery is typically associated with dense urban areas where delivery to the end customer involves navigating complex and congested environments. The article should ideally focus on such settings.

3. **Optimization strategies for last-mile logistics**: The article should discuss strategies or technologies aimed at improving efficiency, reducing costs, or enhancing customer satisfaction in the last-mile segment. This includes task allocation, route optimization, dynamic batching, and real-time dispatching.

4. **Use of advanced technologies**: The implementation of modern technologies such as artificial intelligence, machine learning, or autonomous delivery vehicles to solve last-mile delivery challenges is a key indicator. This includes the use of UAVs, crowdsourced delivery platforms, and deep reinforcement learning for task assignments.

3.1.2 Criteria defining geographical area partitioning in logistics and delivery systems

Geographical area partitioning refers to the division of a geographic region into smaller areas or zones to optimize various operational tasks, such as delivery dispatching, routing, and resource allocation. In the context of logistics and delivery systems, several criteria can determine whether geographical area partitioning is present:

1. **Explicit mention of area division**: The text directly mentions the division of a region into distinct areas for managing deliveries or tasks. This can include terms like "delivery areas," "zones," "regions," or "territories."

2. **Regional distribution of resources**: The system allocates resources, such as delivery vehicles or personnel, based on specific geographic regions. This distribution helps manage workloads and optimize service efficiency within each designated area.

3. **Use of Geographic Information Systems (GIS)**: The implementation of GIS tools to delineate and manage service areas. GIS can create visual and data-driven maps that highlight how areas are partitioned for operational purposes.

4. **Task and resource optimization based on areas**: Strategies that involve optimizing tasks and resources based on geographical areas. This includes methods like dynamic batching, task buffering, or resource rebalancing, where the optimization is geographically influenced.

5. **Dynamic area adjustment**: The capability to dynamically adjust the boundaries or characteristics of geographic areas based on real-time data, demand fluctuations, or operational efficiency requirements.

6. **Community-level or location-based data utilization**: The use of community-level data, Area of Interest (AOI) information, or spatial-temporal correlations to define and manage geographical partitions.

3.2 Themes

The methodology to group the articles in six main categories was feeding ChatGPT with the titles of the articles as well as their corresponding abstract and key words. After that, the chat figured that splitting into 6 main categories was the most appropriate approach.

There are many variations to this problem as previously mentioned. An analysis of the articles allowed to identified 6 different themes, as follows:

- **Category 1: Last-mile delivery optimization with reinforcement learning**: Include articles focused on optimizing various aspects of last-mile delivery using reinforcement learning techniques. They explore different methods, such as adaptive agent-based approaches, DRL, and multi-agent RL, to improve efficiency, reduce delivery times, and enhance customer satisfaction in real-time delivery services. The common theme is the application of advanced machine-learning techniques to solve practical challenges in last-mile logistics.

- **Category 2: Spatial crowdsourcing and task assignment with reinforcement learning**: Include articles that address the optimization of task assignment in spatial crowdsourcing environments using reinforcement learning. They propose frameworks and algorithms that

leverage knowledge graphs, geographic partitioning, and Deep Q Networks (DQNs) to enhance task allocation efficiency and worker satisfaction. The focus here is on utilizing crowdsourcing models to distribute tasks dynamically and efficiently.

- **Category 3: Integration of path planning and task assignment for autonomous vehicles**: These articles are focus on the integration of path planning and task assignment for autonomous vehicles, specifically UAVs and USVs, using reinforcement learning techniques. They address the challenges of navigating complex environments and dynamically assigning tasks to enhance operational efficiency and effectiveness in autonomous delivery systems.

- **Category 4: Electric vehicle charging and allocation optimization**: These articles deal with optimizing the assignment of electric vehicles (EVs) to charging stations using reinforcement learning and agent-based modeling techniques. They focus on improving the efficiency of EV charging infrastructure, minimizing energy costs, and balancing traffic flow and electrical grid loads.

- **Category 5: Logistics and freight optimization with reinforcement learning**: These articles focus on optimizing logistics and freight operations using reinforcement learning techniques. They address challenges in trailer allocation, truck routing, and dynamic fleet assignment, proposing novel frameworks that integrate practical features like bipartite graph assignment and territory-based assignment to improve decision-making efficiency in logistics.

- **Category 6: Mobility-on-demand and dynamic rebalancing**: Unlike other categories that address broader aspects of delivery optimization or logistics, this category only holds one article. It uniquely targets the dynamic adjustment of vehicle positioning based on real-time demand patterns. This approach enhances the efficiency of ride-sharing services by reducing idle time and increasing vehicle utilization, which is a distinct challenge compared to traditional last-mile delivery or task assignment problems.

Each category represents a distinct area of research within the broader context of logistics, delivery optimization, and autonomous vehicle operations, unified by the common application of reinforcement learning and agent-based modeling techniques. This organization highlights the specific problem domains and the innovative solutions proposed to address them.

3.2.1 Last-Mile delivery optimization with reinforcement learning

Articles that focus on last-mile delivery typically address the unique challenges and optimization strategies pertinent to this final delivery phase. For instance, the article Lu et al. (2024)emphasizes strategies to enhance efficiency and service quality in instant delivery services, a quintessential example of last-mile logistics. Similarly, research on Li et al. (2023) delves into the real-time

allocation and matching of delivery tasks to couriers, underscoring the dynamic nature of urban last-mile logistics.

Studies leveraging advanced technologies such as deep reinforcement learning also contribute significantly to this field. The paper Wang et al. (2023b) illustrates how AI can optimize order assignments, thus enhancing the efficiency of last-mile food delivery. Furthermore, the exploration of UAVs in Escribano et al. (2022) demonstrates innovative approaches to overcoming urban delivery challenges.

Conversely, articles that do not focus on last-mile delivery typically address broader aspects of logistics and transportation, such as the optimization of trailer allocation and truck routing, or the strategic planning of EV charging infrastructures. These studies, while valuable, do not directly engage with the specific issues and solutions pertinent to the last-mile segment.

Overall, the critical criteria for defining last-mile delivery research include a focus on the final delivery phase to the end consumer, optimization strategies for this segment, and the use of advanced technologies to address its unique challenges.

For instance, Lu et al. (2024) introduces an Adaptive Agent-Based Order Dispatching (ABOD) approach to manage order dispatching using task buffering and dynamic batching strategies. focuses on real-time task assignment, proposing a time-aware batch matching algorithm combined with deep reinforcement learning to optimize parcel collections to couriers. Similarly, Wang et al. (2023b) and Wang et al. (2023a) aim to enhance delivery efficiency by considering dynamic rider behaviors and cooperative area assignments, respectively.

Other studies, such as Damanik et al. (2022) use multi-agent reinforcement learning to address delivery assignments in hybrid-transit networks, focusing on minimizing fuel consumption and improving coordination. *A deep reinforcement learning approach for the meal delivery problem* and *Courier routing and assignment for food delivery service using reinforcement learning* leverage deep reinforcement learning to optimize meal delivery and courier routing, aiming to maximize profit and efficiency while reducing delays. Each of these studies highlights the versatility and potential of reinforcement learning in transforming last-mile delivery logistics.

3.2.2 Spatial crowdsourcing and task assignment with reinforcement learning

An optimized task assignment framework based on crowdsourcing knowledge graph and prediction focuses on optimizing task assignment in spatial crowdsourcing by leveraging a crowdsourcing knowledge to predict the distribution of tasks and workers, capture worker preferences, and dynamically assign tasks to improve completion rates and worker satisfaction. *Deep Reinforcement Learning for Crowdsourced Urban Delivery* optimizes the assignment of delivery requests to crowdsourced couriers in urban environments to enhance the efficiency and quality of crowdsourced urban delivery. *Task Allocation with Geographic Partition in Spatial Crowdsourcing* addresses task allocation in spatial crowdsourcing systems using geographic partitioning, by allocating tasks to workers in a spatial crowdsourcing context. The framework involves partitioning the area based on allocation centers using Voronoi diagrams and task allocation.

Geographical area partitioning is a critical strategy in optimizing logistics and delivery operations. In the context of modern delivery systems, various approaches integrate this concept to enhance efficiency, reduce costs, and improve service quality. For instance, the article *"Efficient Adaptive Matching for Real-Time City Express Delivery"* mentions the distribution of parcels to stations based on their service regions, inherently involving geographic partitioning to manage deliveries within a city. Similarly, *"GCRL: Efficient Delivery Area Assignment for Last-mile Logistics with Group-based Cooperative Reinforcement Learning"* discusses dividing a city into multiple delivery areas using community-level AOI information to optimize delivery assignments.

In contrast, some articles do not explicitly mention geographical partitioning, such as *"An adaptive agent-based approach for instant delivery order dispatching,"* which focuses on strategies like task buffering and dynamic batching without detailing area divisions. However, the significance of geographic considerations is implied in models and frameworks that optimize tasks based on spatial data, even if formal partitioning is not described.

Further examples include articles like *"An optimized task assignment framework based on crowdsourcing knowledge graph and prediction,"* which involves predicting task distributions based on spatial correlations, indicating a nuanced form of geographic partitioning. *"Integrated Path Planning and Task Assignment Model for On-Demand Last-Mile UAV-Based Delivery"* incorporates case studies that consider geographic partitions for drone deliveries, highlighting practical applications in urban settings.

In conclusion, geographical area partitioning emerges as a pivotal element in modern logistics and delivery frameworks, enhancing operational efficiency through structured and dynamic management of geographic zones. The diverse approaches in the literature reflect the multifaceted nature of this strategy, from explicit area divisions to more implicit spatial optimizations, underscoring its relevance in the field.

3.2.3 Integration of path planning and task assignment for autonomous vehicles

Escribano et al. (2022) focuses on the integration of task assignment and path planning for UAV-based last-mile delivery, whereas Wen et al. (2022) aims to enhance the autonomy and operational efficiency of unmanned surface vehicles (USVs). It addresses the challenges of deploying USVs in complex maritime environments where they need to autonomously navigate, avoid obstacles, and allocate themselves to specific operational areas dynamically.

3.2.4 Electric vehicle charging and allocation optimization

A Data-driven Agent-based Planning Strategy of Fast-Charging Stations for Electric Vehicles integrates EV charging infrastructure with electrical distribution networks using data-driven, agent-based modeling techniques. It considers microscopic EV behaviors to optimize the placement and operation of fast-charging stations (FCSs). *A Reinforcement Learning-based Assignment Scheme*

for EVs to Charging Stations optimizes the assignment of EVs to charging stations in urban environments, finding an optimal policy for assigning EVs to charging stations based on real-time conditions and constraints.

3.2.5 Logistics and freight optimization with reinforcement learning

Kalantari et al. (2023) explores optimizing trailer allocation and truck routing to minimize the total distance traveled by trucks to fulfill pickup and delivery orders. The Min and Kang (2021) introduces a learning-based approach for optimizing dynamic fleet assignment in digital freight brokerages to handle uncertainties and operational complexities in matching delivery requests to trucks.

3.2.6 Mobility-on-demand and dynamic rebalancing

The article of Castagna et al. (2021) is categorized separately because it focuses specifically on improving vehicle rebalancing in ride-sharing enabled Mobility-on-Demand (MoD) systems using a dynamic, demand-responsive approach. It proposes a system that adjusts vehicle positioning based on real-time demand patterns to enhance the efficiency of ride-sharing services.

3.3 Problem formulation

When formulating problems in the context of optimizing logistics and task assignments for vehicle, it is crucial to consider a comprehensive set of elements that define the scope and dynamics of the problem. Comparing different formulations requires a thorough understanding of key topics such as Dataset, Agents, Actions, States, Data Constraints, Reward, and Success Metrics. Each of these components plays a significant role in shaping the problem and the approach taken to solve it.

The **Dataset** is the foundation of any data-driven approach, providing the raw information needed to train models and evaluate their performance. Typical datasets include vehicle movements, delivery requests, environmental conditions, and more. A well-structured dataset ensures that the models can learn effectively and make accurate predictions. Therefore, comparing datasets across different formulations helps in understanding the variability in data sources, preprocessing techniques, and the richness of the information available.

Agents are the decision-makers within the problem space, whether they are autonomous vehicles, delivery drones, or human drivers. Each agent operates based on the information it receives and the strategies it employs. The behavior and capabilities of agents must be clearly defined to understand how they interact with the environment and other agents. By comparing the agent formulations, one can assess the complexity and realism of the models, as well as their applicability to real-world scenarios.

The **Actions** available to agents determine the possible strategies they can adopt. These actions could include routing decisions, task assignments, charging strategies, or obstacle avoidance maneuvers. The diversity and granularity of actions impact the agent's ability to optimize its

objectives. Evaluating different action sets helps in understanding the flexibility and constraints imposed on agents in various formulations.

States represent the current condition of the system and its environment, providing the context within which agents make decisions. States can encompass a wide range of factors, such as vehicle locations, battery levels, traffic conditions, and delivery deadlines. A detailed state representation is essential for capturing the dynamics of the problem accurately. Comparing state definitions highlights how different models capture the complexity of the environment and the level of detail they consider.

Data constraints refer to the limitations and requirements related to the dataset and the problem environment. These could include constraints on data availability, accuracy, temporal resolution, and spatial coverage. Understanding these constraints is critical for assessing the feasibility and reliability of different formulations. By comparing data constraints, one can identify potential challenges and limitations that may affect the performance and applicability of the models.

The **Reward** function guides the learning process by providing feedback on the actions taken by agents. It encapsulates the objectives of the problem, such as minimizing delivery times, reducing energy consumption, or maximizing service quality. A well-designed reward function is crucial for aligning agent behavior with desired outcomes. Comparing reward formulations helps in understanding the priorities and trade-offs considered in different problem settings.

Finally, **Success metrics** are used to evaluate the performance of the models and solutions. These metrics can include measures like delivery efficiency, customer satisfaction, operational cost, and system robustness. Clear and relevant success metrics are essential for objectively comparing different formulations and determining their effectiveness. By analyzing success metrics, one can gauge the practical impact and benefits of various approaches.

In summary, comparing different formulations of problems in logistics and task assignment requires a comprehensive examination of datasets, agents, actions, states, data constraints, reward functions, and success metrics. Each of these components contributes to the overall problem structure and influences the strategies and outcomes of the solutions developed.

Reference	Dataset	Agents	Actions	States	Data Constraints	Reward	Success Metrics
Lu et al. (2024)	Synthesized data from delivery layouts; real-world data from a fresh food e-commerce company in Shanghai.	Couriers and depot agents	Wait and Assign	Location of the customer, maximum acceptable delivery time, and order time	Couriers' maximum driving speed, capacity, and the working hours of the depot	Estimated time of arrival (ETA)	Customer satisfaction; Delivery distance; Completion rate; Hourly working couriers
Li et al. (2023)	Road Networks of Chengdu and New York City from OpenStreetMap	Couriers involved in the delivery process	Decisions regarding delivery tasks	condition of the system at a given time	None	Successful matching of couriers to tasks	Average total revenue, average elapsed time, and average completion ratio
Wang et al. (2023b)	Dataset info 3	Online Food Delivery (OFD) platform	Decisions regarding delivery tasks	Condition of the system at a given time	Data Constraints info 3	Feedback from the environment	AGD, ARTG, 5-minute Grab Rate (GR5), and 2-minute Grab Rate (GR2)
Wang et al. (2023a)	Real-world data from Beijing and Shanghai	Delivery Areas	Selection of couriers by delivery areas	States info 4	Courier grouping	Maximize the completion rates of pickup and delivery tasks	PCDT and PCPT
Tao et al. (2023)	Not specified	EV drivers	Choosing routes, charging stations, and charging times	Transportation, electricity, EV charging network	Investments, charging services, traffic flow, electricity network	Reflects the utility of EV drivers	None

Reference	Dataset	Agents	Actions	States	Data Constraints	Reward	Success Metrics
Kalantari et al. (2023)	Synthetic and real-world	Trucks	Pick up trailers by the trucks	The locations of trucks, trailers, and customers, and the assignments of trailers to trucks	One trailer for each order, enough trailers	Negative distance traveled by the trucks	Total distance traveled, ability to generalize and improvements in the model's performance
Quan and Wang (2023)	CHI (Chicago Taxi Dataset) and TF-TKY	Workers (taxi drivers) and users	Task assignment, tasks to workers with priority	Information about tasks, workers, and their embedding	Workers' average movement speed, maximum reachable distance, moving speed, maximum travel distance	1 for a successfully performed task assignment, -1 for an abandoned task	NFT, NCT, TCR, AWT, AMD
Escribano et al. (2022)	Not specified	UAVs	Pick-up locations, picking up parcels, traveling to drop-off locations, dropping off parcels, moving to charging locations, and charging	own of the 5 states: Idle, Pick-up, Drop-off, Travel to Charger, and Charging	Battery capacity and payload limits for each UAV. waiting time and energy expenditure of the tasks	Maximize profitability by optimizing the delivery paths and task assignments	Total profit of the operation, profitability, collision risk minimization, and efficient path planning
Damanik et al. (2022)	Simulation environment	Entities responsible for moving packages	Six discrete actions: stay in place, move west, east, south, north, or pick up/drop deliveries	The positions of agents and deliveries, as well as their active or done status	Bounded environment and the limitations on the agents' actions and observations	Points for picking up and delivering packages, penalties are applied for unnecessary actions	Delivery time efficiency, the number of accepted orders, system's responsiveness to dynamic demands
Jahanshahi et al. (2022)	Synthetic dataset and the Getir dataset(Turkish grocery delivery company)	Couriers	Accepting or rejecting orders and choosing their next destination	Distance of the courier from the order origin, destination, and the depot	None	Rewarded completed actions within time and penalties for rejecting orders	DTE, number of accepted orders, system's responsiveness

Reference	Dataset	Agents	Actions	States	Data Constraints	Reward	Success Metrics
Aljaidi et al. (2022)	25×25 grid representing an urban area (UK)	Electric Vehicle (EV)	Moving in 4primary directions and in four diagonal directions	Coordinates (x, y) in the grid environment	The distance the EV can travel.The energy consumption per kilometer. The capacity of the EV battery.	Minimizing the total cost of charging	The maximum cumulative reward; total energy cost of the EV
Wen et al. (2022)	Not specified	USVs	Velocity on the x and y axes	Current 2D position (x, y), distance between agents (dU), distance to surrounding obstacles (dO), and distance to target (dT)	Communication distance limitations	Positive rewards for maintaining communication and avoiding threats. Penalties for collision, threat area	Rewards system and the performance of the trained policy
Bozanta et al. (2022)	Not specified	Couriers	Moving up, down, left, right, picking up and delivering orders, maintain location and rejecting orders	Courier locations, order origins, and order destinations	None	Minimize total travel distance	None
Castagna et al. (2021)	Dataset info 4	Each agent controls a vehicle and each vehicle operates autonomously.	Parking (waiting or new zone rebalancing), picking up or dropping off passengers	Agent's position (latitude and longitude), available seats, and its destination	Max. of 5 passengers per vehicle and requests made after 15 minutes are excluded	Rewards for successfully picking up or dropping off passengers, penalties for impossible actions	PSR, Percentage of ride-sharing trips, AWT, detour ratios and DTV
Min and Kang (2021)	The set of delivery requests and the characteristics of trucks	Trucks	Assigning trucks to specific delivery requests	The set of trucks and delivery requests, truck's characteristics like location, availability, and load status	Trucks are restricted to specific delivery areas, and only full truckloads are assumed	Maximizing the total discounted reward over time by effectively matching trucks with delivery requests	Total discounted reward

3.4 Results

In the context of last-mile delivery, urban logistics, and task allocation, several studies have demonstrated the effectiveness of reinforcement learning (RL) and deep reinforcement learning (DRL) methods over traditional approaches in terms of scalability, efficiency, and overall performance.

