FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Optimising Bike-Sharing Systems Using Multi-agent Reinforcement Learning

Iva Ilkova Nikolova

U. PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Master's degree in Informatics and Computing Engineering

Supervisor: Prof. Rosaldo Rossetti

July 30, 2025

Optimising Bike-Sharing Systems Using Multi-agent Reinforcement Learning

Iva Ilkova Nikolova

Master's degree in Informatics and Computing Engineering

Approved in oral examination by the committee:

President: Prof. Ana Paula Rocha
Referee: Prof. Pedro Campos

July 30, 2025

Abstract

Bike-sharing systems are becoming an essential part of city transportation, helping to reduce traffic and pollution. However, a significant challenge for these systems is to ensure that there are enough bikes available at each station when users need them. This dissertation focuses on improving bike redistribution to maintain stations balanced, allowing more people to benefit from the service without waiting.

The study employs multi-agent reinforcement learning, a form of artificial intelligence, to address this issue, treating each bike station as a separate decision-maker. These stations learn to move bikes between each other using a method called Deep Q-Network (DQN). The goal is to move bikes during times when demand is lower, during off-peak hours, so the stations are better prepared for periods of high demand.

The method is tested using computer simulations with data from the Madrid bike-sharing system. Different scenarios are created by changing the initial bike count per station at the beginning of the simulation day. The results demonstrate that the multi-agent DQN algorithm adapts effectively over time, improving key performance metrics, including missed trips, trip completion rates, and station availability. Scenarios with moderate initial bike levels (30 and 40 bikes per station) offer the best balance between user service quality and operational cost. The approach demonstrates the potential of reinforcement learning for dynamic data-driven management in bike-sharing networks, outperforming a baseline method without redistribution.

This paper contributes to the field by combining multi-agent reinforcement learning approaches with urban mobility challenges and highlighting practical aspects for implementing intelligent redistribution strategies. Future work will focus on integrating real-time demand, conducting pilot tests in operational systems, and developing adaptive infrastructure planning to improve flexibility and resilience.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework for achieving a better and more sustainable future for everybody. They address the most important issues facing the world, including poverty, inequality, climate change, environmental degradation, peace, and justice. A few of these objectives are effectively supported by this project, which encourages climate action, stimulates sustainable urban mobility, and encourages economic innovation by optimizing smart bike-sharing systems.

SDG 8 Encouraging efficiency and innovation in urban transportation builds new business opportunities, jobs, and long-term business practices that support economic expansion.

SDG 11 Promoting sustainable urban mobility systems and improving public transportation infrastructure make cities safer and more resilient, reduce pollution, and improve the quality of life for everyone.

SDG 13 Supporting environmentally friendly forms of transportation, like bike-sharing programs, can help reduce greenhouse gas emissions and lessen the effects of climate change.

SGD	**Target**	**Contribution**	**Performance Indicators and Metrics**
8	8.1	Operational efficiency and innovation through AI-enabled bike redistribution.	Improved system reliability and reduced operational costs.
11	11.1	Increased accessibility and sustainability of urban transport via optimized bike-sharing.	Increased bike availability; reduction in urban traffic congestion.
13	13.1	Reduction of greenhouse gas emissions by promoting sustainable mobility.	Estimated CO2 emissions reduction from bike-sharing modal shift.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Rosaldo Rossetti, for their invaluable guidance, encouragement, and support throughout this research project. Their expert advice and constructive feedback have been essential in shaping this work.

I also wish to thank the University of Porto, particularly the Faculty of Engineering (FEUP), for providing the academic environment and resources that made this study possible.

This research was supported by the extensive body of work from previous scholars and practitioners in the field, as cited in the references, whose contributions have greatly informed and enriched this dissertation.

Iva Ilkova Nikolova

Contents

List of Figures

List of Tables

List of Acronyms

DQN	Deep Q-Network
MARL	Multi-Agent Reinforcement Learning
RL	Reinforcement Learning
SOS	Self-Organized System
CSV	Comma-Separated Values
VS Code	Visual Studio Code
AI	Artificial Intelligence
GPU	Graphics Processing Unit
ReLU	Rectified Linear Unit

Chapter 1

Introduction

Bicycles are widely recognized as an environmentally friendly form of urban transportation, suitable for both short and medium-distance journeys [24], [26]. Since the appearance of the first bike-sharing system in the 1960s, it has progressed through three different stages. The initial iteration, known as Witte Fietsenplan (the "white bicycle plan"), was established in Amsterdam in 1965 with a fleet of 50 unsecured bicycles [14]. However, due to the increasing number of thefts and damage, the system was unsuccessful. The introduction of docking stations and small deposits for unlocking bikes marked the arrival of the second generation of coin deposit systems in the early 1990s. Bycyklen in Copenhagen launched the initial widespread bike-sharing program in 1995 [6]. However, theft was still a problem because users were not identified. Later, electronic membership cards were introduced, which helped reduce theft and vandalism, making bike-sharing systems more secure and reliable.

By April 2013, approximately 500 bike-sharing systems had been successfully implemented in numerous cities worldwide as part of the third generation [15]. As of June 2025, there are around 9.6 million active bikes used around the world according to BikeSharingMap [13]. The fourth generation faces the challenges of introducing electric bikes and the advancement of free-floating systems.

Issues related to operations continue to be a significant problem in current bike-sharing systems. The imbalance, resulting from the unpredictability and unevenness of demand, results in situations where some stations have no bikes available, while others are overcrowded. Repositioning is widely recognized as a practical approach to address these issues. This paper proposes the development of an agent-based simulator to replicate the functionality of bike-sharing systems. This involves creating a set of beliefs and approaches that effectively mimic actual behavior.

1.1 Context and Motivation

Globally, cities are facing the difficulties of creating sustainable transportation systems. As populations continue to grow and the urgent need to reduce greenhouse gas emissions increases, cities are increasingly relying on bike-sharing systems instead of traditional transportation methods [11].

These systems offer numerous benefits, including being environmentally friendly, reducing traffic congestion, promoting wellness, and facilitating the use of public transportation.

The implementation of bike-sharing has quickly and extensively spread, with numerous cities committing to these programs. Nevertheless, the rapid speed of execution has not always been supported by careful planning and evaluation. Expanding urban areas require cities to prioritize understanding and optimizing the operation of services to sustainably and effectively meet demand. Creating a reliable forecast tool for planning and overseeing bike-sharing systems is essential in supporting this eco-friendly urban transportation effort.

1.2 Problem Identification

Although bike-sharing systems offer advantages, they also face significant operational challenges. An essential issue is system imbalance, with some stations having a large number of bikes due to changing demand, while others have low numbers. Not having a proper balance can lead to users feeling unhappy, resources not being fully utilized, and increased expenses from having to relocate bikes across stations.

Methods to solve these disparities often require effective strategies, such as using trucks to transport bicycles between stations. However, they can be costly, require extensive efforts, and may not achieve the calculated results. An immediate resolution is necessary to forecast and address these inconsistencies before they lead to service disruptions. If cities cannot simulate and evaluate potential configurations and strategies, they risk implementing ineffective systems that do not meet user needs or financial goals.

1.3 Research Goal

This dissertation focuses on creating and analyzing a multi-agent reinforcement learning framework for optimizing bike redistribution in urban bike-sharing systems. Every bike station is represented as an intelligent agent that can learn to efficiently manage its bike inventory based on demand predictions and the overall system condition. A key aspect of the approach is the use of Deep Q-Networks (DQN), which allows the agents to plan bike transfers during specified off-peak hours. This scheduling enables the system to rebalance bike availability before periods with high demand actively.

In order to address this issue, the study is guided by the following research questions, which will be answered in this paper:

1. What are the most effective strategies for addressing system imbalance and optimizing the distribution of bikes across stations?

2. How can predictive analytics, driven by machine learning, improve the accuracy of demand forecasting in bike-sharing systems?

3. How can a multi-agent simulation model be calibrated to reflect real-world data and scenarios from existing bike-sharing systems accurately?

4. How can multi-agent reinforcement learning (MARL) be used to model the balance of sharing points?

5. How do different rebalancing algorithms impact operational costs, user satisfaction, and system reliability?

6. What role can public participation and feedback play in optimizing the design and management of bike-sharing systems?

1.4 Dissertation Structure

The dissertation is organized as follows: Chapter 2 reviews relevant literature on bike-sharing optimization, predictive analytics, and reinforcement learning approaches. Chapter 3 presents the methodological approach, including data preparation, simulation design, the multi-agent DQN framework, and evaluation metrics. Chapter 4 discusses the results of multiple simulation scenarios and compares system performance across different bike inventories. Finally, Chapter 5 summarizes the findings, mentions the main contribution, and potential directions for future work.

Chapter 2

Literature Review

Bike-sharing systems are now seen as an eco-friendly way to travel in cities for short distances, helping to reduce traffic, emissions, and the need for personal cars. Even though these systems are used in many cities around the world, they encounter operational difficulties, especially when it comes to ensuring an even distribution of bikes among stations. Imbalances occur when some stations experience high demand while others remain underutilized, resulting in a negative impact on user satisfaction and system efficiency.

In this chapter, several techniques are proposed to address these issues, drawn from various papers. These strategies include data-driven forecasting methodologies, machine learning techniques, reinforcement learning frameworks, and user-centered approaches.

2.1 Approaches to Addressing System Imbalance and Bike Redistribution

An essential subject in bike-sharing research is how to address the differences in bike availability across stations. Shi et al. [27] create a complete data-driven framework that forecasts demand and optimizes rebalancing tactics by taking into account many variables such as temporal elements like time changes and holidays, as well as external events like weather. This methodology uses machine learning to train models that predict imbalances and propose corrective actions, especially for electric and hybrid bike systems.

Reinforcement learning (RL) also shows promise in this area. Li et al. [19] use a reinforcement learning approach to develop adaptive redistribution policies in which agents dynamically optimize bike availability by learning from their surroundings. This technique demonstrates RL's ability to manage the unpredictable demand and complexity of bike-sharing operations.

User behavior and preferences are also crucial. Bravo et al. [5] integrate behavior data and real-time user feedback to improve redistribution decisions. Meanwhile, Lu et al. [22] propose a multimodal strategy that combines bike-sharing with public transportation services, encouraging the use of free bike-sharing to help balance usage across the network.

Additionally, Jing et al. [16] present a brand-new data-driven method that uses routine pattern mining to find constant imbalances in both time and space in bike-sharing networks. Similarly, Liu et al. [20] propose a dynamic repositioning approach for bike fleets that adapts in real-time, reducing user wait times and missed trips by effectively repositioning bikes. By modeling bike movements as a dynamic pickup and delivery problem, their approach addresses the operational scalability issues that come with urban networks. Furthermore, Chiariotti et al. [9] optimize bike redistribution policies by combining heuristic algorithms with multi-agent reinforcement learning.

These studies demonstrate that effective redistribution requires a combination of predictive analytics, adaptive learning, and user-centered techniques.

2.2 Predictive Analytics for Demand Forecasting

Accurate demand forecasting is required for adaptive bicycle redistribution. Maleki et al. [23] use simulation-based supervised learning methods to enhance forecasting models, creating hypothetical data. Their model better captures real-world unpredictability by including factors like weather and demand variations.

Similarly, Alaoui et al. [1] highlight the use of previous usage patterns, as well as external elements like weather, to improve predictions. Boufidis et al. [4] use machine learning to examine station-level demand, user behavior, and historical trends. While Bae et al. [3] do not employ machine learning, they instead apply data-driven methodologies to model user behavior and demand distributions.

In addition, McMullen [25] presents time-series and probabilistic demand forecasting models that utilize past data, as well as external characteristics such as weather and special events. Their model enables more effective redistribution planning by accounting for temporal relationships and user behavior unpredictability.

To increase forecasting accuracy in bike-sharing systems, Albuquerque et al. [2] suggest a hybrid predictive model that combines machine learning with outside factors like the weather, the time of the day, and past demand. This model enables employees to forecast demand increases and optimize bike stock levels.

Contardo et al. [12] analyze the critical challenge of balancing public bike-sharing systems dynamically during peak demand periods. Their work focuses on minimizing the number of users who cannot find a bike or a docking spot by strategically allocating a fleet of vehicles from surplus to deficit stations.

These approaches together demonstrate a progression from purely statistical models to hybrid data-driven and machine learning frameworks that better adapt to changing demand trends.

2.3 Calibration of Multi-Agent Simulation Models

Simulating bike-sharing networks with multi-agent models allows researchers to evaluate and optimize redistribution strategies before real-world implementation. By using reinforcement learning in a multi-agent framework, Li et al. [19] enable agents to discover the best way to distribute bikes.

Alaoui et al. [1] model agent interactions more deeply, including bikes, stations, and users as agents that adapt their behavior in response to environmental feedback. Lu et al. [22] employ behavioral choice theories, such as limited capacity and random utility maximization, to enhance simulation realism using real data from Taipei. Using demographic, geographic, and operational data, Bae et al. [3] construct a highly detailed agent-based simulation for Sejong City that mimics real usage and demand.

The accuracy of multi-agent simulations is dependent on proper calibration to real-world data. In order to increase the realism of agent-based simulations, Leclaire et al. [18] present a complete data-driven calibration technique that includes station-level demand, user behavior, and temporal patterns. Their techniques take into account spatial-temporal dependencies, ensuring that agent actions represent the dynamics and limitations of the system.

Cipriano et al. [10] highlight the significance of adding imbalance patterns discovered by machine learning into simulation calibration. This allows models to replicate critical situations found in real systems.

To simulate realistic agent decision-making processes in bike-sharing networks, Lozano et al. [21] present a multi-layer calibration framework that considers demographic data, urban mobility patterns, and historical operational data. Their method enables context-sensitive simulation scenarios.

Chi [8] presents a novel approach to bike redistribution using a self-organized system (SOS) framework, where individual bike users act as agents following simple social rules in a multi-agent simulation. In contrast with standard operator-controlled redistribution techniques, this model places a strong emphasis on user-driven independent decision-making.

These various calibration techniques increase simulation accuracy, which is essential for creating efficient redistribution algorithms and assessing operational strategies.

2.4 Multi-Agent Reinforcement Learning for System Optimization

Multi-agent reinforcement learning (MARL) approaches provide a powerful solution for automated bike redistribution. Alaoui et al. [1] apply MARL more specifically by modeling bikes, rides, and stations as learning agent that dynamically adjust their behaviors using real-time and historical data. While Maleki et al [23] focus mainly on supervised learning, their simulation framework could integrate MARL techniques. Lu et al. [22] expand on this by combining MARL and geographic agent-based models to capture complex transportation choices.

To demonstrate how MARL can enhance system balance through coordinated agent action, Soriguera et al. [28] optimize multi-agent decision-making for trucks, bikes, and stations.

To improve redistribution policies in urban bike-sharing systems, Chiariotti et al. [9] combine MARL with heuristic algorithms to achieve a balance between learning efficiency and operational effectiveness.

Liu et al. [20] offer a large-scale dynamic repositioning framework that improves MARL by using adjustable algorithms to optimize fleet movements. This framework is particularly suitable for highly unpredictable and crowded urban environments.

To further improve MARL, Albuquerque et al. [2] introduce adaptive reward shaping, which guides agents to consider not only short-term costs but also long-term system health and user satisfaction, resulting in more environmentally friendly redistribution strategies.

These studies highlight MARL's ability to handle complexity, scale, and real-time adaptability in bike-sharing systems.