A common theme across these studies is the improvement in system performance when RL-based methods are used. For instance, the Lu et al. (2024) framework, which incorporates task buffering and dynamic batching strategies, improved customer satisfaction by up to 275.42% and reduced delivery distance by 11.38%. Similarly, in the Li et al. (2023) DRL showed superior performance in optimizing task assignment, improving both the quality of assignments and platform revenue by dynamically adjusting the window size.

Reinforcement learning approaches often significantly outperform baseline methods, particularly in real-time decision-making environments. In the Wang et al. (2023a) framework, which utilizes Group-based Cooperative Reinforcement Learning, system efficiency improved by an average of 12%, showcasing the scalability of RL approaches as the number of agents increases. Moreover, the use of attention mechanisms, as seen in Actor-Critic Networks and Feedback Correlation (FC) Networks, enhances model performance by effectively capturing dynamic rider behaviors and improving the quality of recommendations in rider-centered food delivery systems.

Several studies employed graph-based methods alongside RL. For example, Graph Neural Networks (GNNs), used in Multi-level Attention Mechanisms, contributed to improved delivery area assignments. The integration of bipartite graph assignments with DRL, as seen in the RTT-DRL framework, demonstrated strong generalization capabilities and better performance compared to traditional algorithms.

Scalability is a key strength of multi-agent reinforcement learning (MARL) approaches. Multi-Agent Deep Reinforcement Learning (MA-DRL), combined with a Data-Driven Traffic Assignment Model (DA-TAM), improved the management of electric vehicle (EV) charging stations by 33%, optimizing charging demand, wait times, and congestion. Similarly, the MADDPG (Multi-Agent Deep Deterministic Policy Gradient) framework for unmanned surface vehicles (USVs) displayed better scalability and adaptability in dynamic environments.

In food delivery applications, DRL methods such as Dueling DQNs and Double Deep Q-Networks (DDQN) consistently outperform baseline strategies like Q-Learning and rule-based policies. These models achieved higher rewards, better task allocation, and more efficient routing strategies.

Overall, reinforcement learning approaches, particularly DRL, consistently show improvements over baseline strategies across various domains. They exhibit better scalability, efficiency in dynamic task allocation, and strong performance in real-time optimization tasks, making them highly suitable for complex, multi-agent environments such as last-mile logistics, urban delivery, and EV charging systems.

3.5 Summary

This chapter encapsulates the results from the systematic literature review conducted to understand the current landscape of reinforcement learning applications in logistics and transportation. By rigorously filtering and categorizing relevant articles, we have identified significant research contributions and organized them into six core themes. These themes collectively cover advancements in last-mile delivery, spatial task assignment, autonomous vehicle operations, electric vehicle infrastructure, freight logistics, and dynamic mobility services. The review not only clarifies the state of the art in these areas but also provides a comprehensive basis for future research directions and practical implementations.

Chapter 4

Methodology

The following chapter outlines the methodology used in this study, providing a detailed explanation of each step. Beginning with an overview of the synthetic data generation process (Section 4.1) to contextualize where data comes from. It moves through the exploratory data analysis (Section 4.2) and the features developed to generate them. Because of the projects' complexity, major assumptions are made for better understanding and to take away some of its sophistication. Additionally, it includes a detailed description of the dataframe utilized (Section 4.3), which derive from the features created. The chapter then transitions into the process of modeling the problem as a Markov decision process(Section 4.4), leading to the complete implementation of the environment. Finally, it discusses the deep reinforcement learning algorithms applied and the reasoning behind their selection.

4.1 Original data

This section provides a detailed overview of the synthetic data generation process and the feature engineering applied to prepare the data for training our Reinforcement Learning (RL) agent.

Given the initial dataset lacked sufficient features to model the problem effectively as a Markov Decision Process (MDP), additional features were engineered to enhance the dataset. The original dataset comprises 50 working days of last-mile delivery data. However, for the purpose of training and experimentation with our RL agent, this chapter focuses on a subset of 20 working days, which amounts to 60,000 data points.

«Feature engineering involves generating clusters using the k-means algorithm across all 50 days of data. Each cluster is represented by its central point, computed as the average Position_X and Position_Y of all data points within the cluster. These central points help in capturing the spatial distribution of delivery locations and are crucial for understanding and optimizing delivery routes.

The synthetic data was initially created with a focus on real-world applicability, including detailed spatial and demand features. The data used for RL training thus includes these enriched

features to provide a comprehensive representation of the delivery environment, which is critical for developing and refining effective delivery strategies.

4.1.1 Understanding the synthetically generated data

In generating synthetic data for a last-mile delivery system, the focus was on the Porto Metropolitan Area (AMP) in North Portugal, using methods developed by a past FEUP student in their Master's thesis Torres (2022). Given the unavailability of historical data, it was essential to create a realistic dataset to simulate the operations of a delivery company in North Portugal. The data generation process began by defining Basic Units (BUs) as postal codes in the CP7 format. To model the spatial distribution of these BUs, centroids were used to construct a Voronoi Diagram, partitioning the geographic space into cells where each cell is closest to its respective centroid. The diagram was cropped to fit the AMP boundaries, ensuring the validity of the spatial partitioning.

For client simulation, random locations representing potential clients were generated within each Voronoi cell, with the number of clients proportional to the area's population density, derived from intersecting Voronoi cells with Census data. This approach ensured that higher population density areas had more potential clients, reflecting real-world client distribution. The dataset consisted of 150,000 data points across 50 working days, with each day featuring a constant 3,000 deliveries. Each record included coordinates (longitude and latitude) to specify delivery locations.

The daily delivery demand was modeled using a Poisson distribution, with the parameter λ representing the expected number of parcels per client per day. λ was calculated based on the target number of active clients and the total number of potential clients. This method provided a realistic simulation of daily delivery demands by determining active clients and their delivery needs. The dataset included a maximum of 30 drivers, with the selection of 3,000 active clients per day reflecting a diverse and realistic distribution of delivery requests.

4.1.2 Input data: specifications and operational details

The dataset used in this last-mile delivery system simulation captures key operational details relevant to the delivery operations in the North of Portugal. The delivery operations are based on a consistent daily workload of 3,000 deliveries, ensuring a manageable and realistic simulation of day-to-day activities.

The delivery fleet is composed of 80 vehicles, divided into small and medium-sized vans to accommodate different delivery needs. These vehicles operate within a structured geographical framework, with the delivery area divided into sectors. Each sector is assigned to a specific driver, ensuring a balanced workload and optimized route management.

The maximum number of drivers available for these operations is set at 30, with each driver operating within a sector designed to maximize efficiency while minimizing travel distance. The daily workload is calibrated to fit within an 8-hour work schedule, which aligns with the average working hours in Portugal. This setup ensures that the simulated delivery operations are not only efficient but also adhere to typical working conditions, providing a realistic and practical approach to managing last-mile deliveries in the region.

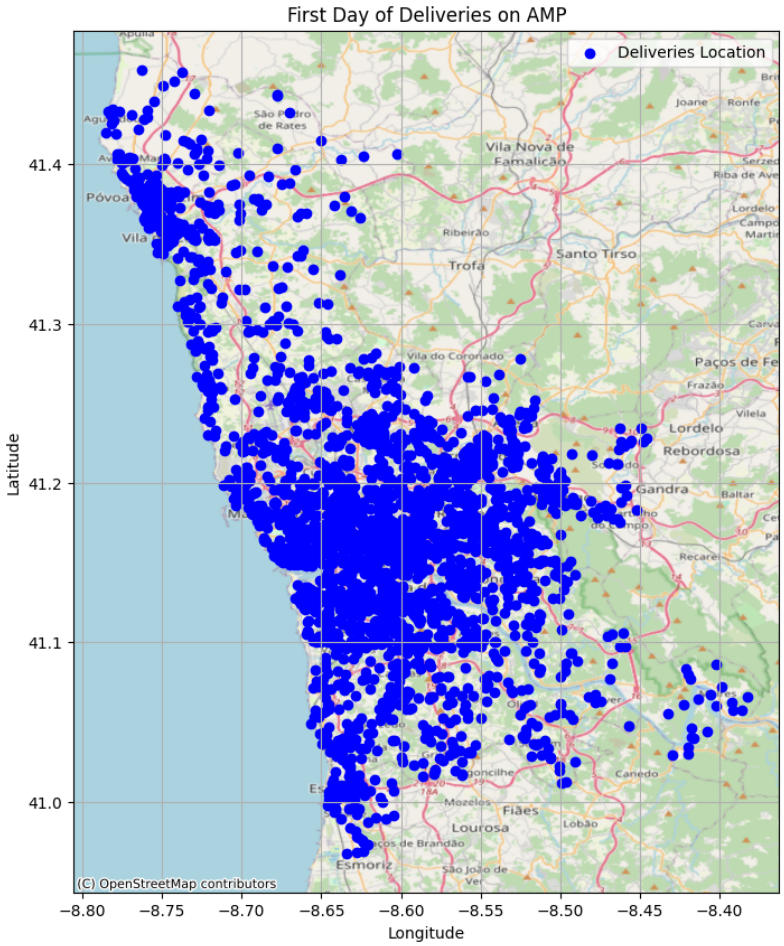

Figure 4.1: Methodology

4.1.3 Methodology

We begin by using **initial data** from 50 working days. **Feature engineering** is applied to create a **final deliveries dataframe**, which includes available workers, delivery IDs, and day-specific data. Next, a **K-means clustering** (with $k = 30$) divides the delivery zones based on delivery density. Each cluster is assigned a central point (X_mean, Y_mean) representing the average delivery location.

After clustering, we compute key **features** like the **average Euclidean distance** from each cluster center to the delivery points and the **distance to the warehouse**. Clusters are categorized by size and visualized using **K-means visualizations** and **cluster daily distributions** to analyze patterns and optimize delivery zones.

Lastly, we focus on **driver allocation** and stress testing. Each day, different cluster centers generate a **stress scenario dataframe**, helping to simulate varying delivery conditions. This leads to the creation of a **driver dataframe** to efficiently allocate drivers according to cluster types.

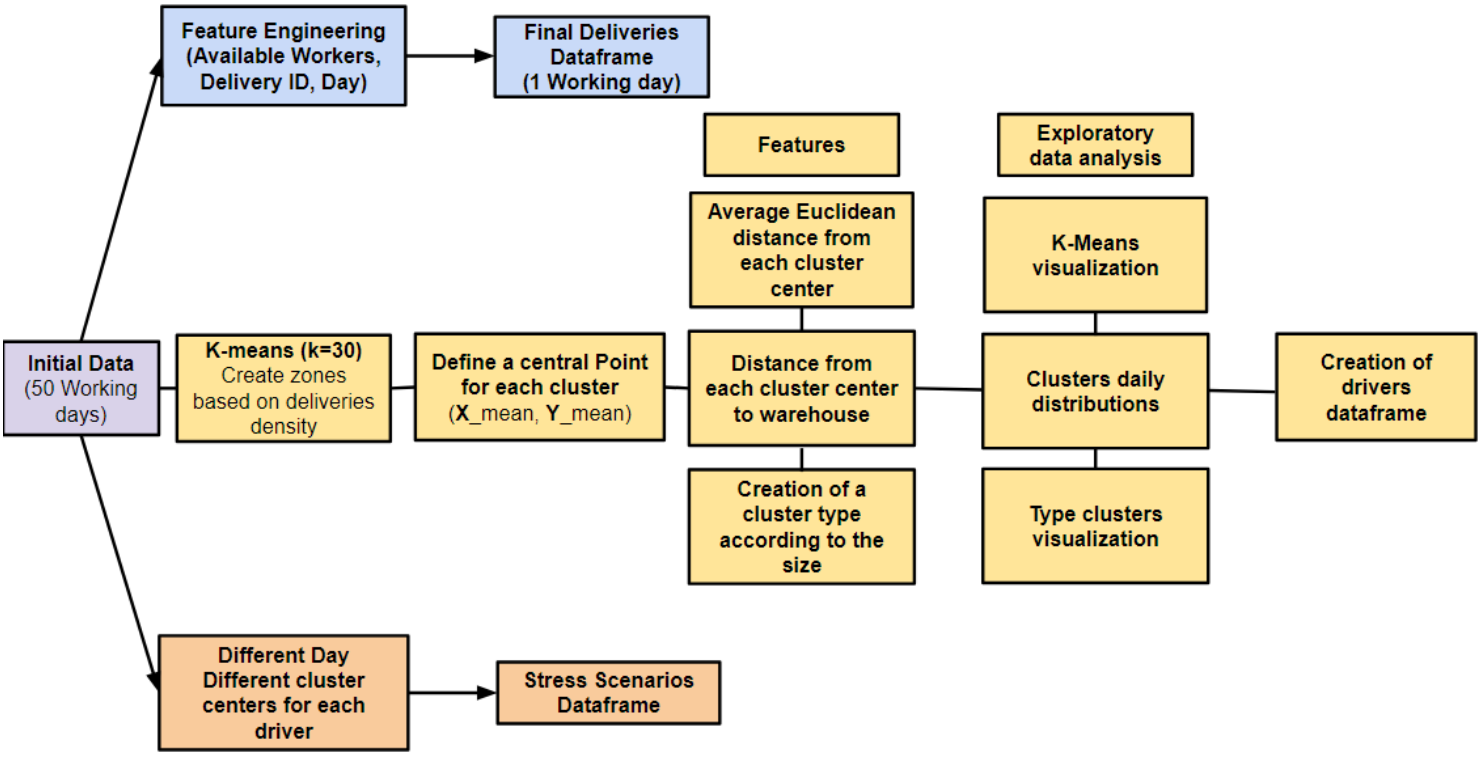

Figure 4.2: Methodology

4.1.4 Feature engineering

The delivery dataset, containing only the x and y coordinates of each delivery point, is insufficient for modeling our problem as a Markov decision process. To address this limitation, additional variables need to be introduced into the dataset. Below, we outline the features that were created and their respective purposes for this project:

- **Available Drivers**: This feature comprises a list containing the IDs of drivers available for each day. It plays a crucial role in driver assignment to deliveries. Initially, for simplicity, all days will have a full complement of 30 available drivers. However, as part of stress testing to evaluate overfitting, some drivers will be intentionally removed, along with certain deliveries.

- **Day**: To uniquely identify each day, a variable named 'day' was introduced. This feature is essential as each day corresponds to an episode in the Markov decision process. Upon reaching the final day, the environment is reset to the initial state, which is day one.

- **Delivery ID**: In order to map driver IDs to deliveries, a unique identifier called delivery ID was created. This variable distinguishes each individual delivery in the dataset.

The Table 4.1 presents 60,000 rows of data which correspond to 20 working days. Without the feature Engineering, the data would exclusively have the Position_X and the Position_Y. That information would not be enough to model the problem as a Markov Decision Process.

The previous data can be described as 'perfect', because each day consistently has 3000 deliveries and 30 available drivers. The data is going to be used to train the agent. It is worth mentioning that the fact that there are no fluctuations in the number of deliveries and available drivers is to test the overfit and how the agent responds in an environment of stress. The stress will be explored deeply in chapter 5, but it is basically the creation of imbalances in the data to see if the agent can adapt its decision making.

Table 4.1: Example Table of the Data with all the new features

Position_X	Position_Y	Day	Available drivers	Delivery id
-8.617901	41.143660	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	0
-8.620139	41.156769	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	1
-8.614335	41.149889	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	2
-8.619269	41.158992	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	3
-8.619709	41.158686	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	4
...	...	...	...	...
-8.635547	41.188199	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	2995
-8.636044	41.187801	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	2996
-8.653333	41.100311	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	2997
-8.643497	41.228701	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	2998
-8.609995	41.115687	1	[0, 1, 2, 3, 4, 5, ..., 26,27,28,29]	2999

4.1.5 Assumptions about the data

In the initial phase of this project, the agent will be trained on a theoretical perfect environment. However, it's important to acknowledge that real-world scenarios are often dynamic and unpredictable. Here are the assumptions made about the data:

- **Consistent daily deliveries**: For the purpose of simulation, each day consists of a fixed number of deliveries, set at 3000. This allows for a standardized testing environment. However, it's understood that in reality, daily delivery volumes can fluctuate due to various factors such as seasonal demand, promotions, or unforeseen circumstances.

- **Stable driver availability**: It is assumed that there are consistently 30 drivers available for delivery each day. While this simplifies the initial modeling, it's acknowledged that in practice, driver availability may vary due to factors like illness, vacations, or other scheduling conflicts.

- **Assumed delivery completion**: In this model, it's assumed that all deliveries are successfully completed. While this simplification facilitates algorithm development, real-world delivery logistics often encounter situations where deliveries cannot be fulfilled due to recipient unavailability or other factors. This aspect will be further explored and addressed in later stages of development.

- **Equal priority for deliveries**: Since there is no chronological information available regarding the purchase of deliveries, it is assumed that all deliveries have equal priority. This simplifies the initial algorithm design, but future iterations may consider implementing priority-based scheduling mechanisms.

- **Alignment of sectors and drivers**: The number of sectors is assumed to be equal to the number of drivers available. This simplifies the distribution of deliveries among drivers, ensuring each driver is assigned to a specific sector for efficient coverage. However, in practice, the alignment of sectors and drivers may vary based on geographical factors, workload distribution, and other considerations.

- **Uniform traffic conditions**: The traffic conditions within each sector remain consistent throughout the day and do not vary based on time or location. While this simplification aids in initial algorithm development, future iterations may incorporate real-time traffic data to optimize delivery routes in response to traffic fluctuations.

- **Vehicles capacity**: It is assumed that the vehicles have capacity to store all packages, and all drivers drive vehicles of the same capacity. This adds simplicity to the environment since it is not necessary to translate the capacity of the vehicle in an additional constraint that adds even more complexity to the environment.

These assumptions provide a foundational framework for developing and testing the delivery optimization algorithm. However, it's important to iteratively refine and adjust these assumptions based on real-world data and feedback to create a more accurate and adaptable model.

4.2 Exploratory Data Analysis

The following section explores all the visualizations carried out in this project. With the goal of a better comprehension of the data as well as key points for the creation of auxiliary materials for the creation of data frames for the environment.

4.2.1 Clustering the deliveries

The exploratory data analysis focused on understanding the geographical distribution of deliveries within the AMP (Area of Maximum Population) by identifying densely populated areas. To achieve this, we applied the k-means unsupervised learning algorithm to cluster the delivery locations. This algorithm was used to partition the data into 30 clusters, corresponding to the maximum number of available drivers in the company. Each delivery location was iteratively assigned to the nearest cluster center, calculated based on Euclidean distance.

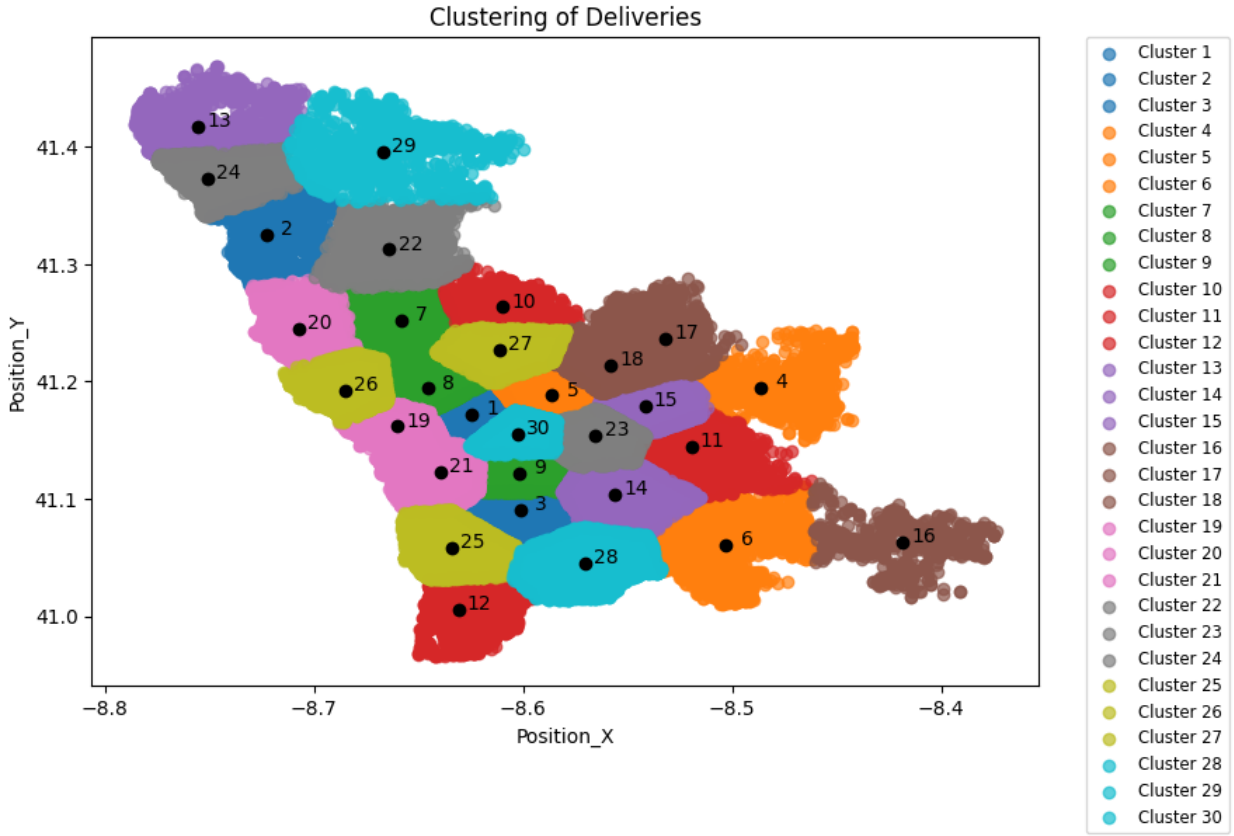

Figure 4.3: Clusters and corresponding central points

The results of this clustering process are depicted in Figure 4.3. The figure not only shows the distribution of the 30 clusters but also highlights the central points of each cluster, which were computed as the average Position_X and Position_Y of all points within each cluster. These central points are crucial as they represent the geographical areas of operation (AOI) for each driver.

By using these central points, we can assign specific AOIs to drivers, ensuring consistent delivery assignments within these zones. Additionally, these central points are essential for calculating rewards as discussed in (Section 4.4). This approach allows for the AOIs to be adjusted as needed to optimize delivery efficiency and enhance overall operational performance.

4.2.2 Cluster types

Clusters exhibit variability in their properties, particularly in terms of the average Euclidean distance of their members to the cluster center. To classify the clusters into distinct categories—small, medium, and large—we established thresholds based on these average distances. This classification serves as the foundation for assigning rewards to each of the three cluster types.

The average Euclidean distance was computed for each cluster by measuring the distance of each point within the cluster from the central point, defined as the mean position of all points in that cluster. The distances were then averaged to derive a single representative value for each cluster.

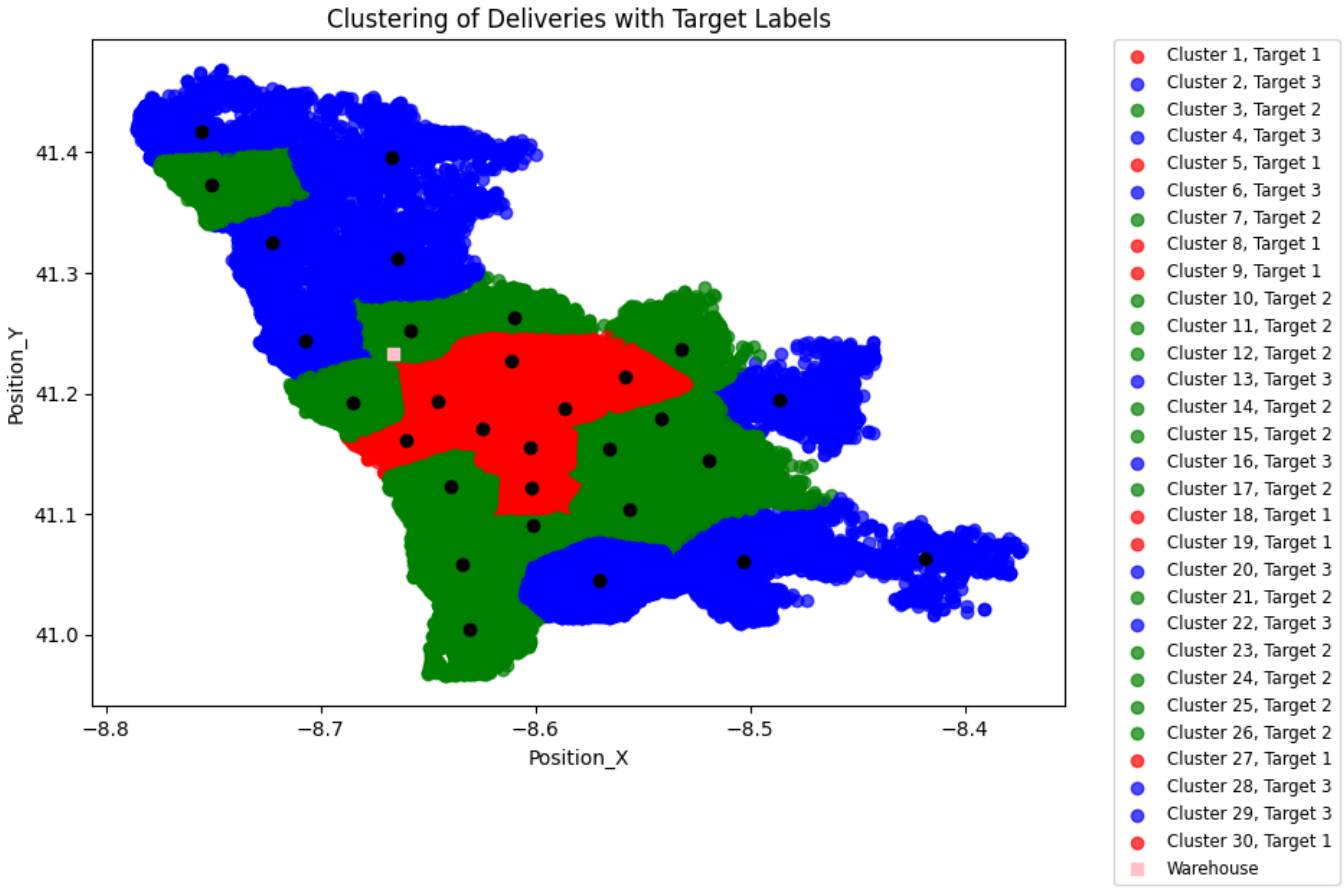

Figure 4.4: Cluster types and warehouse

The classification criteria was defined as follows:

- **Small Clusters**: Clusters with an average Euclidean distance less than or equal to 14 units were classified as small. These clusters are relatively compact, with points closely grouped around the central mean.
- **Medium Clusters**: Clusters with an average Euclidean distance between 14 and 20 units were categorized as medium. These clusters show moderate spread in their data points.

- **Large Clusters**: Clusters with an average Euclidean distance greater than 20 units were identified as large. These clusters have widely dispersed points relative to the central mean.

4.2.3 Daily distribution of the data per cluster

To analyze the daily distribution of delivery counts across different clusters, we calculated the number of deliveries for each day within each cluster. This analysis provides a deeper understanding of the variability and consistency of deliveries across different clusters.

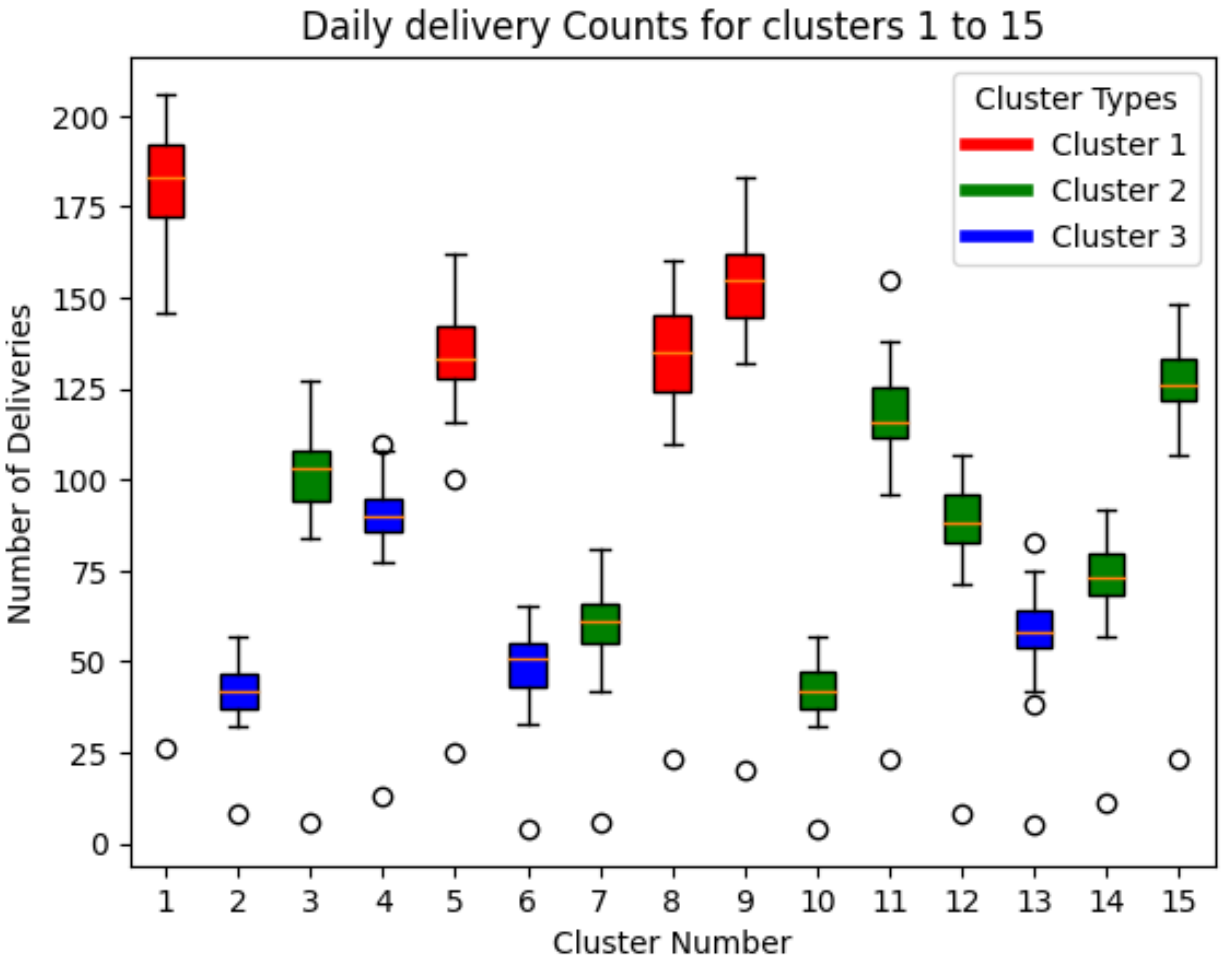

Figure 4.5: Cluster types and warehouse

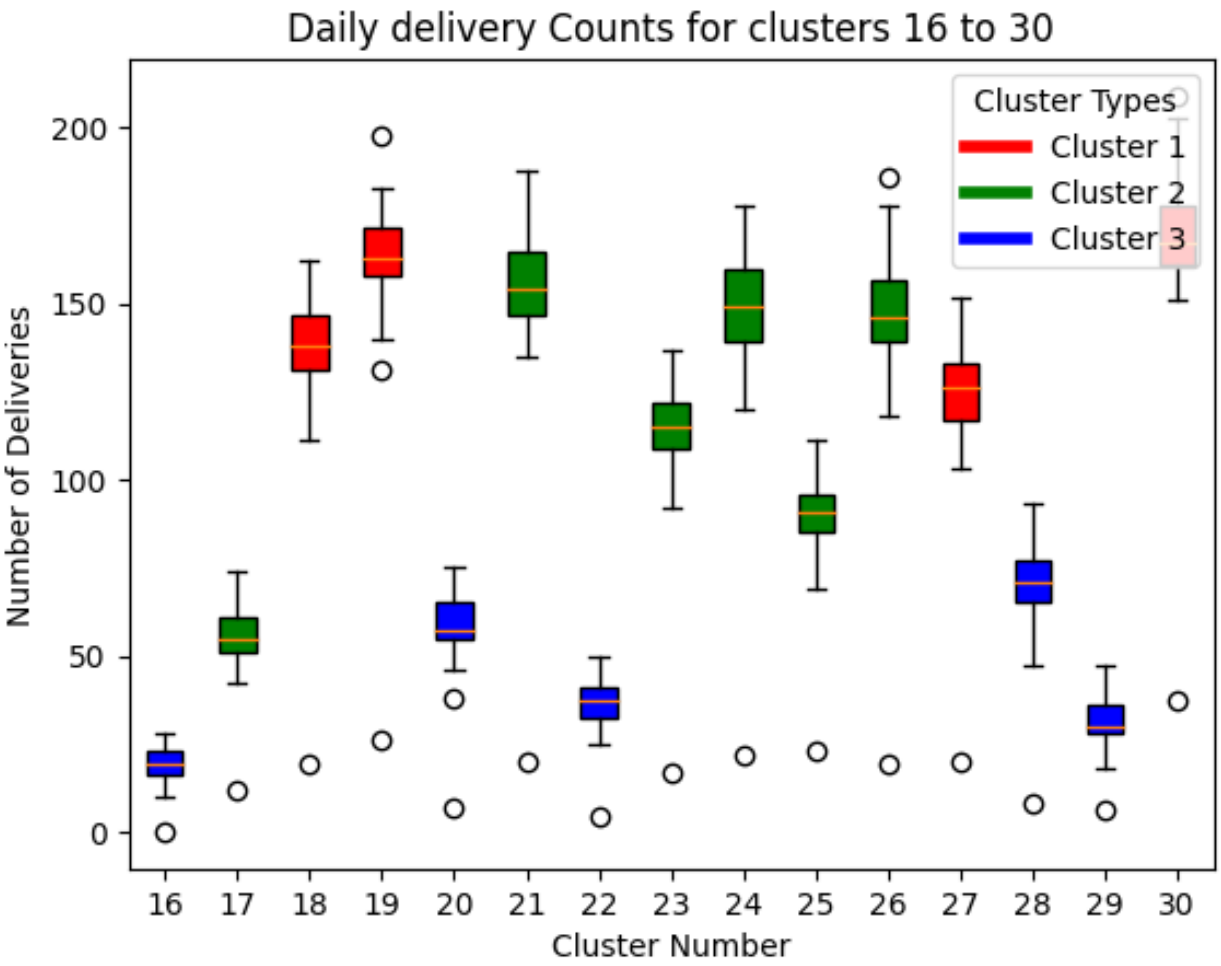

Figure 4.6: Cluster types and warehouse

The box plot in Figure 4.5 shows the delivery counts for each cluster, from Cluster 1 to Cluster 15 and the Figure 4.6 from Cluster 16 to Cluster 30. Here are some key observations:

- **Median**: The red line inside each box represents the median number of deliveries for that cluster.
- **Interquartile Range (IQR)**: The box represents the IQR, where the lower edge of the box is the 25th percentile, and the upper edge is the 75th percentile. This range shows the middle 50% of the data.
- **Whiskers**: The whiskers extend to the smallest and largest values within 1.5 times the IQR from the lower and upper quartiles, respectively.
- **Outliers**: Points outside the whiskers are considered outliers and are plotted individually.

From the box plot, we can see the variability in delivery counts across different clusters. For example, Cluster 1 has a higher median and a larger IQR compared to other clusters, indicating

more variability in daily deliveries. Cluster 10, on the other hand, shows fewer deliveries and a tighter distribution.

This detailed analysis of delivery counts helps in understanding the operational efficiency and consistency of different clusters, providing valuable insights for further optimization.

4.2.4 Daily Delivery Counts by Cluster Type

The analysis of daily delivery counts by cluster type, as presented in Figure 4.7, categorizes deliveries into three main types based on delivery distances: Cluster 1 (red), Cluster 2 (green), and Cluster 3 (blue). This categorization allows for a detailed examination of delivery patterns and operational efficiency across varying delivery distances.