2.5 Effects of Rebalancing Algorithms on Costs and User Experiences

Rebalancing algorithms have a significant impact on user satisfaction, system reliability, and operational costs. Shi et al. [27] examine how different rebalancing strategies affect these metrics, showing their crucial role in system performance. Bravo et al. [5] take a user-centered perspective, evaluating algorithms based on travel behavior and satisfaction data to establish their real-world performance. Boufidis et al. [4] analyze station-level demand data to understand how rebalancing affects user experience and operational efficiency. Emphasizing the need for carefully selecting rebalancing procedures, Soriguera et al. [28] compared periodic and continuous rebalancing, finding significant differences in cost and system stability.

Jing et al. [16] analyze the importance of reducing costs while maintaining high service quality. They analyze metrics such as the number of redirecting changes made and the total distance driven by empty repositioning vehicles, demonstrating how effective rebalancing techniques can reduce unnecessary movements and associated expenses. Liu et al. [20] also address cost efficiency while maintaining availability by developing repositioning strategies that minimize travel distance and rebalancing costs. Lozano et al. [21] demonstrate how careful algorithm design and calibration to local demand characteristics can balance decisions, thereby affecting user satisfaction and operating costs.

The static rebalancing problem in bike-sharing systems is examined by Chemla et al. [7] with a focus on maintaining service quality while reducing operating expenses. In order to reach target stock levels, they model the problem as a trained pickup and delivery routing task for a single vehicle that involves redistributing bikes among stations.

Balancing operational costs with user satisfaction remains a key challenge and research focus.

2.6 Role of Public Participation and Feedback

Implementing public engagement and user feedback improves bike-sharing system design and management. In order to increase system usability and user acceptance, Alaoui et al. [1] highlight the integration of user preferences and travel behavior into decision-making. To improve overall usability, Bravo et al. [5] suggest using user data to inform changes.

Boufidis et al. [4] emphasize the role of user feedback in improving demand prediction and visualization, which informs system management. Bae et al. [3] employ user behavior data indirectly, focusing on its importance even in the absence of clear feedback. Additionally, Leclaire et al. [18] and Lozano et al. [21] integrate behavioral and demographic information into agent behavior models, enabling the system to enhance responsiveness and consider a range of user preferences without requiring direct user involvement.

All of these findings highlight how essential it is to understand different user requirements and include such information to improve bike-sharing systems' responsiveness and design, opening the door for more direct public engagement techniques in future research.

2.7 Summary

Bike-sharing systems have developed as an environmentally friendly solution for urban mobility, but they face challenges such as system imbalance, demand forecasting, and effective rebalancing. To address these issues, researchers developed advanced simulation and decision-support tools. Data-driven machine learning models, reinforcement learning for adaptive rebalancing, user-centric approaches based on behavioral data, and multimodal integration that connects bike-sharing and public transit are among the strategies for addressing system imbalance. Predictive analytics improves demand forecasting by using supervised learning models trained on both historical and synthetic data. Multi-agent simulation models are calibrated based on reinforcement learning, behavioral theories, and detailed demographic and geographic information. MARL optimizes agent interactions, which improves system efficiency. Static and dynamic rebalancing algorithms have been tested to see how they affect costs, reliability, and user satisfaction.

Chapter 3

Methodological Approach

This study aims to optimize bike redistribution in bike-sharing systems, a crucial aspect of sustainable urban mobility. It creates and assesses intelligent redistribution strategies using a comprehensive simulation framework. The study employs two primary approaches: a basic scenario, which represents the natural flow of bikes without any interventions, and a demand-based redistribution scenario, which utilizes historical demand data to relocate bikes during off-peak hours. The primary goal is to design, implement, and evaluate a multi-agent reinforcement learning system based on Deep Q-Networks (DQN) to optimize the bike redistribution. Every station is represented as an independent agent that makes choices based on the demand forecast and its state. Essential performance metrics, such as missed trips, trip completion rates, station availability, and rebalancing costs, are implemented to measure the system's performance systematically. The study provides an in-depth description of the methods used in this research, including the implementation of the multi-agent DQN framework, data preparation techniques, and the simulation environment.

3.1 Experimental Process

3.1.1 System Approaches: Basic and Demand-Based Redistribution

The evaluation of the bike-sharing system is based on two main approaches that aim to improve operational efficiency and user satisfaction progressively. The first method, known as the **Basic Approach**, is the model that represents the natural, unregulated movement of bicycles across the network. In this case, bike movements are entirely based on user interactions, with bikes picked up and returned to stations according to real-world usage patterns. There is no additional intervention or optimization strategy applied in the system, making this approach a critical benchmark for understanding how well the system manages demand and supply under everyday conditions. By simulating regular user activity without optimization, the Basic Approach provides a clear overview of the system's ability to maintain availability and service quality.

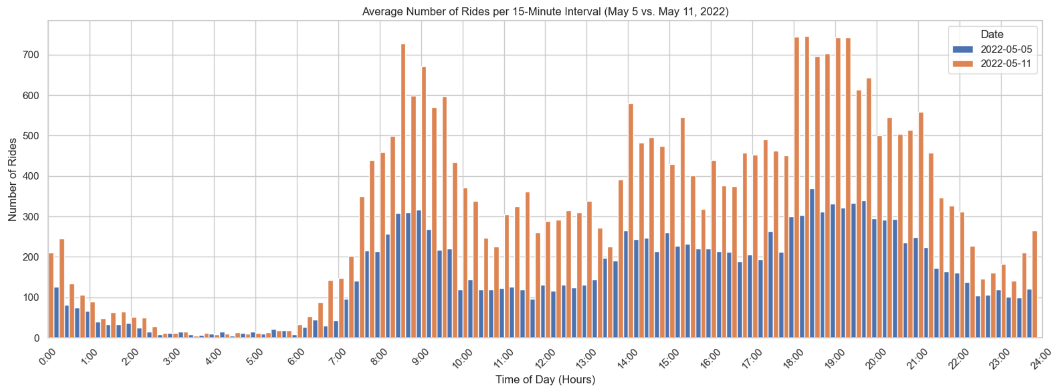

Figure 3.1: Average number of Rides for Both Days in 15-minute Intervals

Building upon the baseline, the second approach introduces an intelligent **Demand-Based Redistribution** strategy. This method solves one of the key operational challenges faced by bike-sharing systems: geographical and temporal imbalances in bike availability. Using historical demand data and predictive analytics, redistribution actions are concentrated during off-peak hours, specifically between 12 and 1 PM (Figure 3.1). During this window, bikes are strategically moved from stations with surplus inventory to those forecast to experience high demand in the following peak hours. This predictive reallocation tries to prevent shortages and minimize the number of missed trips, improving system adaptability. The effectiveness of this demand-driven method is carefully tested using performance metrics, which demonstrate its impact on the overall system performance.

Together, these techniques represent the system's progression from a passive, user-driven model to an actively managed, data-informed redistribution framework. This dual approach provides a solid foundation for evaluating the operational benefits and limitations of reinforcement learning-enabled bike redistribution under demand conditions.

3.1.2 Project Goals

The primary objective of this project is to optimize bicycle distribution within a shared network to improve service reliability and customer satisfaction. The focus is on redistributing bikes ahead of peak demand hours, ensuring that stations that are expected to see high usage have sufficient stock before these critical times. By pre-positioning bikes at these high-demand locations, the system aims to minimize shortages during peak hours, enabling users to initiate and complete trips without interruption.

This redistribution technique must balance between two conflicting considerations. Firstly, it aims to minimize overstocking stations, which can lead to inefficiencies such as more idle bikes and higher operational costs due to unnecessary relocation. Secondly, it aims to source bikes mostly from stations with consistent but lower demand. Most importantly, the approach maintains a baseline inventory at these lower-demand stations, allowing customers in these locations to still start and finish their trips.

By establishing this balance, the project expects to improve the overall balance of bike availability throughout the system. The goal is to create a dynamic, demand-responsive redistribution mechanism that maintains efficient operational growth, enabling seamless daily bike-sharing experiences.