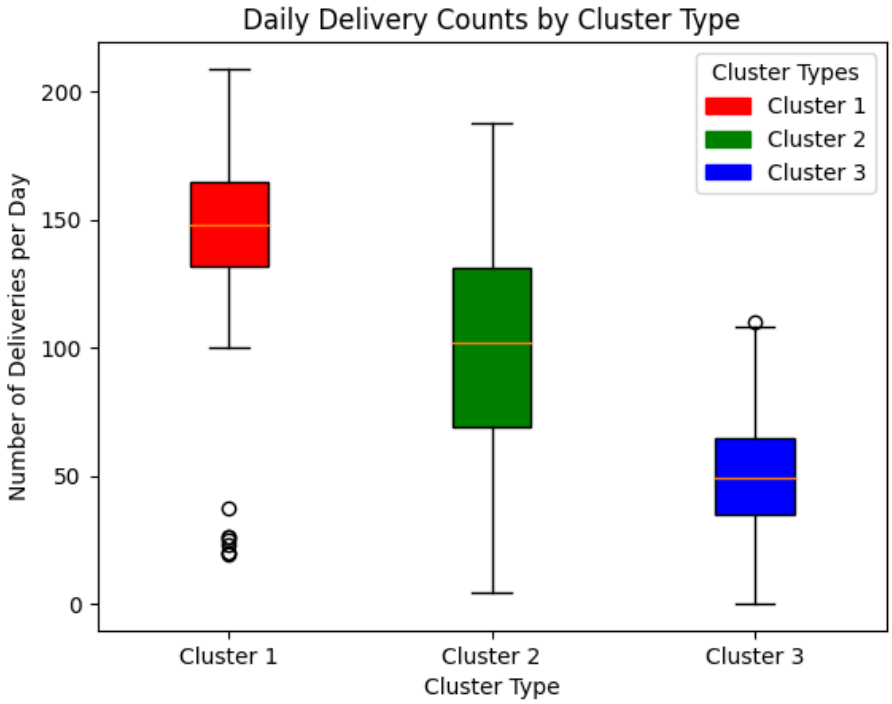

Figure 4.7: Daily Delivery Counts by Cluster Type

- **Median Delivery Counts**: Cluster 1 exhibits the highest median daily delivery count, indicating a high volume of deliveries likely due to shorter delivery distances, allowing more deliveries per day. Cluster 2 shows a moderate median, suggesting a balanced workload for mid-range distances, while Cluster 3 has the lowest median, aligning with expectations for longer delivery distances.
- **Distribution Spread and Variability**:
  - **Cluster 1** has a relatively narrow interquartile range (IQR), indicating consistent daily delivery counts. However, several low outliers suggest occasional days with lower

delivery volumes, possibly due to fluctuating demand or operational limitations in high-density zones.

  - **Cluster 2** exhibits the widest IQR, reflecting substantial variability in delivery counts. This broad range could be due to mid-range distances, where daily delivery numbers fluctuate more with varying demand.
  - **Cluster 3** shows moderate variability, with a lower range of delivery counts. This aligns with the nature of long-distance deliveries, where delivery counts are consistently low due to extended travel times.

- **Outliers**: - Cluster 1 has multiple low outliers, potentially indicating days of limited driver availability or lower demand in short-distance regions. - Cluster 3 has a high outlier, possibly signifying a rare surge in demand in long-distance zones, even though such occurrences are infrequent.

These insights into daily delivery patterns highlight the operational differences across cluster types. Cluster 1 may benefit from higher driver allocation due to its high delivery volume potential. Cluster 2's variable counts suggest a need for flexible scheduling, while Cluster 3's lower, consistent demand could be managed by optimized route planning for long distances.

Understanding these patterns is essential for improving resource allocation, optimizing driver workload, and adapting to demand patterns unique to each cluster type.

4.2.5 Impact of Cluster Size on Delivery Assignments

The size of a cluster directly influences the number of deliveries that can be assigned to a driver. As the cluster type increases from small to large, the average Euclidean distance between delivery points also increases, necessitating more travel within the cluster to complete the route.

To systematically determine the maximum number of deliveries each driver can handle, a framework was developed that considers both the driver's distance to the warehouse and the type of cluster they are assigned to. This framework adjusts the delivery capacity based on the cluster's classification (small, medium, or large) and the driver's proximity to the warehouse.

The maximum number of deliveries a driver can handle is based on an average value of 110 deliveries. Initially, this number was estimated after an informal inquiry with a driver, who reported handling around 80 deliveries. However, this figure was subsequently increased to 110 to accommodate the downward adjustments that would later result from the formula.

The maximum number of deliveries is calculated as follows:

- **Distance to Warehouse Adjustment:** The base number of deliveries starts at 110, which is then adjusted by subtracting a factor proportional to the driver's distance from the warehouse. Specifically, the adjustment is $0.085 \times$ distance_to_warehouse.

- **Cluster Type Adjustment:** The adjusted delivery capacity is further modified based on the cluster type:

- **Small Clusters**: Drivers can handle the full adjusted delivery capacity, calculated as $\text{round}(110 - 0.085 \times \text{distance\_to\_warehouse})$.
- **Medium Clusters**: Drivers can manage slightly fewer deliveries, at 95% of the adjusted capacity.
- **Large Clusters**: Drivers are assigned 85% of the adjusted capacity due to the greater distances between delivery points.

This relationship is summarized in the following calculation:

$$\text{Max Deliveries} = \begin{cases} \text{round}(110 - 0.085 \times \text{distance\_to\_warehouse}) & \text{for small clusters} \\ \text{round}(0.95 \times (110 - 0.085 \times \text{distance\_to\_warehouse})) & \text{for medium clusters} \\ \text{round}(0.85 \times (110 - 0.085 \times \text{distance\_to\_warehouse})) & \text{for large clusters} \end{cases}$$

This approach ensures that each driver is assigned a workload that is feasible given the cluster size and their location relative to the warehouse, thereby optimizing delivery efficiency and service quality.

4.3 Derived Dataset from the EDA

The table presented provides a detailed overview of the clustering analysis performed on delivery locations using the k-means algorithm. By analyzing the geographic positions (latitude and longitude) of the delivery points, 30 distinct clusters were identified, each represented by its mean position (X and Y coordinates). This clustering approach allows for a more structured view of delivery regions, aiding in efficient routing and resource allocation. Each cluster is further analyzed to compute the average and maximum distances from the centroid (mean position) to individual delivery points. These distances serve as the basis for assigning a target value to each cluster, categorizing them into three distinct groups. This categorization helps in prioritizing resources, with clusters having a higher average distance indicating more spread-out delivery points, requiring additional logistical attention.

Additionally, the distance from each cluster's centroid to a central warehouse is calculated, providing crucial insights into the overall efficiency of delivery operations. Assigning specific driver IDs to each cluster ensures accountability and enables a streamlined approach to managing deliveries. This data-driven methodology allows for optimization in terms of travel distance, fuel consumption, and delivery times, while also offering the flexibility to adapt delivery strategies based on the geographical characteristics of each cluster. The k-means clustering, combined with the distance metrics, ultimately supports operational decision-making by identifying areas that need more attention due to their distance from the warehouse or their internal delivery dispersion.

Cluster	Mean Position X	Mean Position Y	Target	Distance to Warehouse	Driver ID
1	-8.624443	41.171397	1	74.870712	1
2	-8.722956	41.324867	3	107.543580	2
3	-8.601231	41.090119	2	157.370757	3
4	-8.486279	41.194720	3	184.397347	4
5	-8.586355	41.187931	1	92.169191	5
6	-8.502789	41.060574	3	238.030822	6
7	-8.658511	41.251604	2	20.038767	7
8	-8.645663	41.193977	1	44.524293	8
9	-8.601807	41.122011	1	128.751891	9
10	-8.609769	41.263406	2	64.327968	10
11	-8.519368	41.144054	2	172.154580	11
12	-8.630836	41.005142	2	230.914928	12
13	-8.755393	41.416731	3	203.817552	13
14	-8.556085	41.103658	2	170.332401	14
15	-8.541633	41.178790	2	136.327796	15
16	-8.418336	41.062883	3	301.111118	16
17	-8.531880	41.236821	2	134.768237	17
18	-8.558063	41.213299	1	110.360570	18
19	-8.660146	41.161653	1	71.908131	19
20	-8.707016	41.244343	3	41.904005	20
21	-8.639575	41.122605	2	113.918518	21
22	-8.663997	41.312475	3	79.246729	22
23	-8.565283	41.153725	2	128.814205	23
24	-8.750902	41.372821	2	163.036449	24
25	-8.633938	41.058560	2	177.738147	25
26	-8.684971	41.191860	2	45.302275	26
27	-8.611528	41.227223	1	55.404740	27
28	-8.570173	41.045490	3	211.092298	28
29	-8.667274	41.395355	3	162.085525	29
30	-8.602589	41.155415	1	100.792870	30

Table 4.2: Cluster data with mean positions, target, distance to warehouse, and driver ID.

4.4 Modelling the problem as a Markov decision process

In this section, we model the logistics problem as a Markov Decision Process (MDP). This approach allows us to formally define and analyze the system of assigning deliveries to drivers while accommodating the dynamic nature of demand and logistical constraints. By framing the problem in terms of states, actions, and rewards, we can leverage MDP techniques to optimize delivery assignments and manage operational efficiency.

4.4.1 Definition of the problem

The problem consists of having multiple drivers available for the day along with multiple deliveries. We should build logistic sectors, and maintain a consistent sector for each driver throughout the days. This enables each driver to know each area better, therefore becoming faster and more comfortable with the areas of the deliveries.

However, these areas need to adapt to daily changes in demand. This means the logistic sectors must adjust to fluctuations in daily deliveries, sometimes increasing in size and sometimes decreasing. This adaptability is the core of the problem, along with addressing overfitting and stress scenarios, which will be tested later.

This requires considering restrictions such as the number of deliveries per driver, especially since some drivers are responsible for zones farther from the warehouse. This distance results in longer trips from the warehouse to their designated zones and back. Additionally, there is a criterion for determining when an episode ends: either when it is no longer possible to assign any more deliveries to drivers, or when all deliveries have been assigned. This can only occur if all assigned deliveries are fewer than the maximum possible deliveries for the day.

4.4.2 Modelling

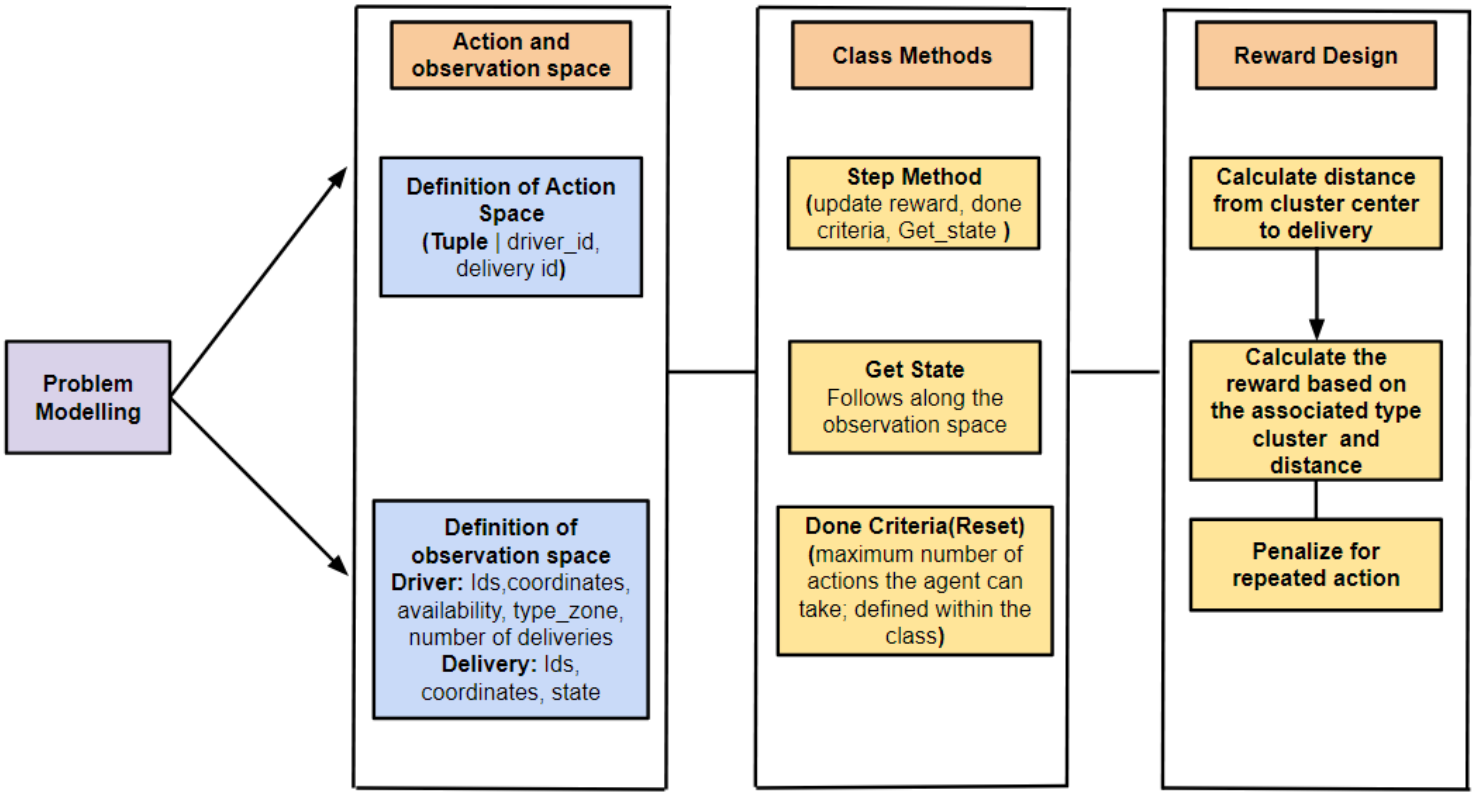

Figure 4.8: Problem Modelling

4.4.3 Problem Formulation

A Markov Decision Process (MDP) is formally described by a 4-tuple (S, A, P, R), which encapsulates the essential elements for modeling decision-making scenarios in environments with stochastic transitions. The MDP framework provides a structured approach to optimizing decision-making policies by defining the components involved in the decision process.

In this context, the nomenclature in Table 4.3 refers to the specific symbols and their meanings used in describing the MDP for the delivery assignment problem. Each symbol represents a crucial element of the MDP:

Symbol	Description
N_d	Number of delivery IDs
N_{drivers}	Number of available drivers
S	State space, capturing the status and attributes of drivers and deliveries
S_t	State at time t
S_{t+1}	Resulting state at time $t+1$
a	Action as a pair (d,i), where d is the driver ID and i is the delivery ID
A	Action space, set of all possible actions
$\mathbf{d}_{\text{ids}}$	Vector of driver IDs
$\mathbf{D}_{\text{coords}}$	Matrix of driver coordinates (x, y)
$\mathbf{d}_{\text{avail}}$	Vector of driver availability (1 if available, 0 otherwise)
$\mathbf{d}_{\text{zone}}$	Vector of driver type zones
$\mathbf{d}_{\text{num}}$	Vector of the number of deliveries assigned to each driver
$\mathbf{l}_{\text{ids}}$	Vector of delivery IDs
$\mathbf{L}_{\text{coords}}$	Matrix of delivery coordinates (x, y)
$\mathbf{l}_{\text{state}}$	Vector of delivery states (1 if active, 0 otherwise)
$P(s' \mid s,a)$	Probability of transitioning from state s to s' after action a
Pr	Denotes probability

Table 4.3: Summary of nomenclature used in the MDP formulation for the delivery assignment problem.

4.4.4 Action space

In this work, the action space A represents all possible actions that can be taken to assign delivery IDs to driver IDs. Given the finite number of delivery IDs and available drivers, each action defines a specific assignment of a delivery to a driver.

In mathematical terms, the action space A can be described as the set of all possible pairs (d,i), where d is a driver ID and i is a delivery ID.

Considering that N_d is the number of delivery IDs and N_{drivers} is the number of available drivers, an action $a \in A$ is represented as a pair (d,i) such that $d \in \{1,2,\ldots,N_{\text{drivers}}\}$ and $i \in \{1,2,\ldots,N_d\}$.

- Each action (d,i) assigns delivery ID i to driver d.
- This action space covers all possible assignments of deliveries to drivers.

For example, if $N_d = 5$ and $N_{\text{drivers}} = 3$, the action space A includes pairs such as:

$$A = \{(1,1),(1,2),(1,3),(1,4),(1,5),(2,1),(2,2),(2,3),(2,4),(2,5),(3,1),(3,2),(3,3),(3,4),(3,5)\}$$

This means that an action could be, for instance, $(1,3)$, assigning delivery 3 to driver 1, or $(2,4)$, assigning delivery 4 to driver 2. The complete action space encompasses all possible such pairings.

4.4.5 State space

In the context of our delivery assignment problem, we need to formalize the state space and action space to effectively use a Markov Decision Process (MDP).

The state space S consists of various components that represent the status and attributes of both drivers and deliveries. Each state $s \in S$ can be described by the following elements:

- **Driver Information:**
  - driver_ids: Vector of driver IDs, $\mathbf{d}_{\text{ids}} \in \mathbb{Z}^{N_{\text{drivers}}}$.
  - driver_coords: Matrix of driver coordinates (x, y), $\mathbf{D}_{\text{coords}} \in \mathbb{R}^{N_{\text{drivers}} \times 2}$.
  - driver_availability: Vector of driver availability (1 if available, 0 otherwise), $\mathbf{d}_{\text{avail}} \in \mathbb{Z}^{N_{\text{drivers}}}$.
  - driver_type_zone: Vector of driver type zones (1, 2, or 3), $\mathbf{d}_{\text{zone}} \in \mathbb{Z}^{N_{\text{drivers}}}$.
  - driver_number_of_deliveries: Vector of the number of deliveries assigned to each driver, $\mathbf{d}_{\text{num}} \in \mathbb{Z}^{N_{\text{drivers}}}$.
- **Delivery Information:**
  - delivery_ids: Vector of delivery IDs, $\mathbf{l}_{\text{ids}} \in \mathbb{Z}^{N_d}$.
  - delivery_coords: Matrix of delivery coordinates (x, y), $\mathbf{L}_{\text{coords}} \in \mathbb{R}^{N_d \times 2}$.
  - delivery_state: Vector of delivery states (1 if active, 0 otherwise), $\mathbf{l}_{\text{state}} \in \mathbb{Z}^{N_d}$.

In summary, the state space S can be expressed as:

$$S = (\mathbf{d}_{\text{ids}}, \mathbf{D}_{\text{coords}}, \mathbf{d}_{\text{avail}}, \mathbf{d}_{\text{zone}}, \mathbf{d}_{\text{num}}, \mathbf{l}_{\text{ids}}, \mathbf{L}_{\text{coords}}, \mathbf{l}_{\text{state}})$$

where:

$$\mathbf{d}_{\text{ids}} \in \mathbb{Z}^{N_{\text{drivers}}}$$
$$\mathbf{D}_{\text{coords}} \in \mathbb{R}^{N_{\text{drivers}} \times 2}$$
$$\mathbf{d}_{\text{avail}} \in \mathbb{Z}^{N_{\text{drivers}}}$$
$$\mathbf{d}_{\text{zone}} \in \mathbb{Z}^{N_{\text{drivers}}}$$
$$\mathbf{d}_{\text{num}} \in \mathbb{Z}^{N_{\text{drivers}}}$$
$$\mathbf{l}_{\text{ids}} \in \mathbb{Z}^{N_d}$$
$$\mathbf{L}_{\text{coords}} \in \mathbb{R}^{N_d \times 2}$$
$$\mathbf{l}_{\text{state}} \in \mathbb{Z}^{N_d}$$

This comprehensive state space captures all the necessary details about drivers and deliveries for effective decision-making in the delivery assignment problem.

4.4.6 Reward

The reward system for drivers is based on the Euclidean distance from the cluster center to each delivery ID, using their coordinates. The Euclidean distance d between the cluster center (x_c, y_c) and the delivery ID (x_d, y_d) is calculated using the formula:

$$d = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2} \tag{4.1}$$

This distance is then multiplied by 1000 to scale the value for practical use.

Once the Euclidean distance is determined, it is used in the appropriate reward function. There are three types of reward functions corresponding to three different types of clusters: small, medium, and large. The Euclidean distance is substituted into Equation 4.2 for its respective cluster type.

4.4.6.1 Reward for cluster types

The reward functions are designed to ensure **Compactness**—the ability of a sector to contain only delivery IDs, ideally assigned to the responsible driver, with minimal noise (i.e., deliveries intended for other drivers). This approach minimizes driving distances and forms well-defined clusters. The reward functions involve three key parameters, refined through trial and error to match the desired design. To support this, **GeoGebra** was used to visualize the reward functions and assess how the parameters impact and drive adjustments in reward distribution, helping clarify how factors like distance from the center influence penalties. These parameters are adjusted for small, medium, and large zones to account for variations in delivery density and adaptability requirements.

The general reward function is defined as:

$$R(x) = Ae^{-Bx} - \log_C(x+1) \tag{4.2}$$

where:

- A is the coefficient of the exponential function, determining the initial reward value when $x = 0$.
- B is the coefficient determining the rate at which the reward decreases with distance.
- C is the base value of the logarithmic function, indicating the penalty slope for exceeding the area limit.
- X is the distance previously calculated from the center of the driver associated to the delivery and its coordinates.

Specific reward functions for each cluster type are as follows:

- **Small areas**: These areas have a higher concentration of deliveries, meaning the cluster center is relatively close to the delivery points. As such, small areas are less adaptable which means they should not expand that much. They should also have a steep penalty slope for deviations from the border. The corresponding reward function can be defined as follows:

$$S(x): \quad 50e^{-0.032x} - \log_{1.2}(x+1) \tag{4.3}$$

- **Medium areas**: Medium zones share more characteristics with small zones than large ones. They also have a higher delivery density, leading to a border limit closer to the cluster center, though slightly more adaptable than small areas.

$$M(x): \quad 50e^{-0.042x} - \log_{1.5}(x+1) \tag{4.4}$$

- **Large areas**: Larger zones are more sensitive to changes in daily delivery demand, requiring a greater border extent. The penalty slope for large areas is small, reflecting the need for higher adaptability and less severe penalties for exceeding the defined borders.

$$B(x): \quad 80e^{-0.056x} - \log_{6}(x+1) \tag{4.5}$$

- **Penalty for repeated delivery IDs**: For all cluster types, the repetition of a delivery ID incurs an additional penalty of -200 to discourage inefficiency and ensure each delivery is performed only once.

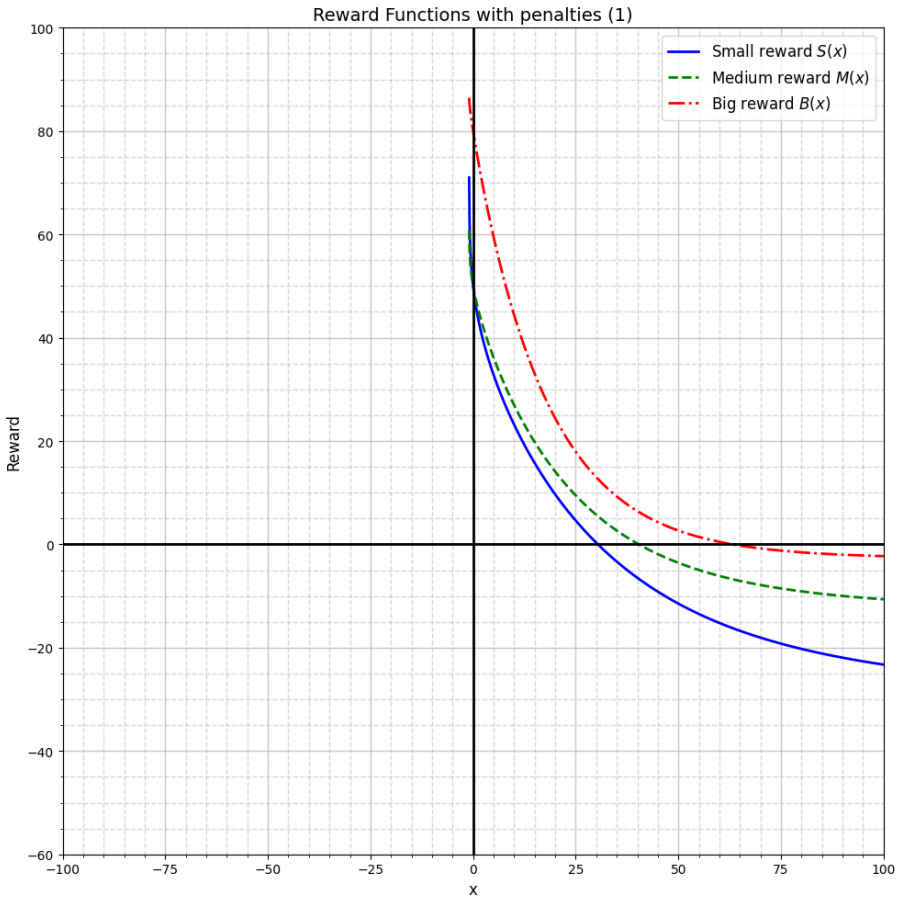

Figure 4.9: Reward functions with penalties

In addition to the previously defined reward functions, an alternative set of reward functions has been introduced for small, medium, and large zones. These new functions aim to enhance flexibility by reducing the penalty for exceeding the area limits. This makes them suitable for situations where adaptability is prioritized over strict border constraints.

The new reward functions are defined as follows:

- **Small areas**: Here, the reward function provides a higher initial reward and decreases at a slower rate compared to the first reward function. The absence of a steep penalty allows small zones more flexibility in expansion.

$$S_2(x): \quad 100e^{-0.05x} - \log_{15}(x+1) + 3 \tag{4.6}$$

- **Medium areas**: Similar to the small areas, this function provides a higher reward coefficient and a less aggressive penalty. This encourages medium zones to adapt more freely to delivery density changes.

$$M_2(x): \quad 150e^{-0.04x} - \log_{15}(x+1) + 3 \tag{4.7}$$

- **Large areas**: The reward function for large zones also follows the same pattern, with an even slower decay and minimal penalty. This is designed to accommodate large zones that require significant flexibility due to variability in delivery demand.

$$B_2(x): \quad 200e^{-0.03x} - \log_{15}(x+1) + 3 \tag{4.8}$$

The key **difference between the first and second reward functions** is the penalty approach. In the second set, the penalty term, $-\log_{15}(x+1)$, features a larger base value and a consistent additive term of $+3$. This means **penalties for deviations are much less severe**, thereby allowing clusters to expand their borders with fewer restrictions. This makes the second reward function set particularly useful in contexts where delivery zones need greater adaptability to accommodate fluctuations in demand.

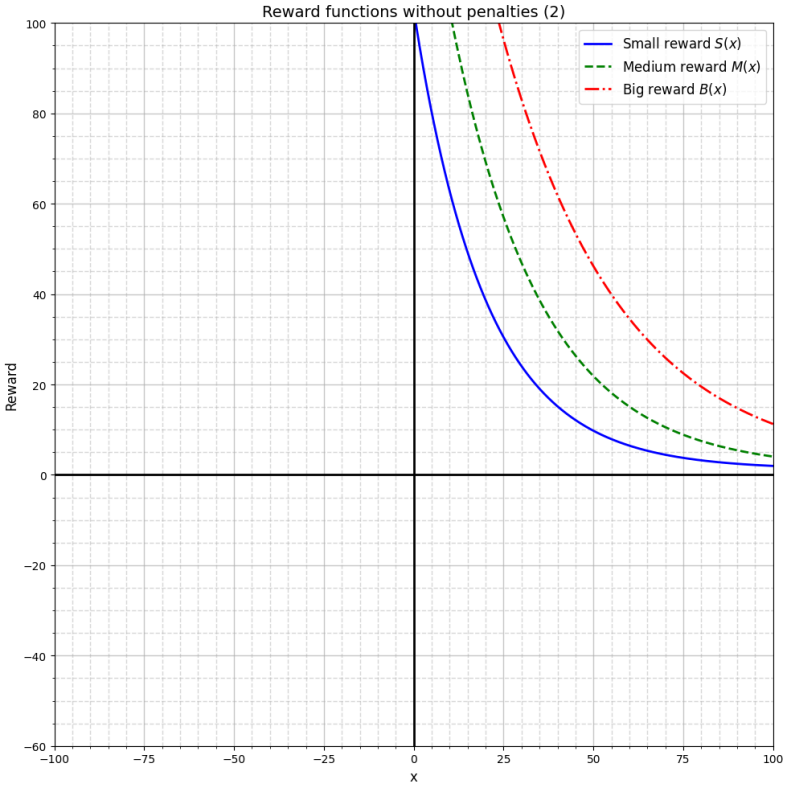

Figure 4.10: Reward functions with penalties

4.5 Summary

In this chapter, we conducted an exploratory data analysis to understand the distribution of deliveries across various geographical regions. By employing the k-means clustering algorithm, we identified 30 distinct clusters representing the maximum number of available drivers. The clustering approach enabled us to pinpoint central points for each cluster, thereby facilitating the assignment of deliveries to specific drivers, aiming to maintain consistent and adaptable areas of operation.

We formalized the problem as a Markov Decision Process (MDP), defining the state space, action space, and transition probabilities. The state space consists of mapping delivery IDs to driver IDs, while the action space involves assigning delivery IDs to drivers. Transition probabilities capture the dynamics of how states evolve in response to actions. Our reward structure is based on the Euclidean distance from cluster centers to delivery points, with specific reward functions tailored for small, medium, and large clusters, ensuring a balanced and fair workload distribution among drivers.

Constraints were implemented to guarantee the uniqueness of delivery assignments and to maintain an equitable workload for drivers. The goal is to adapt logistical sectors to daily changes in delivery demand while avoiding overburdening any single driver. This approach helps ensure efficient and effective delivery management, optimizing both the drivers' performance and the overall delivery process.

Chapter 5

Results

5.1 Performance Results

Compatible Models and Results

The Stable Baselines3 library supports several reinforcement learning algorithms that are compatible with our defined action and observation spaces. The action space in our problem is a `MultiDiscrete` space, and the observation space is a `Dict` space. Based on these specifications, the following algorithms from Stable Baselines3 are suitable for our needs:

- **Proximal Policy Optimization (PPO)**: PPO is an on-policy algorithm that can handle both discrete and continuous action spaces, including multi-discrete action spaces. It is robust and widely used for various reinforcement learning tasks.
- **Advantage Actor-Critic (A2C)**: A2C is another on-policy algorithm capable of handling both discrete and continuous action spaces. It is particularly suitable for environments with multi-discrete action spaces.

Other algorithms such as Deep Q-Network (DQN), Soft Actor-Critic (SAC), Twin Delayed DDPG (TD3), and Distributional Reinforcement Learning with Quantile Regression (QR-DQN) either do not support multi-discrete action spaces or are designed for continuous action spaces, making them less suitable for our specific problem.

5.1.1 Experiment Design

The experiments were designed to assess the performance of PPO and A2C algorithms under different configurations. The key variables manipulated in these experiments include the **number of actions**, **total timestamps**, **number of episodes**, and the **reward function**. The variations were aimed at evaluating how each algorithm adapts to changes in the action space complexity and the training environment's scale.

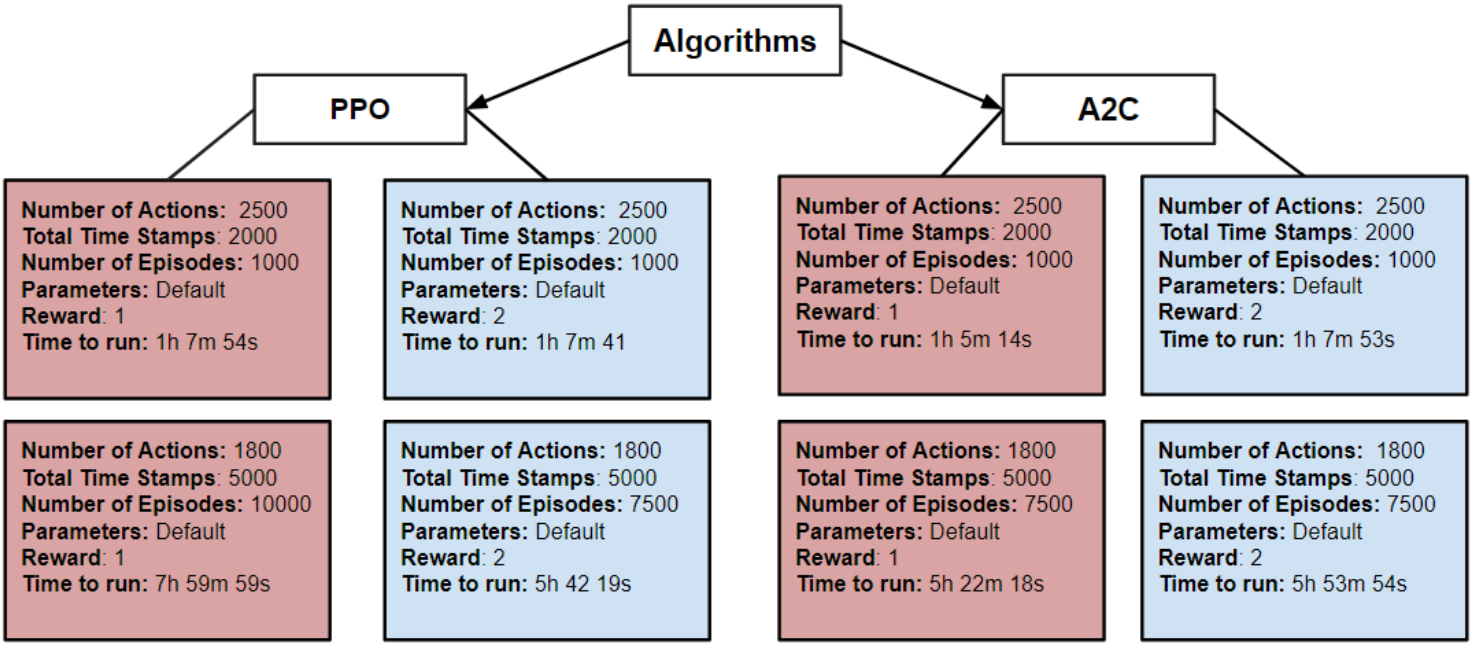

Figure 5.1: Diagram of PPO and A2C Experiments

5.1.2 Analysis with Plots

To gain deeper insights into the performance of the algorithms during training, two types of plots were generated for analysis:

- **Reward Plot:** This plot keeps track of the cumulative reward obtained over the episodes. It helps in visualizing how quickly the algorithm converges to an optimal policy and how stable the learned policy is over time.
- **Repeated Actions Plot:** This plot counts the number of repeated actions per episode, which provides insight into the exploration-exploitation trade-off made by the algorithm. A high number of repeated actions may indicate that the agent is getting stuck in local optima or over-exploiting a particular strategy.

These plots are crucial for evaluating the learning behavior of the algorithms, as they highlight both the stability and adaptability of the policies developed by PPO and A2C. The reward plot

helps identify the efficiency of reward accumulation, while the repeated actions plot offers a deeper look into the agent's decision-making process throughout the episodes.

5.1.3 Motivation Behind Experiment Configurations

We strategically adjusted the number of actions to explore the trade-offs between action complexity and the training duration. For example, in some setups, the number of actions was reduced to allow an increase in the total timestamps. This modification aimed to give the algorithms more opportunities to explore strategies over a more extended period, potentially leading to improved learning outcomes.

Additionally, the number of episodes was significantly increased in some scenarios, allowing the algorithms to engage in a more extensive training process, thereby enabling a deeper exploration of the environment. The variations in reward values were intended to analyze the impact of different incentive structures on each algorithm's performance.

5.1.4 Experiment Details

- **PPO Experiments:**
  - **Experiment 1:** 2500 actions, 2000 timestamps, 1000 episodes, reward value of 1. *Time to run: 1 hour 7 minutes and 54 seconds.*
  - **Experiment 2:** 1800 actions, 5000 timestamps, 10000 episodes, reward value of 1. *Time to run: 7 hour 59 minutes and 59 seconds.*
  - **Experiment 3:** 2500 actions, 2000 timestamps, 1000 episodes, reward value of 2. *Time to run: 1 hour 7 minutes and 41 seconds.*
  - **Experiment 4:** 1800 actions, 5000 timestamps, 7500 episodes, reward value of 2. *Time to run: 5 hour 42 minutes and 19 seconds.*
- **A2C Experiments:**
  - **Experiment 1:** 2500 actions, 2000 timestamps, 1000 episodes, reward value of 1. *Time to run: 1 hour 5 minutes and 14 seconds.*
  - **Experiment 2:** 1800 actions, 5000 timestamps, 7500 episodes, reward value of 1. *Time to run: 5 hour 22 minutes and 14 seconds.*
  - **Experiment 3:** 2500 actions, 2000 timestamps, 1000 episodes, reward value of 2. *Time to run: 1 hour 7 minutes and 53 seconds.*
  - **Experiment 4:** 1800 actions, 5000 timestamps, 7500 episodes, reward value of 2. *Time to run: 5 hour 53 minutes and 54 seconds.*

5.1.5 Experiment Goals

The primary goal of these experiments is to understand how the PPO and A2C algorithms handle changes in the action space complexity and the effect of different reward structures. By systematically varying these parameters, we aim to identify the optimal configuration that leads to faster convergence and more stable performance in our reinforcement learning environment.

5.1.6 Proximal Policy Optimization Results

In this section, we present the initial combination of parameters and environment settings used in our experiments. Below, we summarize the default configuration that was employed across all experiments in Table 5.1. This configuration played a crucial role in shaping the agent's training and performance outcomes.

Parameter	**Value**
`approx_kl`	0.011718975
`clip_fraction`	0.137
`clip_range`	0.2
`entropy_loss`	-11.4
`explained_variance`	5.96e-08
`learning_rate`	0.0003
`loss`	1.1e+04
`n_updates`	10
`policy_gradient_loss`	-0.0476
`value_loss`	2.26e+04

Table 5.1: Training Parameters and Values

Following the visual data that illustrates the evolution of the agent's reward over the course of training, we provide a detailed analysis explaining why the agent did not achieve significant learning progress.