3.1.3 Multi-Agent Deep Q-Network Framework

To properly handle the dynamic and complex nature of bike redistribution, this study employs a multi-agent reinforcement learning (MARL) technique, specifically Deep Q-Networks (DQN). In this framework, each station is modeled as an independent agent that learns to make decisions about bike transfers based on its current state and environmental observations. The multi-agent DQN method allows stations to collaborate on redistribution policies by learning from ongoing interactions within the system. This strategy addresses challenges such as changing demand patterns, station imbalances, and operational constraints.

Each agent uses a neural network to approximate the best action-value function, allowing it to choose actions that maximize progressive rewards over time. The reward function is carefully designed to balance between several goals, like eliminating wasteful bike movements and maintaining an optimal inventory level.

By training these agents simultaneously, the system achieves a coordinated redistribution strategy, which is adapted to real-time conditions and demand forecasts. That is why the use of multi-agent DQN forms the core intelligence behind the redistribution mechanism.

3.2 Simulation Framework

This section discusses the development of a simulation framework for evaluating the Madrid Bike Sharing System. It describes how the dataset was sourced, cleaned, and analyzed. The simulation environment is introduced, including interactive system dynamic visualizations in real-time. The time management approach is explained, enabling the analysis of practical system behavior.

The framework forms a user-friendly platform for studying the impact of multi-agent redistribution strategies on bike-sharing performance under varying demand conditions.

3.2.1 Data Preparation and Processing

The dataset used for this study is sourced from the *Madrid Bikes Sharing Dataset* available on Kaggle from 2022 [17]. This large dataset covers a full year of operations and contains detailed trip records for the Madrid bike-sharing system.

Two main CSV files were taken out of the dataset for analysis: *all_stations.csv*, which includes detailed information about each bike station, including its geographic coordinates, and *all_trips.csv*, which documents individual bike trips, including start and end times, origin and destination stations, and trip durations.

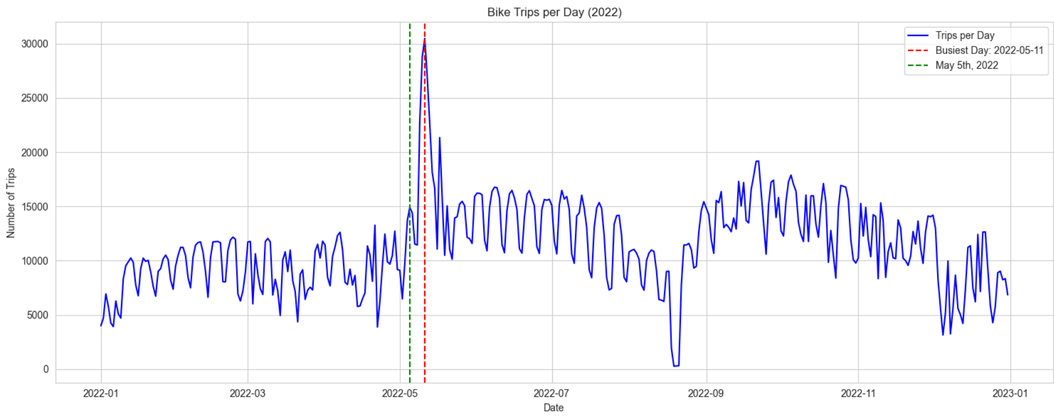

Figure 3.2: Bikes Trips per Day Throughout the Entire Year

To guarantee reliability and precision, the first step required data cleaning. This process involves eliminating duplicate or invalid entries, correcting date-time formats, and addressing missing or inconsistent values. For example, trips with zero or negative duration were not included. Additionally, to verify that the stations were situated within the city boundaries, geographical data is also verified. Standardization of station identifiers and names is performed to ensure accuracy throughout the simulation environment and across datasets.

An exploratory analysis is carried out after data cleaning to understand usage patterns and identify key demand variations. During this stage, May 11th is recognized as the busiest day of the year, with almost twice as many trips as on a more typical day, such as May 5th (Figure 3.2). This realization brought the selection of these two days as representative examples for examining the bike-sharing system under contrasting demand conditions.

3.2.2 Simulation Environment

Throughout the simulated days, real-time information on the bike-sharing system's operations is presented as an interactive visual interface in the simulation environment. At the top of the interface, the current simulation time is shown, allowing users to follow the days' development from start to finish.

Two side-by-side maps, at the center of the interface, show station usage patterns under different approaches (Figure 3.3). The left map represents the **Basic Scenario**, which shows bike movements between stations naturally without any optimization or redistribution. This baseline visualization enables users to observe how bikes move throughout the day under typical conditions.

The right map displays the same simulation period with the **Demand-Based Redistribution** approach with multi-agent reinforcement learning. To demonstrate how the algorithm impacts the network's ability to balance bike availability, this scenario incorporates intelligent bike redistribution during off-peak hours. Users can visually compare the differences in station statuses and bike distribution between the two approaches in real time.

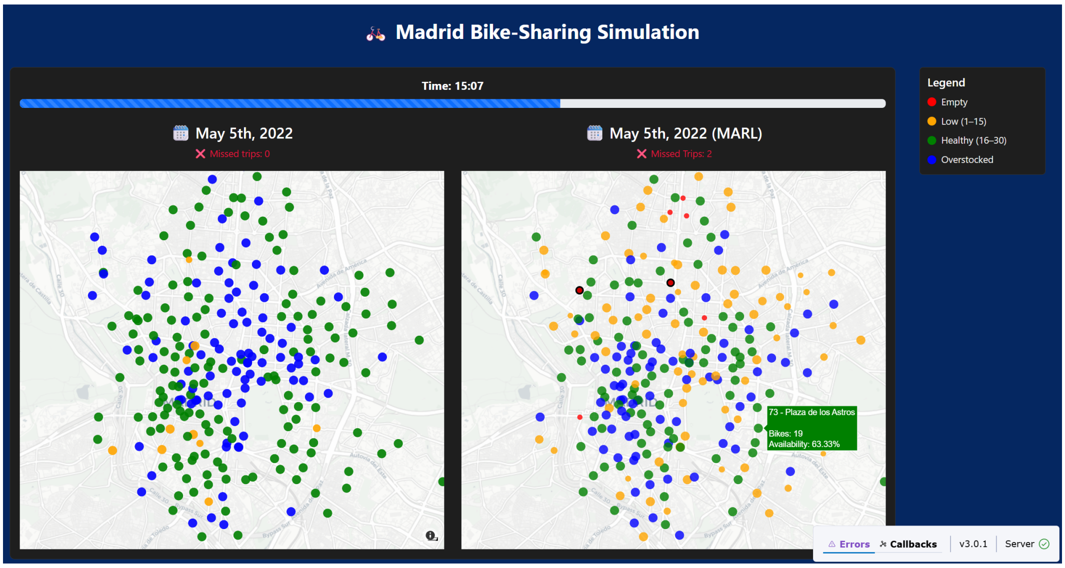

Figure 3.3: Simulation Environment for May 5th during the Simulation

Two additional sets of maps, which reproduce the same comparative view for May 11th (Figure 3.4), the busiest day in the dataset, are shown below the first two maps. These sets demonstrate how the system behaves during periods of extremely high demand.

To the right of the maps, a color-coded legend describes the current status of each bike station, indicating the number of bikes available. Four color levels are presented by station markers that change color dynamically throughout the simulation:

- Red: Empty stations with zero bikes.
- Orange: Low stock, typically 1 to 15 bikes.
- Green: Healthy stock, generally between 16 and 30 bikes.
- Blue: Overstocked stations exceeding 30 bikes.

Another feature of the simulation is the ability to track missed trips in real time. A **missed trip** occurs when a user attempts to start a journey from a station with no available bikes in it. Their number is shown above the map, and stations that have missed trips are indicated with a black border. During the simulation, users can identify systemic issues thanks to this immediate visual feedback.