5.1.6.1 Experiment 1

The following plots illustrate the evolution of the agent's learning behavior over the course of training with Reward (1):

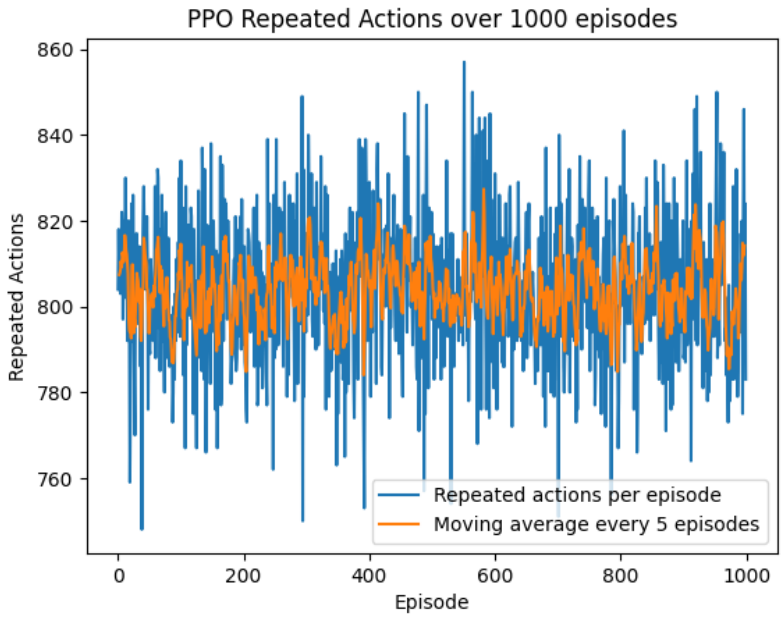

Figure 5.2: PPO Repeated Actions over 1000 Episodes

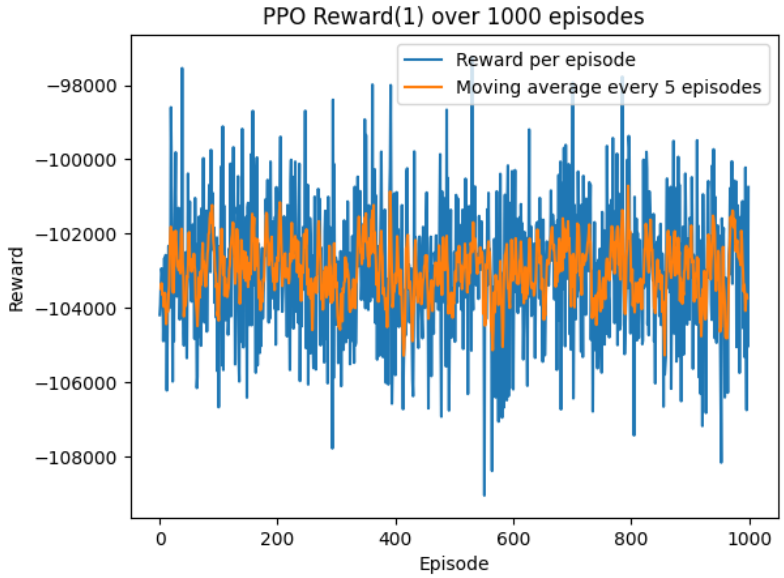

Figure 5.3: PPO Reward (1) over 1000 Episodes

In Figure 5.2, the stable pattern of repeated actions suggests consistent exploration but reveals minimal optimization in action selection. The cumulative reward in Figure 5.3 fluctuates without

a significant upward trend, indicating that the agent struggles to converge on an optimal policy under Reward (1).

5.1.6.2 Experiment 3

With Reward (2), similar trends are observed:

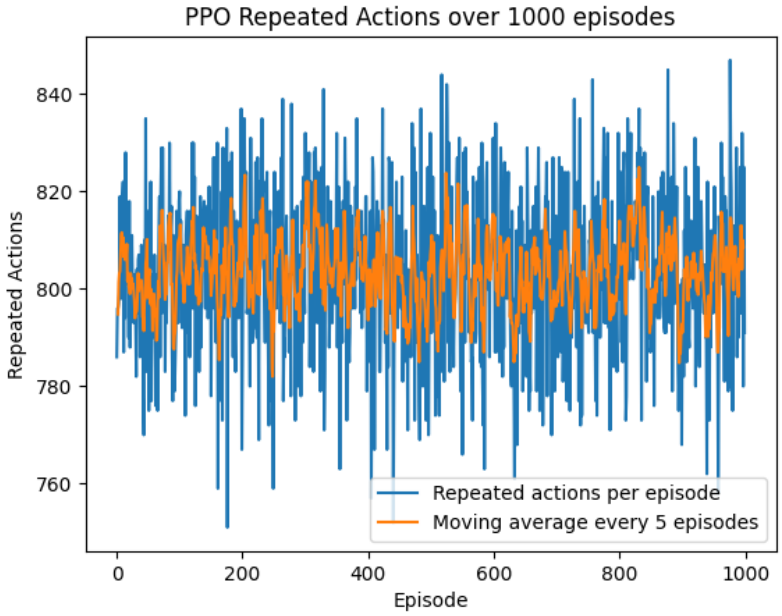

Figure 5.4: PPO Repeated Actions over 1000 Episodes with Reward (2)

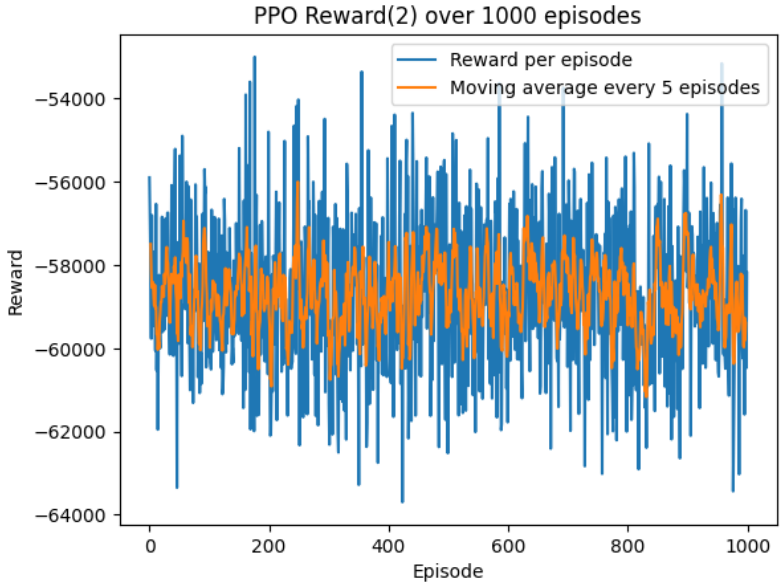

Figure 5.5: PPO Reward (2) over 1000 Episodes

As shown in Figure 5.4 and Figure 5.5, both the repeated actions and reward trends remain largely unchanged. Despite the modified reward structure, learning progress remains limited, and cumulative rewards fail to display a consistent upward trend, underscoring challenges in reaching an optimal policy with Reward (2).

Analysis and Interpretation

Across both experiments, the agent demonstrates limited learning progress. The lack of a clear upward trend in cumulative rewards implies that the agent faces challenges in developing an optimal policy under either reward structure. Key observations include:

- **Parameter Suitability:** The chosen PPO parameters might not be well-suited to this environment, potentially restricting the agent's effectiveness. Tuning parameters specific to the complexity of the action space and reward dynamics could improve performance.
- **Exploration-Exploitation Balance:** The high variability in repeated actions per episode suggests that the agent may be over-exploring rather than effectively exploiting promising strategies, highlighting a possible need to fine-tune the balance between exploration and exploitation.
- **Action Space Complexity:** The large action space (2,500 actions) likely increases the difficulty of policy learning, adding to the cognitive load on the agent. Reducing action space complexity or adopting hierarchical actions might support better policy learning and stability.
- **Reward Structure Limitations:** Although Reward (2) was designed to encourage learning, its similarity in outcomes to Reward (1) suggests that both reward structures might lack sufficient incentive clarity to drive optimal learning behavior.

These findings underscore that both reward structures, as implemented, are inadequate to catalyze significant improvements in learning, suggesting a need for more refined reward designs in future work.

5.1.6.3 Experiment 2 and 4

These experiments tested the impact of increasing timesteps and total episodes while adjusting the reward structure. To simplify the learning process, the action space was reduced to 1,800 actions, with 5,000 timesteps per episode. Experiment 2 utilized Reward (1) over 10,000 episodes, while Experiment 4 applied Reward (2) over 7,500 episodes.

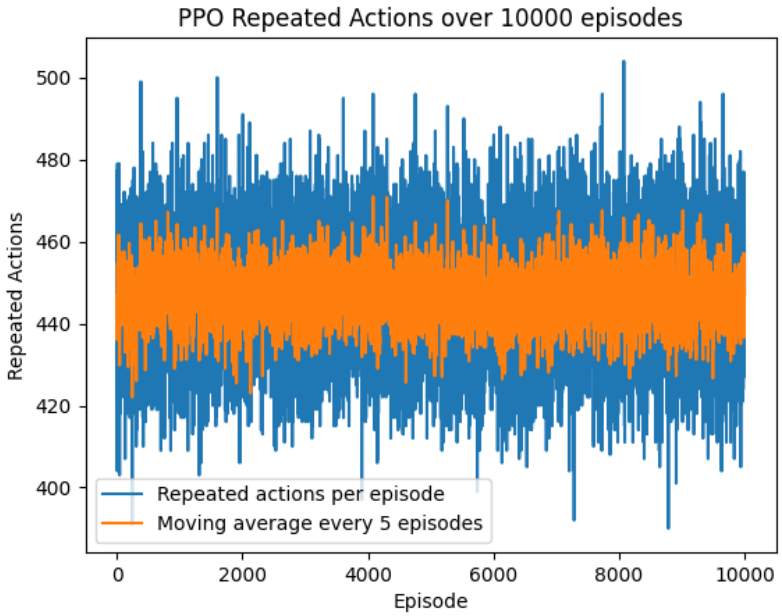

Figure 5.6: PPO Repeated Actions over 10,000 Episodes with Reward (1)

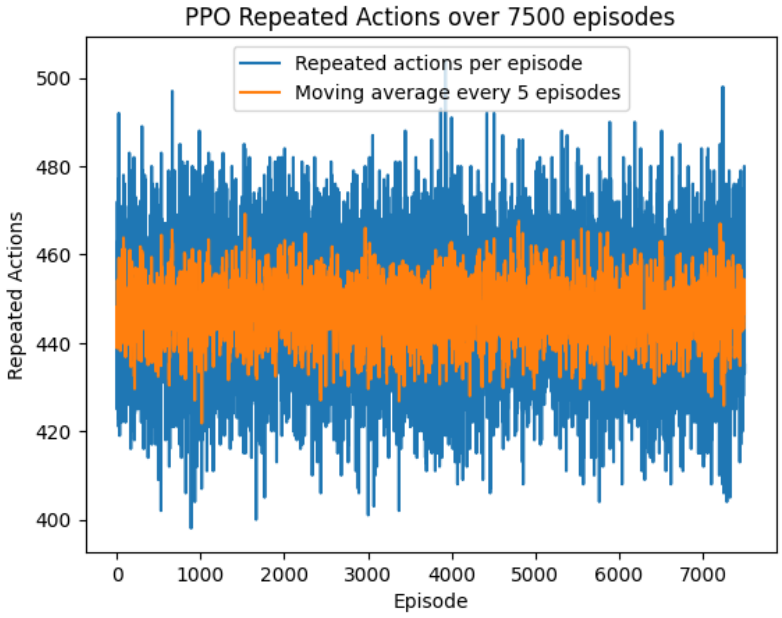

Figure 5.7: PPO Repeated Actions over 7500 Episodes with Reward (2)

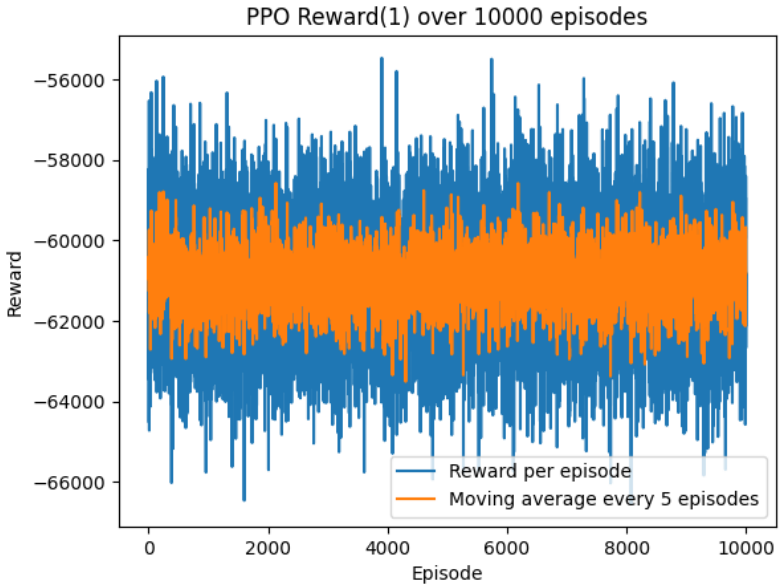

Figure 5.8: PPO Reward (1) over 10,000 Episodes

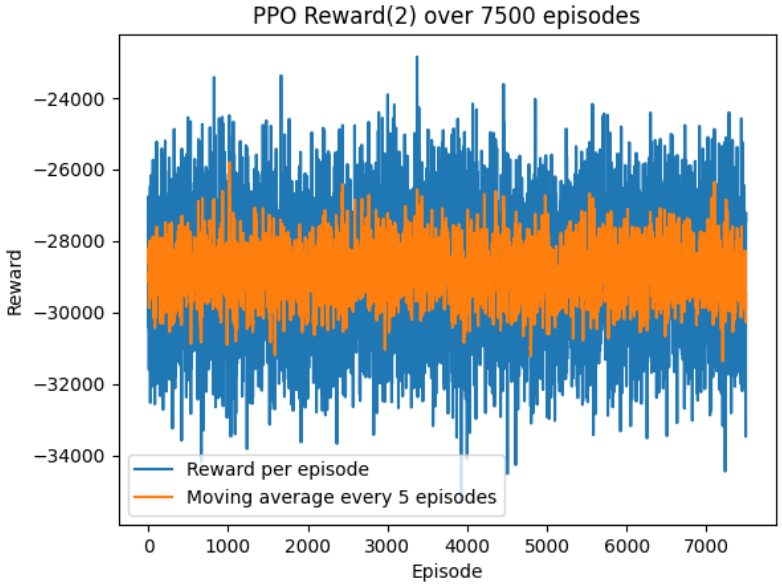

Figure 5.9: PPO Reward (2) over 7,500 Episodes

In Figure 5.6 and Figure 5.7, the variability in repeated actions indicates distinct exploration behaviors under each reward structure. With Reward (1), repeated actions cover a broader range, suggesting persistent exploration. In contrast, Reward (2) yields more consistent but still fluctuating actions, implying stability without significant optimization in selection.

The cumulative reward trends (Figure 5.8 and Figure 5.9) show limited improvement with either reward, with rewards fluctuating around a mean rather than trending upwards. This suggests

that neither reward structure effectively guides the agent toward optimal policy convergence.

Analysis and Interpretation

From these observations, we conclude:

- **Repeated Actions and Exploration:** The wider range of repeated actions with Reward (1) reflects extensive exploration, but without yielding stable, rewarding actions. Reward (2) slightly stabilizes this pattern, yet fails to guide the agent toward an optimal sequence of actions.
- **Effectiveness of Reward Structures:** Reward (2) was designed to improve learning, yet cumulative reward trends show limited benefits over Reward (1). Both rewards lack upward momentum, pointing to the need for refinement to better support learning objectives.
- **Comparative Performance of Rewards:** Reward (2) may be marginally more effective, as seen in the narrower range of repeated actions, yet this consistency does not translate to better cumulative performance, suggesting limited impact.
- **Recommendation for Reward Refinement:** Both rewards require re-evaluation to drive more effective learning. Future studies should consider alternative rewards that better balance exploration and exploitation. Approaches like reward shaping, incremental rewards, or milestone-based incentives could provide clearer guidance for the agent's learning path.

In summary, these findings highlight the limitations of the current reward structures in driving optimal learning outcomes. Future work could explore alternative strategies, including hierarchical rewards or agent adjustments, to enhance learning adaptability and efficiency in complex environments.

5.1.7 First Results: A2C

In this section, we present the initial combination of parameters and environment settings used in our experiments, which was conducted with the same configuration as PPO. Notably, the parameter values shown in Table 5.2 reflect those at the conclusion of the training process and may have undergone some adjustments by the model due to the total timesteps constraint.

Parameter	Value
fps	54
iterations	800
time_elapsed	73
entropy_loss	-0.218
explained_variance	0
learning_rate	0.0007
n_updates	799
policy_loss	1.35
value_loss	3.32e+03

Table 5.2: Final Training Metrics for A2C Model at 4000 Timesteps

5.1.7.1 Experiment 1 and 3

The following plots illustrate the evolution of the agent's learning behavior over the course of training using A2C under two different reward structures. The trends in repeated actions and cumulative rewards provide insights into the agent's exploration consistency and optimization challenges.

In Figure 5.10, the pattern of repeated actions demonstrates that while the A2C agent engages in consistent exploration, it lacks effective optimization in action selection. Additionally, the cumulative reward in Figure 5.11 shows fluctuations without a significant upward trend, indicating limited progress toward an optimal policy under Reward (1).

Under Reward (2), similar trends are observed:

As shown in Figure 5.12 and Figure 5.13, both the repeated actions and cumulative reward trends remain largely unchanged, even with the modified reward structure. This suggests that the A2C agent's learning progress is minimal, with no observable improvement in cumulative reward over time under Reward (2) either.

5.1.7.2 Experiment 2 and 4

In these experiments, we tested the impact of increasing the timesteps and total episodes while adjusting the reward structure. To reduce complexity and potentially improve the agent's learning process, the action space was reduced to 1800 actions, with the number of timesteps set at 5000. Experiment 2 used Reward (1) with 10,000 episodes, and Experiment 4 applied Reward (2) over 7500 episodes.

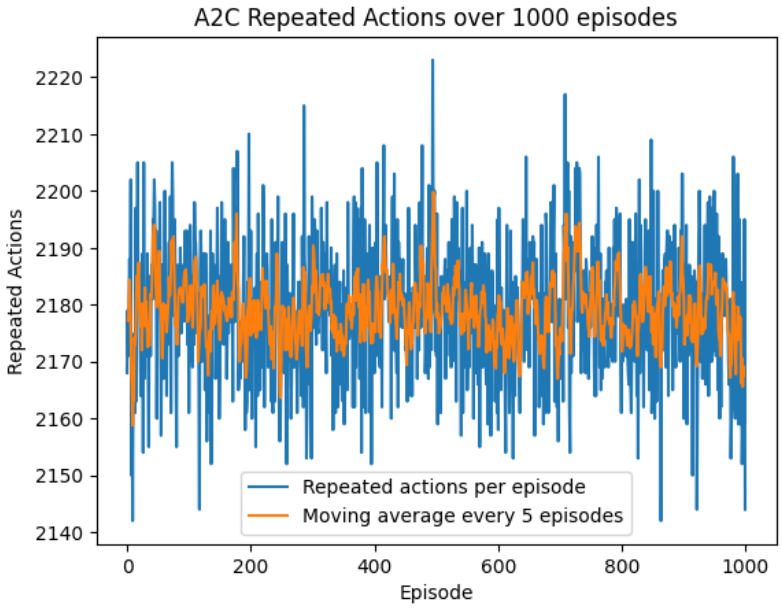

Figure 5.10: A2C Repeated Actions over 1000 Episodes with Reward (1)

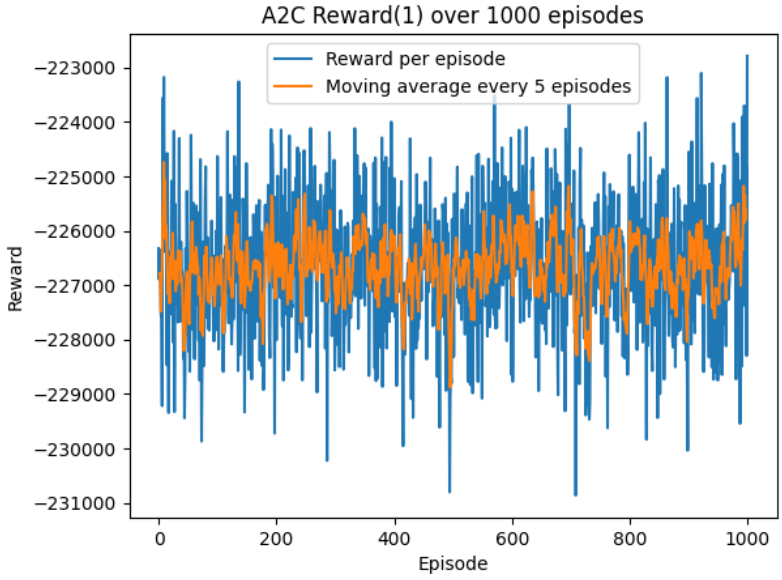

Figure 5.11: A2C Reward (1) over 1000 Episodes

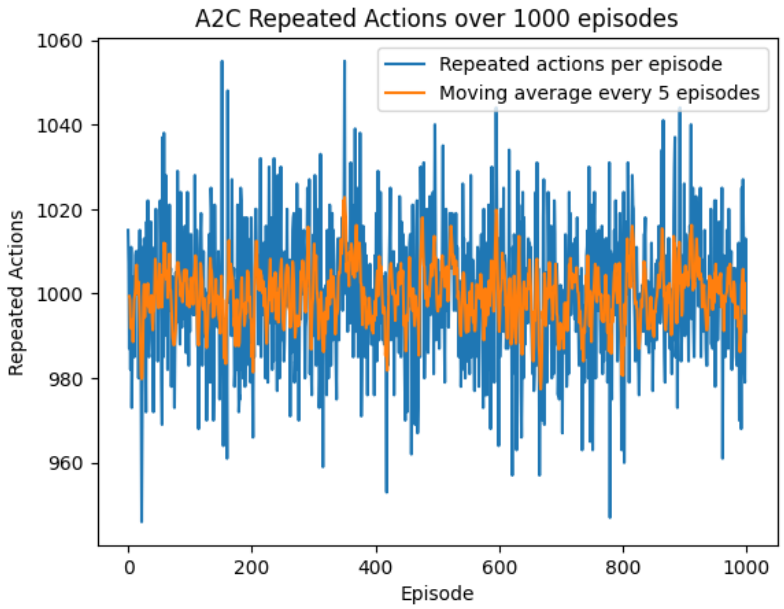

Figure 5.12: A2C Repeated Actions over 1000 Episodes with Reward (2)

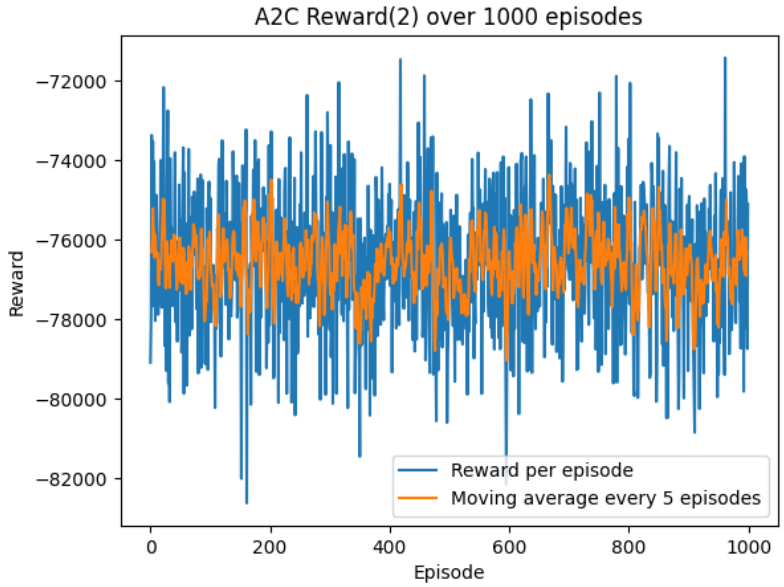

Figure 5.13: A2C Reward (2) over 1000 Episodes

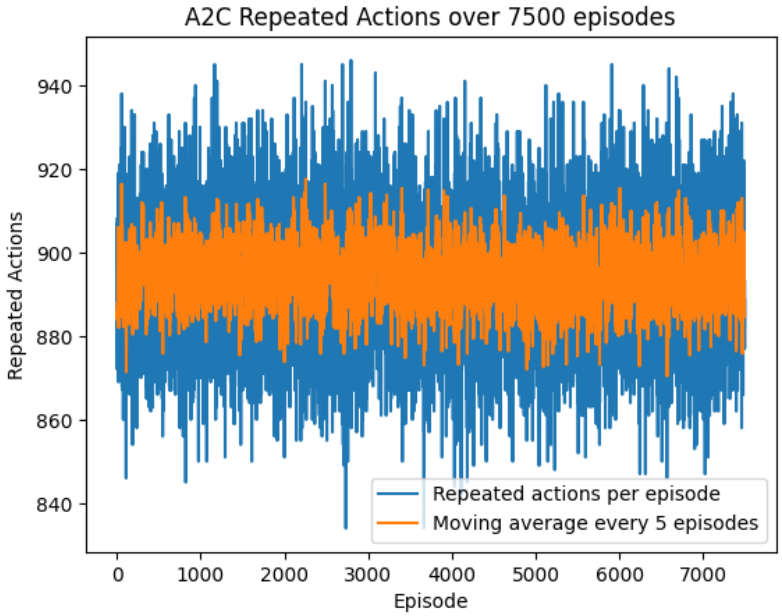

Figure 5.14: A2C Repeated Actions over 10,000 Episodes with Reward (1)

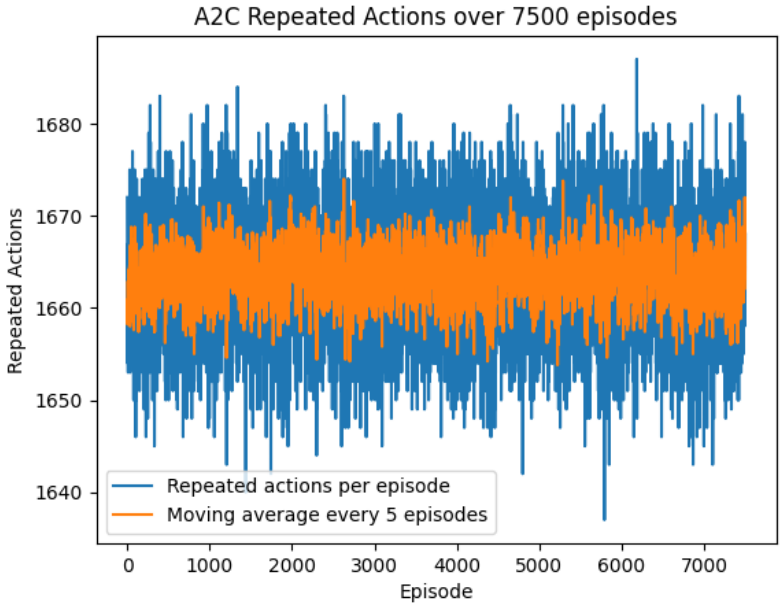

Figure 5.15: A2C Repeated Actions over 7500 Episodes with Reward (2)

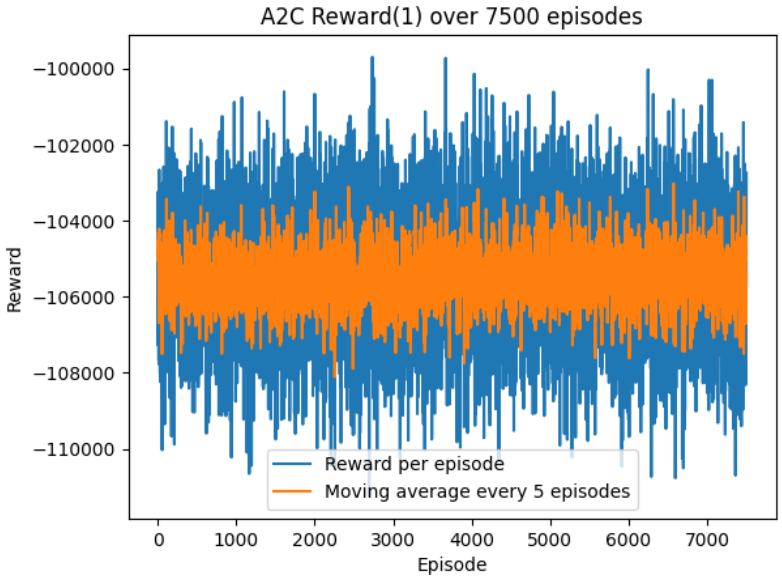

Figure 5.16: A2C Reward (1) over 10,000 Episodes

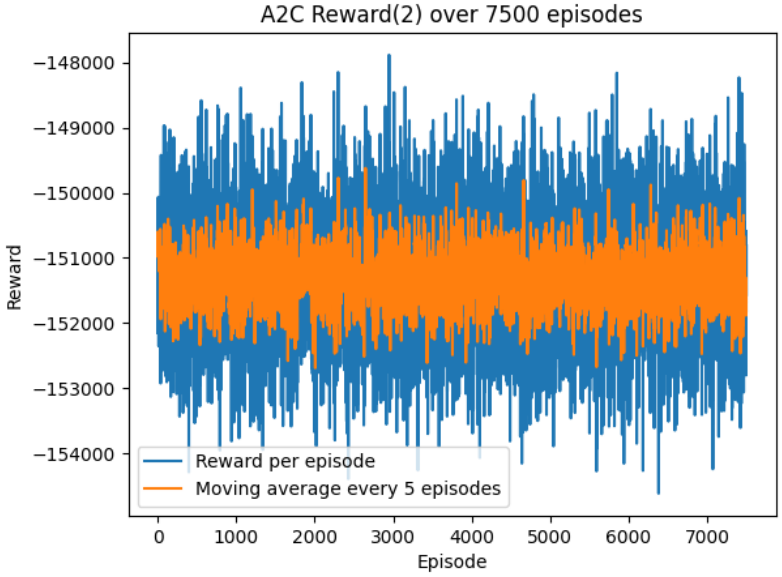

Figure 5.17: A2C Reward (2) over 7500 Episodes

In Figure 5.14 and Figure 5.15, the variability in repeated actions per episode reveals insights into the agent's behavior under each reward structure. With Reward (1), repeated actions span a wider range, suggesting more frequent exploration of different actions. In contrast, with Reward (2), the repeated actions are slightly more stable but still fluctuate significantly. This indicates that while Reward (2) led to more consistent repeated actions, the A2C agent did not fully exploit or optimize action selection.

Furthermore, the cumulative reward trends shown in Figure 5.17 indicate that Reward (2) does not promote substantial improvement over time. The reward fluctuates around a mean with no clear upward trend, suggesting that neither reward structure was effective in driving the A2C agent toward a consistently optimal policy.

Analysis and Interpretation

From these observations, we draw the following conclusions:

- **Repeated Actions and Exploration:** Reward (1) (Experiment 2) resulted in a wider variability of repeated actions, which may imply that the A2C agent is exploring more but struggling to find stable, rewarding actions. Reward (2) (Experiment 4), while showing less variability, still led to frequent repeated actions, indicating that neither reward structure successfully guided the agent toward optimal action sequences.

- **Effectiveness of Reward Structures:** The cumulative reward under Reward (2) (shown in Figure 5.17) did not show significant improvement over time, similar to Reward (1) in Experiment 2. Although Reward (2) was designed to incentivize more effective learning, its impact on overall performance appears limited. Neither reward structure produced an upward trend in cumulative reward, suggesting that both need further refinement to better support learning progress.

- **Comparative Performance of Reward (1) and Reward (2):** Although neither reward led to marked improvement, Reward (2) may be slightly more suitable for this environment, as it resulted in a narrower range of repeated actions, indicating more consistent behavior by the A2C agent. However, because the cumulative rewards did not improve significantly with Reward (2), this consistency did not translate into better overall performance.

- **Need for Reward Refinement:** Both rewards, in their current forms, seem insufficient to drive learning improvements. Future experiments should consider modifying the reward structure to better balance exploration and exploitation, potentially incorporating reward shaping techniques to more effectively guide the agent's behavior.

In conclusion, these results underscore the limitations of both reward structures in promoting optimal learning for the A2C agent. Future studies should explore alternative reward strategies or adjustments in agent architecture to enhance learning outcomes in this complex environment.

5.2 Overfitting: Stress Scenarios

To assess the robustness and adaptability of the reinforcement learning (RL) model in real-world applications, we subjected it to various stress scenarios that introduce unbalanced conditions compared to the training environment. The agent was initially trained with a dataset comprising 30 drivers and 3000 deliveries. The following scenarios were tested to evaluate how well the model handles deviations from this balanced setup:

1. **Reduced Drivers and Deliveries**: In this scenario, both the number of drivers and deliveries are scaled down while preserving the overall proportion found in the training data. This approach is designed to evaluate the model's performance when operating on a smaller scale, assessing its ability to make optimal delivery assignments and maintain balanced workloads even with reduced resources.

2. **Imbalance in Drivers and Deliveries**: This scenario creates a higher delivery-to-driver ratio by significantly reducing the number of drivers compared to deliveries. It aims to test the model's resilience and efficiency under increased workload pressure on individual drivers, examining how well it manages performance when resources are strained.

3. **Increased Pressure per Driver**: Here, the setup is closer to the training data, but with a more substantial reduction in drivers rather than in deliveries. This scenario highlights the model's ability to manage higher demands per driver, testing if it can sustain effective performance without a proportional decrease in workload.

4. **Assignment to Different Central Points and Clusters**: In this approach, drivers are assigned to various central hubs, with clustering algorithms used to reorganize delivery groupings and assignments. The aim is to evaluate the model's adaptability to altered geographic configurations and assess the efficiency of delivery assignments with different cluster types and central points.

5. **Deployment in a Different Geographical Region**: This scenario involves testing the model in a new geographical area distinct from the training region, such as the metropolitan area of Lisbon. The goal is to assess the model's adaptability to different geographic constraints and infrastructure, evaluating its ability to transfer learned strategies to a new environment.

6. **Scaling to Larger Operations**: In this case, the model is tested in a significantly larger operational environment, with an increased number of drivers, deliveries, and a broader territory. This scenario evaluates the model's scalability and its capability to manage a more complex and expansive operation, testing its limits in handling larger-scale logistics.

The stress scenarios can test the limitations of the RL model when faced with conditions significantly different from the training environment. While the model can perform adequately under slightly reduced proportions, larger deviations may led to notable declines in efficiency and

balance. This underscores the importance of maintaining a balanced training environment and suggests that additional strategies, such as adaptive learning or hybrid approaches combining RL with heuristic methods, might be necessary to handle more extreme variations in delivery and driver numbers.

5.3 Analyzing Overfit

In order to thoroughly analyze the performance of the reinforcement learning (RL) model and detect signs of overfitting, it is essential to follow a structured approach. First, we must train a well-performing model by optimizing it on the training data. Once the model achieves satisfactory performance, it can be downloaded and tested on real or simulated datasets that reflect the target operational environment. This process will help determine whether the model is generalizing well or if it is overfitting to the training data.

5.3.1 Visualizing Delivery Assignments by Driver

A key part of the overfit analysis involves creating a visualization to illustrate the delivery assignments. In this visualization, each driver is represented by a distinct color, and the deliveries assigned to them are marked in that color on a map. This visual representation helps quickly identify patterns in the delivery assignments, such as clustering efficiency and workload balance.

Steps to create the visualization:

1. Assign a unique color to each driver for mapping deliveries.
2. Plot the locations of all deliveries and their assigned drivers.
3. Highlight cluster centers and use lines or markers to show distances between delivery points and cluster centers.
4. Include legends and labels to easily identify each driver and cluster.

5.3.2 Action Masking and Business-Critical Done Criteria

In addition to the visualization of delivery assignments, several other considerations are essential for improving the model's efficiency and alignment with business objectives. These include masking techniques to manage actions already taken by the agent and defining a clear done criteria based on business constraints.

Action Masking to Improve Efficiency: Action masking can be employed to enhance the RL model's decision-making process by eliminating invalid actions, such as reassigning deliveries that have already been completed. This helps the agent focus on relevant actions, reducing unnecessary computation and improving overall performance.

- **Eliminating Completed Actions**: Once a delivery is assigned to a driver, that delivery point should be masked so the agent no longer considers it in future actions. This prevents multiple assignments of the same delivery.

- **Driver Capacity Masking**: Drivers can also be removed from the action space once they reach their maximum delivery capacity, which prevents overloading and ensures the model respects operational limits.

Done Criteria Relevant to the Business: To align the RL model with practical business needs, a specific done criteria must be defined to indicate when the agent's task is complete. For a delivery-based operation, the task can be considered complete either when all drivers have reached their maximum number of deliveries or when no more deliveries remain.

- **Driver Maximum Deliveries**: The task completes when all drivers have been assigned the maximum deliveries allowed by business rules, ensuring workload distribution within operational constraints.

- **All Deliveries Assigned**: Alternatively, the episode can terminate when all deliveries in the dataset are assigned, regardless of drivers' remaining capacities, ensuring completion within a single run.