Additionally, the interface supports detailed station-level information. The user can view information, including the station name and the number of bikes available, by moving the cursor over any station marker. This feature improves user interaction and understanding of individual station statuses.

A summary panel displaying essential information about all evaluation metrics is shown at the end of the simulation, positioned above the map. These metrics provide a clear picture of how the system performed throughout the day, displaying information such as the total number of

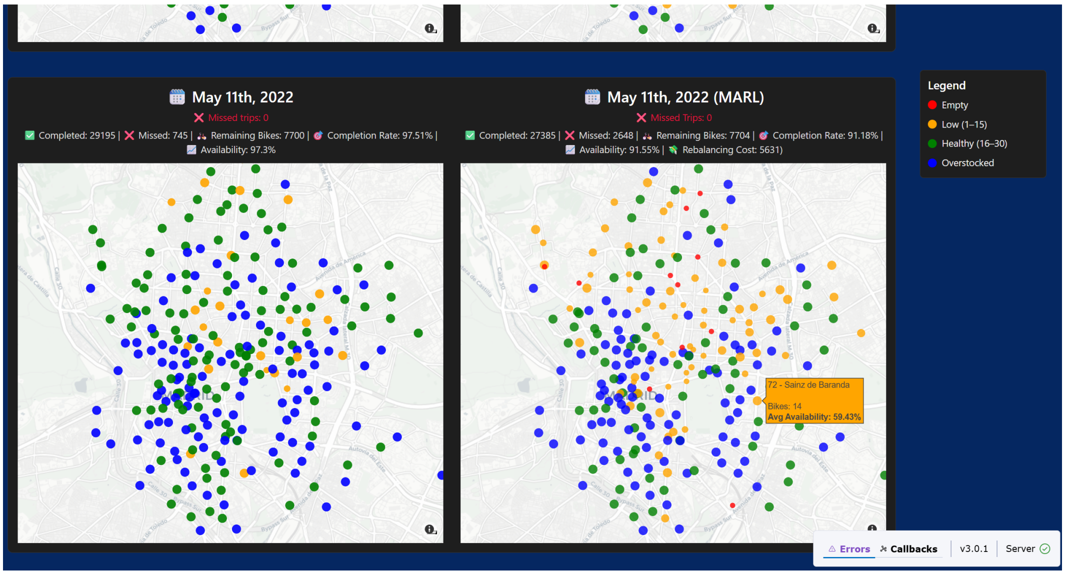

Figure 3.4: Simulation Environment for May 11th at the end of the Simulation

completed trips, the number of missed trips, the overall trip completion rate, the average station availability, and the total rebalancing cost.

The combination of these visualizations and monitoring tools enables a detailed evaluation of the bike-sharing system's operations under various conditions and demand.

3.2.3 Time Management

The simulation recreates a complete 24-hour day of bike sharing operations, providing a detailed representation of trip activities and system dynamics throughout the day. This simulation progresses in a time interval of five minutes in order to balance between realistic runtime and visualization clarity. Because of this speedy time scale, the entire day is compressed and viewed more quickly because the five-minute simulation step is equal to five minutes of real-world time.

A time display at the top of the simulation interface makes it easy for the user to understand how the simulated time is progressing. The display provides real-time context by displaying the current simulated clock time and a progress bar visually indicating how far the simulation has progressed toward the end of the 24 hours. Users can quickly determine how long the simulation will last thanks to these visual signals, which improve the interactive experience and make it easier to monitor system performance over time.

3.3 Multi-Agent Setup Design

This section presents the multi-agent system's design and implementation for bike redistribution. Each bike station within the network is presented as an individual agent, which makes decisions based on observations and local interactions with neighboring stations. These agents operate under a shared Deep Q-Network (DQN) to learn effective redistribution strategies.

By enabling decentralized decision-making, the multi-agent approach attempts to improve the overall system performance by sending or requesting bikes to and from nearby stations. This setup tries to minimize missed trips, reduce unnecessary bike movements, and keep balanced station inventory, using reinforcement learning.

This section covers essential topics, including the definition of agent observations and actions, justification for using a shared DQN model, timing of agent actions during simulation, agent coordination and communication mechanisms, and the design of the reward function that guides agent learning.

3.3.1 Agent Representation

In the context of a bike-sharing optimization system, each bike station is represented as an autonomous agent. An independent decision-making agent observes its surroundings, processes essential information, and takes actions to achieve specific goals. By simulating bike stations as agents, they allow the system to decentralize decision-making, which enables each station to react dynamically to help the overall system's performance.

To make decisions, agents must constantly monitor their surrounding environment and internal state, which gives them the context they need to predict future demand. Without such insights, agents would act blindly, leading to ineffective redistribution.

Each agent's observation space in the system is formed carefully to capture both the current status and the demand trends in the near future. These features include:

- **Current Bike Count** (*current_bike_count*): The actual number of bikes that are at the station at any given time in the simulation.
- **Historical Demand Next Hour** (*historical_demand_next_hr*): Based on past trip data, this shows how many bike departures are expected in the next hour.
- **Historical Inflow Next Hour** (*historical_inflow_next_hr*): Based on past trip data, this shows how many bike arrivals are expected in the next hour.
- **Outgoing Demand Over 5 Hours** (*outgoing_5hr*): The total number of expected bike departures over the next five hours.
- **Empty Station Ratio** (*was_empty_ratio*): The percentage of the simulation time where the station had no bikes in it.
- **Full Station Ratio** (*was_full_ratio*): The percentage of the simulation time where the station was at its maximum capacity.
- **Current Hour** (*current_hour*): An integer representation of the current hour.
- **Previous Action** (*previous_action*): The agent;s most recent action, like "do_nothing" or redistributing bikes.

- **Top Partner Station** (*top_partner_1, top_partner_2, top_partner_3*): The three main partner stations that the agent can work with to arrange bike transfers.

Together, these observed features give each agent complete and valuable information about its current condition and future demand. This provides the basis for an efficient multi-agent coordination system to independently adjust bike distribution, reduce missed trips, and optimize availability at low operational costs.

3.3.2 Action Space

In reinforcement learning, the action space defines the set of all possible actions that an agent can perform. These actions directly influence the environment and the agent's future state. That is why each station agent must make an informed decision about how to intelligently redistribute bikes to maintain a balanced system, minimize missed trips, and prevent overstocking. Therefore, these redistribution options must be easily captured in the action space.

Each station agent has seven different actions in their action space, which are represented by integers from 0 to 6 and present different bike redistribution commands or a lack of action:

- **Action 0: Do Nothing** - The agent decides not to perform any redistribution at this moment, maintaining the current bike count.
- **Action 1 to 3: Send 5 Bikes to Top Partner Stations** - These actions represent sending five bikes to one of the agent's top three partner stations (*top_partner_1, top_partner_2, or top_partner_3*). The partner stations are selected dynamically according to demand and supply characteristics. For instance, action 1 sends bikes to the top partner station 1, action 2 to partner 2, and action 3 to partner 3.
- **Action 4 to 6: Request 5 Bikes from Top Partner Stations** - These actions represent requesting five bikes from one of the agent's top three partner stations. Likewise, action 4 requests bikes from *top_partner_1*, action 5 from *top_partner_2*, and action 6 from *top_partner_3*.

The quantity moved is restricted to a minimum of five bikes, but it carefully checks the station's bike count not to fall below zero, and the receiving station's not to go over its maximum capacity. Additionally, depending on where bikes are most likely to be needed or surplus, each station selects up to three other stations with which to interact. However, if there are not enough suitable stations to engage with, the agent safely chooses the action "Do Nothing" to prevent invalid transfers.

3.3.3 Shared Deep Q-Network Model

A Deep Q-Network (DQN) is a type of neural network used in reinforcement learning to help agents choose the most rewarding actions over time by predicting the importance of performing specific actions. There are two main ways to apply this idea: either each agent trains a separate DQN, or all agents use the same DQN model.