These masking techniques and done criteria help ensure that the RL model remains efficient, complies with real-world business rules, and optimizes the delivery assignment process without exceeding operational constraints. They also prevent unnecessary actions, speed up learning, and focus the agent on feasible actions.

5.3.3 KPIs for Performance Evaluation

To assess the quality of the model's delivery assignments and identify potential overfitting, several Key Performance Indicators (KPIs) can be defined, focusing on the spatial and clustering performance of the model:

1. **Closeness of Deliveries to Cluster Centers**: This KPI measures how close deliveries are to the center of the cluster they belong to. A low average distance between deliveries and the cluster center suggests a well-optimized model. If closeness decreases significantly in new scenarios compared to training, it may indicate overfitting.

$$\text{Closeness} = \frac{1}{N}\sum_{i=1}^{N} d(\text{Delivery}_i, \text{ClusterCenter})$$

   The ideal goal is to minimize this distance, ensuring that drivers travel the shortest possible distances for grouped deliveries.

2. **Outliers**: Outliers are deliveries that are significantly farther away from the cluster center compared to the rest of the deliveries in that cluster. Overfitting may lead to a higher number of outliers when tested in environments different from the training data. Outliers can be detected by identifying deliveries with distances beyond a certain threshold (e.g., 1.5 times the interquartile range of distances).

$$\text{Outliers} = \{\text{Delivery}_i \mid d(\text{Delivery}_i, \text{ClusterCenter}) > \text{Threshold}\}$$

3. **Noise**: Noise represents deliveries that are assigned to one cluster but are geographically closer to another cluster. These mismatches in clustering may suggest that the model is not generalizing well. High noise levels in new environments could be an indicator of overfitting.

$$\text{Noise} = \{\text{Delivery}_i \mid d(\text{Delivery}_i, \text{OtherClusterCenter}) < d(\text{Delivery}_i, \text{AssignedClusterCenter})\}$$

4. **Driver Workload Balance**: A balanced workload is critical for efficient operations. This KPI measures how evenly deliveries are distributed across drivers. A model that has overfitted to the training data may show imbalanced workloads when applied to different datasets.

$$\text{WorkloadBalance} = \frac{\text{Max Deliveries Assigned to a Driver}}{\text{Min Deliveries Assigned to a Driver}}$$

The closer this ratio is to 1, the more balanced the workload distribution is.

5. **Cluster Compactness**: This metric evaluates how tightly packed deliveries are within their clusters. A well-clustered group will have a high degree of compactness, with deliveries grouped closely together, minimizing travel time. A decrease in compactness in new environments could suggest that the model has overfitted to the original training data.

$$\text{Compactness} = \frac{1}{|C|} \sum_{i,j \in C} d(\text{Delivery}_i, \text{Delivery}_j)$$

Lower values indicate more compact clusters, which is ideal for reducing delivery times.

To effectively analyze overfitting, it is necessary to train a high-performing model, visualize the delivery assignments with a distinct color for each driver, and then evaluate the model's generalization ability using relevant KPIs. KPIs such as closeness, outliers, noise, workload balance, and compactness will provide valuable insights into how well the model performs across various scenarios and help detect signs of overfitting.

5.4 Summary

This chapter presents the performance results of reinforcement learning models applied to the delivery optimization problem, focusing on Proximal Policy Optimization (PPO) and Advantage

Actor-Critic (A2C) algorithms. Both models were evaluated based on their ability to handle the complex action space and reward structure of the environment, which includes 2700 possible actions and a challenging reward function with exponential decay and negative penalties.

The PPO agent failed to make significant progress due to issues with the reward function, action space complexity, and insufficient training time. The model struggled to learn effective policies, as the reward values were too small or negative, and the vast action space overwhelmed the agent. The A2C agent also failed to learn a truly effective policy.It was not able to consistently manage the complexity of the environment, indicating limited robustness.

While the models were not tested in six different stress scenarios during this study, this section provides guidelines for how future researchers could evaluate the robustness and adaptability of reinforcement learning models to unbalanced or scaled conditions. These scenarios could include reducing drivers and deliveries, creating imbalanced driver-to-delivery ratios, and testing the models in new geographical areas. If such scenarios are tested, they may reveal how the models handle slight deviations from the training environment and whether larger deviations might significantly reduce efficiency and balance, potentially leading to overfitting.

It is important to note that overfitting analysis was not carried out in this study. Instead, the results and observations from this chapter provide a **framework and guidelines** for detecting and addressing overfitting if further analysis is desired. Future researchers can utilize these observations to prevent overfitting when scaling or testing in more diverse environments.

Chapter 6

Conclusions and Future Work

In this chapter, firstly the the main conclusions and achievements drawn from the research and experimental methodology developed in this thesis are presented in subsection 6.1. After that, in subsection 6.2, the main obstacles and limitations that were encountered during the elaboration of this thesis are described. Lastly, in subsection 6.3, directions for future work are presented, either as a continuation of this work or in complementary areas.

6.1 Conclusions

The presented thesis aimed, firstly, to study the ability of Reinforcement Learning (RL) to learn the allocation of uneven distribution deliveries to a set of drivers/vehicles in an urban logistic scenario (last-mile). To that end, the focus was not only on the resulting allocation but also on maintaining a flexible Area of Interest (AOI). Additionally, the research aimed to explore the implications of overfitting and its mitigation. These objectives were broken down into three research questions, stated in section 1.3, which this thesis aims to answer.

First, a comprehensive overview of the topics covered in this thesis and an analysis of the challenges of applying RL in logistics were presented in chapters 2 and 3, respectively, to fully contextualize this work. Considering the objectives of the thesis, the methodological approach in chapter 4 was distinguished into two different parts. In the first part, a clustering setting was applied to analyze a dataset of deliveries within an urban geographical area. In the second part, the deliveries allocation problem was formalized as a Markov Decision Process (MDP). This methodology was structured to address the three research questions.

The results attained during the first part of the methodological approach demonstrated that the RL agent was able to learn even in the presence of a complex reward structure and varying distribution of deliveries. This directly addresses research question RQ1:

RQ1: Is it possible to maintain a consistent Area of Interest (AOI) for couriers that adjusts to daily changes in order distributions?

Based on the results, the answer to RQ1 is **No**. The RL agent was not capable of dynamically adjusting the AOI for couriers, accommodating daily fluctuations in order distributions while maintaining consistency in performance and allocation efficiency.

For the second research question:

RQ2: Is the reinforcement learning agent prone to overfitting?

While no formal overfitting analysis was conducted in this study, this thesis provides a **framework** and **guidelines** for addressing overfitting. These guidelines outline strategies for detecting and mitigating overfitting when scaling the system or testing it in more diverse environments. Thus, while RQ2 was not explicitly tested, the tools and methods proposed in this work enable future researchers to evaluate and mitigate overfitting risks effectively.

Finally, for the third research question:

RQ3: Can the reinforcement learning agent be deployed in a near real-world scenario, and what would be the implications of doing so?

The results indicate that the RL agent has the potential to be deployed in near real-world scenarios. However, deployment would require careful consideration of **masking techniques** to handle unforeseen variations and adherence to **business constraints**, such as operational costs and customer satisfaction. These factors would play a crucial role in ensuring the scalability and efficiency of the RL system in practical applications.

In summary, this thesis successfully addressed RQ1 and provided actionable insights and a framework for future studies related to RQ2 and RQ3. The findings support the feasibility of using RL in last-mile logistics, with potential avenues for further research and practical deployment.

6.2 Research limitations

The results of this project indicate that while Reinforcement Learning (RL) offers a sophisticated approach to addressing the problem of delivery assignment and optimization, it may not be the most practical or efficient method in this context. Heuristics and meta-heuristics, when combined with machine learning techniques such as k-means clustering, and a proper querying of the database, provide a more robust and adaptable solution. These methods are less computationally intensive and offer greater flexibility in adjusting to delivery patterns and demand changes.

One significant drawback of using RL in this scenario is the need for frequent retraining of the model whenever there are changes in the problem formulation. This retraining requirement can increase computational costs and complexity, making RL less feasible for dynamic and rapidly changing environments. *However, in future work, approaches such as meta reinforcement learning, ensembles, or continuous learning could be considered to address the non-stationarity of the environment. These advanced methods could help mitigate the limitations of standard RL by adapting more efficiently to changing conditions without the need for constant retraining.*

It is crucial to implement a platform that tracks key performance indicators (KPIs) to monitor and evaluate the performance of the model's decisions. This platform would enable continuous assessment of the model's effectiveness and help identify areas for improvement. KPIs such as

delivery times, driver workload balance, and customer satisfaction can provide valuable insights into the model's operational success.

Another consideration is the potential for overfitting in RL models. Since RL relies on historical data and specific problem formulations, there is a risk that the model may become overly tailored to past data, leading to suboptimal performance when faced with new and unseen scenarios. This highlights the importance of regularly updating and validating the model with fresh data to maintain its relevance and accuracy. *This could be particularly relevant in offline RL, where the model results from interaction with the environment. In such cases, it is critical to design the environment to generate data with a wide variety of states. However, the drawback in this scenario would be the inherent trade-off between exploration and exploitation, which is a common challenge in RL models.*

In contrast, heuristics and meta-heuristics, supported by machine learning methods like k-means clustering, offer a more practical approach. By creating clusters daily and determining mean X and mean Y coordinates for each cluster, these methods can effectively adapt to daily changes in delivery demands while maintaining consistent and manageable areas of operation for each driver. This adaptability ensures that the system remains responsive and efficient, reducing the likelihood of overfitting and enhancing overall performance.

6.3 Further Work

Building upon the insights gained from this project, future work should focus on enhancing the practical application of delivery optimization models. One promising direction involves creating daily clusters and computing central points for each cluster. By updating these clusters daily, the system can more accurately reflect changes in delivery demand and driver availability, enabling dynamic adjustments and improving responsiveness to fluctuations in delivery patterns. Additionally, leveraging efficient clustering methods like k-means combined with SQL querying could simplify data retrieval and improve system responsiveness.

A robust real-world implementation and operational framework is also essential. This framework should include mechanisms for real-time data tracking and key performance indicator (KPI) monitoring to continuously assess and optimize model performance. Incorporating business-specific *masking* techniques could enable the system to prioritize deliveries based on criteria such as urgency, customer preferences, or service-level agreements. Implementing a *done criteria* approach could further ensure that the optimization algorithm accurately identifies when objectives have been met, reducing unnecessary recalculations and enhancing computational efficiency.

Considering a multi-agent reinforcement learning (MARL) framework could enhance both performance and adaptability. In this approach, each delivery vehicle operates as an independent agent within a coordinated network, enabling distributed decision-making. This setup allows agents to dynamically adjust routes and workloads in response to changes in delivery demand or geographical constraints. Through inter-agent communication and workload balancing, the system

could achieve more efficient coverage, improve overall performance by minimizing route overlaps, and adapt to real-time shifts in demand patterns.

Integrating advanced machine learning techniques could further improve the model's capacity to respond to complex, real-world logistics challenges. Implementing attention mechanisms would help the model focus on critical aspects of delivery assignments, such as congested areas or high delivery volume locations, improving efficiency in urban environments where delays or traffic patterns impact delivery times. Graph Neural Networks (GNNs) could model the intricate spatial and relational structures of urban delivery systems, where relationships between locations, delivery points, and available drivers play a critical role. This approach would enable the model to capture spatial dependencies more accurately, improving route optimization and area coverage while reducing computational redundancy.

To ensure robustness and scalability, stress testing with larger, more complex datasets is essential. Testing under *stress scenarios* such as spikes in delivery volume, driver shortages, or abrupt changes in delivery areas would provide valuable insights into the model's performance under unbalanced conditions. This phase would also allow for the assessment of computational efficiency and reliability under real-world, high-stakes conditions, identifying areas where further enhancements are necessary.

Addressing the limitations of standard reinforcement learning in dynamic environments could benefit from continuous learning techniques or *meta reinforcement learning* approaches. These methods would enable the system to adapt in real-time to changing delivery patterns and environmental factors without requiring constant retraining. Leveraging meta-learning could help the model generalize across various delivery patterns and conditions, ensuring relevance and efficiency in fast-paced last-mile logistics scenarios while reducing computational overhead.

Finally, optimizing computational efficiency in both model training and deployment remains a critical focus. Techniques such as the *First-In-First-Out (FIFO) heuristic* for priority handling, *masking* to prevent repeated actions, and *SQL querying for priority management* can manage high-priority deliveries while minimizing redundant calculations. Additionally, enhancing the reward structure to balance between exploration and exploitation would allow the model to operate more efficiently, especially in non-stationary environments where delivery demands and routes vary frequently.

In summary, future work should prioritize developing adaptive, multi-agent systems that integrate advanced machine learning techniques for handling spatial dependencies, explore continuous learning for real-time adaptability, and stress-test the model with larger datasets. These enhancements will support a more scalable, computationally efficient reinforcement learning framework for deployment in dynamic, last-mile delivery logistics environments.

References

Mohammad Aljaidi, Nauman Aslam, Xiaomin Chen, Omprakash Kaiwartya, Yousef Ali Al-Gumaei, and Muhammad Khalid. A reinforcement learning-based assignment scheme for evs to charging stations. In *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pages 1–7, 2022. doi: 10.1109/VTC2022-Spring54318.2022.9860535.

Portal DO ESTADO DO AMBIENTE. EmissÕes de gases com efeito de estufa. Available at https://rea.apambiente.pt/content/emiss%C3%B5es-de-gases-com-efeito-de-estufa/, Accessed last time in july 2024, 2024.

Aysun Bozanta, Mucahit Cevik, Can Kavaklioglu, Eray M. Kavuk, Ayse Tosun, Sibel B. Sonuc, Alper Duranel, and Ayse Basar. Courier routing and assignment for food delivery service using reinforcement learning. *Computers Industrial Engineering*, 164:107871, 2022. ISSN 0360-8352. doi: https://doi.org/10.1016/j.cie.2021.107871. URL https://www.sciencedirect.com/science/article/pii/S0360835221007750.

Alberto Castagna, Maxime Guériau, Giuseppe Vizzari, and Ivana Dusparic. Demand-responsive rebalancing zone generation for reinforcement learning-based on-demand mobility. *AI Communications*, 34:1–16, 01 2021. doi: 10.3233/AIC-201575.

Joshua Julian Damanik, Hans Kasan, and Han-Lim Choi. Solving delivery assignment in hybrid-transit network using multi-agent reinforcement learning. In Jinwhan Kim, Brendan Englot, Hae-Won Park, Han-Lim Choi, Hyun Myung, Junmo Kim, and Jong-Hwan Kim, editors, *Robot Intelligence Technology and Applications 6*, pages 485–497, Cham, 2022. Springer International Publishing.

Jose Escribano, Huan Chang, and Panagiotis Angeloudis. Integrated path planning and task assignment model for on-demand last-mile uav-based delivery. In Jesica de Armas, Helena Ramalhinho, and Stefan Voß, editors, *Computational Logistics*, pages 198–213, Cham, 2022. Springer International Publishing. ISBN 978-3-031-16579-5.

David Ha and Jürgen Schmidhuber. World models, 2018.

Instituto Nacional de Estatística. Sociedade da informação e do conhecimento - inquérito à utilização de tecnologias da informação e da comunicação nas famílias, November 22 2021. URL https://www.ine.pt/xportal/xmain?xpid=INE&xpgid=ine_destaques&DESTAQUESdest_boui=473570976&DESTAQUESmodo=2. Accessed: 2024-01.

Hadi Jahanshahi, Aysun Bozanta, Mucahit Cevik, Eray Mert Kavuk, Ayşe Tosun, Sibel B. Sonuc, Bilgin Kosucu, and Ayşe Başar. A deep reinforcement learning approach for the meal delivery problem. *Knowledge-Based Systems*, 243:108489, 2022. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2022.108489. URL https://www.sciencedirect.com/science/article/pii/S0950705122002088.

Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

Saeid Kalantari, Reza Safarzadeh Ramhormozi, Yunli Wang, Sun Sun, and Xin Wang. Trailer allocation and truck routing using bipartite graph assignment and deep reinforcement learning. *Transactions in GIS*, 27:1020 – 996, 2023. URL https://api.semanticscholar.org/CorpusID:258732163.

Yafei Li, Qingshun Wu, Xin Huang, Jianliang Xu, Wanru Gao, and Mingliang Xu. Efficient adaptive matching for real-time city express delivery. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5767–5779, 2023. doi: 10.1109/TKDE.2022.3162220.

Miaojia Lu, Xinyu Yan, Shadi Sharif Azadeh, and Pengling Wang. An adaptive agent-based approach for instant delivery order dispatching: Incorporating task buffering and dynamic batching strategies. *International Journal of Transportation Science and Technology*, 13:137–154, 2024. ISSN 2046-0430. doi: https://doi.org/10.1016/j.ijtst.2023.12.006.

Daiki Min and Yuncheol Kang. A learning-based approach for dynamic freight brokerages with transfer and territory-based assignment. *Computers Industrial Engineering*, 153:107042, 2021. ISSN 0360-8352. doi: https://doi.org/10.1016/j.cie.2020.107042. URL https://www.sciencedirect.com/science/article/pii/S0360835220307129.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Junyuan Quan and Ning Wang. An optimized task assignment framework based on crowdsourcing knowledge graph and prediction. *Knowledge-Based Systems*, 260:110096, 2023. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2022.110096. URL https://www.sciencedirect.com/science/article/pii/S0950705122011923.

Alexandra Samet. Last-mile delivery: What it is and what it means for retailers, October 12 2023. URL https://www.emarketer.com/insights/last-mile-delivery-shipping-explained/. Accessed: 2024-01.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 576–589. Curran Associates Inc., 2017.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, second edition edition, 1998.

Yuechuan Tao, Jing Qiu, Shuying Lai, Xianzhuo Sun, and Junhua Zhao. A data-driven agent-based planning strategy of fast-charging stations for electric vehicles. *IEEE Transactions on Sustainable Energy*, 14(3):1357–1369, 2023. doi: 10.1109/TSTE.2022.3232594.

Gustavo Macedo Torres. Sectorization for last-mile delivery. Master's thesis, Faculdade de Engenharia da Universidade do Porto, July 2022.

Hai Wang, Shuai Wang, Yu Yang, and Desheng Zhang. Gcrl: Efficient delivery area assignment for last-mile logistics with group-based cooperative reinforcement learning. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 3522–3534, 2023a. doi: 10.1109/ICDE55515.2023.00269.

Xing Wang, Ling Wang, Chenxin Dong, Hao Ren, and Ke Xing. An online deep reinforcement learning-based order recommendation framework for rider-centered food delivery system. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):5640–5654, 2023b. doi: 10.1109/TITS.2023.3237580.

Jiayi Wen, Shaoman Liu, and Yejin Lin. Dynamic navigation and area assignment of multiple usvs based on multi-agent deep reinforcement learning. *Sensors*, 22(18), 2022. ISSN 1424-8220. doi: 10.3390/s22186942. URL https://www.mdpi.com/1424-8220/22/18/6942.