This project presents a shared DQN model for all stations, meaning that instead of maintaining separate models for every bike station, a single neural network is trained using the experiences from all agents. This shared model captures the general demand and redistribution patterns by learning from all stations in the network.

The shared approach improves generalization by allowing the model to recognize and implement redistribution strategies that work across stations, instead of overfitting individual cases. It also speeds up learning because knowledge from one station benefits all others through shared updates. Additionally, because only one network needs to be updated, it improves training effectiveness by lowering computational costs.

3.3.4 Action Execution Timing

Due to the absence of real-world data detailing how bike redistribution occurs overnight to prepare the system for the next day, the multi-agent reinforcement learning system begins each day using the system's end state from the previous day. That is why an equal redistribution of bikes is done between 3:00 AM and 4:00 AM to simulate a realistic system "restart". The Deep Q-Network (DQN) algorithm and the MARL agents do not control this redistribution phase, but it acts as an approximate balancing step to distribute bikes evenly among stations.

The core agent-driven redistribution, where MARL and DQN mechanisms actively learn and make decisions, occurs during off-peak hours between 12:00 PM and 1:00 PM. Agents monitor the system's condition during this hour-long period, where they choose actions to transfer or request bikes and progressively improve their redistribution policy from the system's feedback.

Each redistribution action is modeled with a one-hour delay before the moved bikes reach their destination station, to represent the mechanism of bike movement accurately. This delay reflects the time required to transfer those bikes physically.

Together, these timing strategies provide a more accurate and real-time representation of how the bike-sharing system undergoes daily rebalancing, combining evenly balanced resets with intelligent adaptive redistribution.

3.3.5 Agent Interaction and Communication

In the multi-agent system, individual bike stations operate as agents that coordinate redistribution by selective interactions. Rather than examining all stations in the network, each agent dynamically selects its top three partner stations based on current demand and supply imbalances. These partner stations are selected to maximize the efficiency of bike movements by concentrating on users who will benefit the most from borrowing or lending bikes.

Agents can prioritize redistribution based on locations with surpluses or deficits by selecting partner stations based on their past demand data and current condition. During each decision-making step, agents evaluate their own inventory and demand forecast in comparison to their partners before deciding whether to send bikes, request bikes, or remain idle.

Communication between agents is indirect and decentralized, relying on shared knowledge of partner connections rather than direct message passing. This method enables each agent to make decisions independently while still coordinating with other agents. This partner-based interaction model improves scalability and efficiency by making sure that bike movements make a significant contribution to the system without unnecessary or duplicate relocations.

3.3.6 Reward Function Design

The reward function r of this project is carefully designed to help agents strike a balance between minimizing missed trips and rebalancing costs, while also ensuring effective bike redistribution. It is a complex function that represents important system metrics to evaluate the agent's actions. It is mathematically expressed as follows:

$$
\begin{aligned}
r = & -\text{missed\_weight} \times \text{missed\_trips} && (3.1)\\
& -\text{move\_weight} \times \text{bikes\_moved} && (3.2)\\
& -5 \times \text{overflow\_attempts} && (3.3)\\
& -2 \times (\text{current\_bike\_count} - \text{ideal\_level}) && (3.4)\\
& +\text{zero-miss bonus (added separately)} && (3.5)
\end{aligned}
$$

- **Missed Trips Penalty:** This is the biggest negative portion of the reward function, with a penalty of 50.0 (*missed_weigh*) for a missed trip. A missed trip can occur when a user attempts to rent a bike but none are available, which has a direct impact on system performance and user satisfaction.

- **Bike Movement Cost:** The total number of bikes relocated during the redistribution is penalized less heavily (*move_weight* = 0.005), but still detects pointless moves that result in operational expenses.

- **Overflow Penalty:** If a station attempts to receive more bikes than it can handle (*overflow_attempts*), a penalty of 5 is deducted for each overflow. This keeps the inventory level balanced without resulting in overcrowding stations.

- **Deviation from Ideal Level:** A penalty of 2 is integrated to encourage each station agent to keep its bike inventory close to an optimal target level. It combines short-term immediate demands with long-term supply-demand matching.

- **Zero-Miss Bonus:** This bonus comes additionally at the end of the simulation day. It offers a positive 20 to stations that have no missed journeys during the entire simulation day, and encourages them to continue this trend.

Together, these components allow agents to learn a redistribution policy that strikes a balance between lowering operating costs, maintaining balanced inventory, and maximizing customer satisfaction (by minimizing missed trips).

3.4 Deep Q-Network (DQN) Implementation Details

This section gives a full explanation of the Deep Q-Network (DQN) implementation, which serves as the foundation of the multi-agent reinforcement learning framework. It begins with a detailed description of the neural network architecture, then continues with the training process, including experience replay, target network updates, and epsilon-greedy exploration strategy. Finally, the section provides an explanation and justification for the hyperparameters chosen for training. These components work together to give the agent the capacity to learn and improve the redistribution policies in a dynamic and complex urban environment.

3.4.1 DQN Architecture

This project's Deep Q-Network (DQN) aims to predict the optimal action-value function *Q(s, a)* for each agent, serving as a function approximator. In this context, an agent is equivalent to a bike station, where s indicates the station's current state and a represents a possible action. The network's primary purpose is to predict, for each given state, the expected cumulative future reward for each possible action. This predictive capability enables the agent to make decisions that improve the bike-sharing system's long-term performance.

The input to the DQN consists of a state vector generated by the agent's observations of its environment. This state vector includes several essential features, such as the current number of bikes available at the station, predicted bike demand for the next hour and the next 5 hours, and ratios representing how often the station was empty or full. Additionally, the current hour of the day, the agent's previous action, and identifiers for the top partner stations are also included to guide redistribution decisions.

Architecturally, the network consists of completely linked layers that handle the fixed-length state vector. The network starts with an input layer that is the same size as the dimensions of the state vector, to ensure that all crucial characteristics are captured. This is followed by two hidden layers, each with 128 neurons and using ReLU (Rectified Linear Unit) activation functions. ReLU is chosen because it helps the network learn better by avoiding some common training problems. Having two hidden layers is a good balance to learn complex patterns without making the training too slow or difficult.

The network's output layer provides Q-values for each possible action the agent can take. These values represent future benefits of each activity. The action with the highest Q-value is then selected by the agent, which is expected to produce the best long-term outcomes.

This network design is well-suited to the situation. Because the input is a list of various types of information in no particular sequence, fully connected layers are effective here, enabling the network to understand the relationships between the features. The network learns quickly and consistently when ReLU activation is used and the output layer is in line with reinforcement learning's objective of the quality of each action.

In summary, the DQN network links the agent's observations of the environment to the optimal actions it can take. The network enables each bike station agent to make informed decisions about bike distribution by learning how to estimate the value of each action based on the present condition.

3.4.2 Training Process

The Deep Q-Network (DQN) training method is a crucial component of the project's reinforcement learning setup. It teaches the agents how to make better decisions when redistributing bikes among stations. To learn the most effective redistribution tactics, the agent continuously interacts with the simulation environment. At each decision step, the agent examines the current status of its corresponding bike station, chooses an action based on its policy, and receives a reward for the action's immediate effect. The state, action, reward, and next state are all created by these interactions and saved in a replay buffer.

The replay buffer is important because it allows the agent to learn from a wide range of previous experiences rather than relying only on recent transitions. The DQN parameters are updated periodically by sampling random batches of experiences from the replay buffer.

To further stabilize learning, a separate target network is maintained and updated less frequently than the main one. The predicted Q-values are computed from the main network's predictions for the current states and actions, while the target Q-values use a different target network. The target network provides more stable Q-value targets during training, reducing oscillations and divergence.

An epsilon-greedy (ε-greedy) policy encourages exploration by allowing the agent to pick actions to find new strategies randomly. The epsilon value starts high to promote exploration and eventually decreases to a lower bound, moving the focus to applying the learned strategy as training advances.

Training is organized in episodes corresponding to simulated days. At the end of each episode, the network parameters are updated based on previous experience. This repeated procedure continues until the agent achieves sufficient redistribution performance, which will be indicated by improvements in the evaluation metrics.

3.4.3 Hyperparameters

Several hyperparameters rule the learning dynamics and performance of the DQN in this project. The **Replay Buffer**, which keeps track of previous form experiences (state, action, reward, next state, done), is an essential component in training. This buffer is set at 50,000 experiences and helps to break the correlation between subsequent data by randomly picking batches of experiences during training.

Using batches of 64 samples selected at random from the replay buffer, the DQN refreshes its knowledge. This batch size balances the quality of learning updates and their computing cost. To enable the neural network to efficiently process the input states, actions, rewards, and next states, each batch is transformed into tensors.

Reducing the difference between the target and predicted Q-values is the primary purpose during the training. The target Q-values are calculated based on the immediate reward plus the discounted expected future rewards, with a discount factor (γ) set to 0.99. To prevent overestimating future benefits, the formula takes into account whether or not the episode has ended.

Every 250 training steps, the weights of the main network are copied to the target network. This update delay prevents the target values from changing too quickly, allowing training to remain stable and avoid instabilities.

For optimization, the Adam algorithm is used with a learning rate of 0.000005 (5e-5), because it performs well at training deep networks and has adaptive learning capabilities. To reduce the loss, the gradients are backpropagated through the main Q-network, adjusting its parameters.

The training also uses a **ε-greedy** strategy to strike a balance between exploring new actions and what the agent has learned. The agent initially explores the environment by taking random actions, starting with an exploration rate (ε) of 1.0. Over time, ε gradually drops to a minimum of 0.01 by a factor of 0.9995 with each training step. This allows the agent to rely more on the learning policy as it gets experience.

As the simulation progresses, the agent collects new experiences and periodically updates its policy, resulting in continual training. This allows the agent to improve its decision-making for bike redistribution based on observed system states and past outcomes.

3.5 Evaluation Design

The section describes the framework used to evaluate the performance of the bike-sharing system under different experimental situations. It begins by describing the five scenarios that change the initial quality of bikes per station to determine how inventory levels affect the system. Continuing with the evaluation comparison in the system behavior for two real-world days, under typical and peak demand. Finally, the last section describes the key performance metrics used to measure system effectiveness, allowing objective comparison and analysis across scenarios and days.

3.5.1 Initial Inventory Scenarios: Varying Bike Counts per Station

The evaluation of the proposed bike-sharing redistribution system is conducted using a series of five separate scenarios, each differing from the others in the initial number of bikes allocated per station at the start of the simulation day. This serves as a key factor to analyze how the system performs under baseline conditions of bike availability.

The five possible scenarios start with a bike count of 15, 20, 30, 40, and 50 per station. These values are chosen to represent a range of inventory levels, illustrating situations in a real-world bike-sharing network. The total number of bikes present in the system scales with the initial allocation, as shown in Table 3.5. For example, 15 bikes per station corresponds to around 3,849 total bikes in the system, whereas 50 bikes per station corresponds to approximately 12,844 total bikes.

Total Bikes in the System Based on Initial Bikes per Station

Initial Bikes per Station	Total Bikes in System
15	3849
20	5134
30	7704
40	10274
50	12844

Figure 3.5: Total Bikes in the System Based on Initial Bikes per Station

By systematically adjusting the initial count across these scenarios, the evaluation isolates the influence of fleet size on key system performance indicators. That is why all other parameters, such as demand patterns, redistribution policies, and simulation conditions, remain constant to ensure a fair comparison of observed performance differences to the initial bike allocation.

3.5.2 Evaluation Based on Two Representative Days

To fully analyze the system's performance, the evaluation included two independent real-world demand days that serve as test cases for detecting variations in user behavior and system stress. The targeted dates are May 5th and May 11th, which have different daily trips and provide other operational challenges (Table 3.2).

May 5th is a typical weekday with moderate demand and a normal amount of trips for the whole day. Evaluations on this day offer information on how the redistribution system performs under regular conditions.

In contrast, May 11th is a high-demand day, with about twice the journeys compared to May 5th. This difference in demand shows peak stress conditions where the system is challenged to maintain availability and efficiently rebalance the fleet despite the increased usage.

By analyzing system activity across two contrasting days, the evaluation captures both routine and peak-period performance, testing the robustness and adaptability of the redistribution strategy. This two-day strategy allows a deeper understanding of how the inventory initialization and redistribution algorithm react to changing demand levels, which is important for a real-world scenario.

3.5.3 Evaluation Metrics

In this project, four primary metrics are used to evaluate the performance of the bike-sharing system under different starting conditions: *Missed Trips*, *Trip Completion Rate*, *Availability Rate*, and *Rebalancing Cost*. These metrics provide complementary insights into user satisfaction, system reliability, and operational efficiency across all simulation scenarios.

3.5.3.1 Missed Trips

Missed Trips occur when users attempt to start a journey from a station, but there are no bikes available at that time. This situation directly reflects unmet user demand and is a key sign of system performance failure.

In this project, the system tracks missed trips in two ways:

- **Per-station missed trips:** The number of missed trip attempts recorded individually at each bike station.
- **Total missed trips:** The sum of missed trips across all stations for the entire network during each simulation run.

The primary purpose of bike-sharing system optimization is to reduce missed rides as much as possible. A lower missed trip count means more users are able to start their journeys without delay, increasing overall user satisfaction.

By focusing on missed trips, the system encourages the redistribution algorithm to manage bike supplies and avoid empty stations during peak demand times. Reducing missed trips shows that the redistribution strategy is successful and the system is responsive to user needs.

3.5.3.2 Trip Completion Rate

The Trip Completion Rate, presented as a percentage, is the proportion of user trips that are successfully completed within the bike-sharing system. A trip is considered completed if it successfully starts when bikes are available at the origin station and ends when the bike is docked at the destination. If the origin station does not have any bikes to start a journey, the trip is considered missed or incomplete.

The Trip Completion Rate is calculated as the ratio of successful trips to the total number of attempted trips, expressed as a percentage. The formula is:

$$\text{Trip Completion Rate} = \frac{Number\ of\ Successful\ Trips}{Total\ Number\ of\ Attempted\ Trips} * 100 \tag{3.6}$$

This measure reflects the system's reliability and capacity to meet user demands. A higher completion rate indicated that most customers are able to begin and end their journey without experiencing shortages.

The Completion Rate is essential for evaluating operational strategies, such as the redistribution algorithm and capacity planning, as well as providing insight into the quality of user experience across different scenarios.

3.5.3.3 Availability Rate

The Availability rate is a crucial indicator used to evaluate how well the bike-sharing system keeps bikes available at individual stations and across the entire network. It is expressed as a percentage and calculates both the station and system levels.

The Availability Rate for each station is calculated by dividing the number of bikes currently available by the total capacity of the stations:

$$\text{Availability Rate}_{\text{station}} = \frac{Number\ of\ Bikes\ at\ Station}{Station\ Capacity} * 100 \quad (3.7)$$

The metrics show a station's operational condition at any given moment, and if the rate is close to 100%, it means that the station is well-stocked with bikes, while a low rate indicates the station is empty or almost empty, potentially leading to missed rides.

After Each simulation run, the system-wide Availability Rate is calculated by averaging the availability rates of all individual stations:

$$\text{Availability Rate}_{\text{system}} = \frac{1}{N}\sum_{i=1}^{N} \text{Availability Rate}_{\text{station}_i} \quad (3.8)$$

where N is the total number of stations.

The system-wide Availability Rate provides a comprehensive overview of how well the entire bike-sharing network meets user demand for bikes and docking spaces throughout the day. A lower rate suggests widespread shortages or overcrowding, while a higher rate indicates a more balanced and user-friendly system.

By monitoring Availability Rate at both station and system levels, the system may detect problematic stations that need to be redistributed and evaluate overall fleet management efficiency.

3.5.3.4 Rebalancing Cost

Rebalancing Cost is a key operational parameter that measures the overall effort and expense involved in redistributing bikes across stations to keep the system balanced. In a bike-sharing system, rebalancing requires guaranteeing that stations have enough bikes, especially during peak hours or after periods of uneven usage.

In this project, Rebalancing Cost is calculated as the total number of bike movements performed by the system's redistribution mechanism across the fixed period (between 12 and 13 PM). Each bike movement from one station to another adds to the overall cost, indicating the physical effort, time, and resource investment (such as trucks, labor, and fuel) required to maintain optimal bike distribution.

A lower rebalancing cost shows a more efficient system that minimizes operational expense while ensuring adequate bike availability. In contrast, a high cost indicates frequent and perhaps expensive bike relocation, which may increase operational cost and reduce overall system efficiency.

Balancing between cost and service quality indicators, such as missed trips and availability rate, is critical. The goal is to minimize missed trips and maximize availability while keeping the rebalancing cost at a sustainable level.

Throughout the simulation scenarios, these metrics help to examine the trade-offs between operational expenses and user satisfaction, guiding decisions on optimal initial allocation and redistribution strategies.

3.6 Tools and Technologies

This section provides an overview of the main tools, libraries, frameworks, and environments used to design, develop, and evaluate the bike-sharing system. These technologies are selected based on their capacity to handle large-scale data, support reinforcement learning algorithms, enable effective simulation, and allow clear visual outcomes.

3.6.1 Programming Language

Python is chosen as the primary programming language because of its adaptability, readability, and huge ecosystem for scientific computing and machine learning. Python's simple syntax speeds up development cycles and integrates various libraries. In addition, its broad use in the AI field guarantees access to cutting-edge resources, which is critical for quick experimentation and problem-solving.

3.6.2 Data Manipulation

Pandas library offers a strong data structure, including DataFrames, to effectively load, preprocess, and analyze the large amount of trips and station data generated during the simulations. Its simple syntax allows complex data transformations, grouping, and collection, which is required for generating evaluation criteria, like missed trips and station availability.

NumPy improves the system by providing high-performance arrays with many dimensions and mathematical functions, which are necessary for vectorized tasks such as representation construction, reward calculations, and batch processing during reinforcement learning training.

3.6.3 Reinforcement Learning

PyTorch is used to develop the basic Deep Q-Network (DQN) model, which controls the multi-agent bike redistribution approach. Its dynamic computation graph allows for flexible model design and experimentation with network architectures. The use of a GPU improves the processing efficiency of a large replay buffer and enables faster training iterations.

Additionally, PyTorch's integration with Python allows debugging and model inspection, making it easier to modify hyperparameters and analyze learning behavior.

3.6.4 Visualization

Dash, a Python framework for developing web apps, is used to create interactive visualizations that dynamically display simulation results. Combined with Plotly's rich plotting capabilities, it helps to create maps that illustrate station locations, real-time bike movements, and redistribution processes. These visual tools are both handy for qualitative and quantitative research, allowing for observation of system behavior, identifying bottlenecks, and verifying the impact of redistribution strategies over the two simulation days.

3.6.5 Development Environment

Visual Studio Code (VS Code) is the primary integrated development environment used for coding, testing, and debugging. Its support for Python development includes IntelliSense for code completion, an integrated terminal, version control integration with Git, and debugging tools, helping to speed the development process. VS Code's flexible system also makes code formatting and cleaning easier, keeping the code quality and maintainability throughout the project development.

3.7 Summary

This chapter describes the approach to solving the problem of bike redistribution in a city bike-sharing system. The study contrasts two primary strategies: a simple model in which bikes move naturally with user demand, and a smarter model, which uses historical data to move bikes during off-peak hours. The central component of this study is a multi-agent Deep Q-Network (DQN) system, in which every bike station functions as an independent agent learning to make better decisions depending on its current situation.

The chapter explains how to set up the simulation using actual real data from Madrid's bike sharing system. It covers how the data is cleaned and prepared, how the simulation environment shows bike movement visually, and how time is managed to ensure the simulation runs smoothly.

Essential parts of the multi-agent system are detailed, including how agent observe their stations, the different actions they take, and how all agents learn together with a shared DQN model. The reward function, designed to reduce missed trips and unnecessary bike movements, the timing of actions, and how agents communicate are also described in a detailed way.

The evaluation part also presents how the system is tested on two days with five different starting numbers of bikes per station. Finally, the chapter lists the main tools and technologies to build and run the project.

Overall, this chapter sets the stage for understanding the design and testing of the bike redistribution system. The next chapter will show the results and discuss how well the approach works in practice.

Chapter 4

Results and discussion

This chapter presents and analyzes the key performance outcomes of the bike-sharing optimization framework in five distinct initial-inventory scenarios. Each scenario varied only in the number of bikes assigned to each station at the beginning of the simulated day, specifically examining the configurations of 15, 20, 30, 40, and 50 initial bikes per station. By keeping all other model parameters and operational conditions constant, system-wide evaluation metrics-such as trip completion rate, missed-trip count, average station availability rate, and total rebalancing cost, were evaluated effectively.

The motivation for this multi-scenario approach is twofold. First, in a real-world bike-sharing system, determining how many bikes to pre-position at each station significantly influences both user satisfaction (through bike availability) and operational expense (through the cost of fleet maintenance and rebalancing). Too few bikes per station can lead to a high number of unachieved rental requests early in the day, or too many can increase the rebalancing cost and the number of idle bikes. By systematically adjusting the inventory levels in increments of five (from 15 to 50 bikes), a controlled experimental framework was created, which illustrates the trade-offs between under-stocking and over-stocking.

Second, the stability and adaptability of the suggested multi-agent DQN-based redistribution algorithm were confirmed by assessing performance across different starting levels. The simulations in each scenario went through two main rebalancing phases: an equal-spread adjustment during the early morning hours (3 AM–4 AM) and an intelligent agent-driven redistribution during the midday off-peak hours (12 PM–1 PM). The observed performance differences could be safely defined as differences in the initial inventory alone because rebalancing rules and customer demand patterns were constant in all scenarios.

Section 4.1 details the numerical and graphical results for Scenario 1, in which each station begins with 15 bikes. We report the total number of completed and missed trips, the overall trip-completion percentage, average station availability, cumulative rebalancing cost, and qualitative heat-map visualizations of station statuses at key time points. Section 4.2 replicates the same analysis for Scenario 2 (starting inventory = 20 bikes), and Sections 4.3, 4.4, and 4.5 follow suit for 30, 40, and 50 initial bikes per station, respectively. Section 4.6 (“Comparative Analysis”) synthesizes the findings across all five scenarios, highlighting the starting inventory with optimally balanced missed-trip minimization, availability maximization, and controlled rebalancing overhead. Finally, Section 4.7 provides a concise summary of the key findings, identifying the scenario that achieves the highest operational efficiency under the specified demand profile and algorithmic strategy.

4.1 Scenario 1: 15 Bikes per Station

In Scenario 1, each station begins the simulation day with 15 bikes, which represents the lowest initial inventory scenario among those analyzed. The number of missed trips is particularly high on May 11th, the highest-demand day, where the trips surpassed 7,000, indicating severe under-stocking (Table 4.1). Although there are some improvements as the simulation continued, the missed trips reached more than 11,000 at their highest point. For May 5th, the number is significantly lower, averaging around 2,000, suggesting that even under moderate demand levels, 15 bikes per station are unable to meet user requirements.

The completion rate on May 11th starts significantly lower, where it began from around 70%, showing difficulties in bike supply, and improved to 75-80% over time. However, on May 5th, the dynamic was similar to the busiest day, but the results were higher, showing the system was ranging between 87-92% after the first 30 simulation runs.

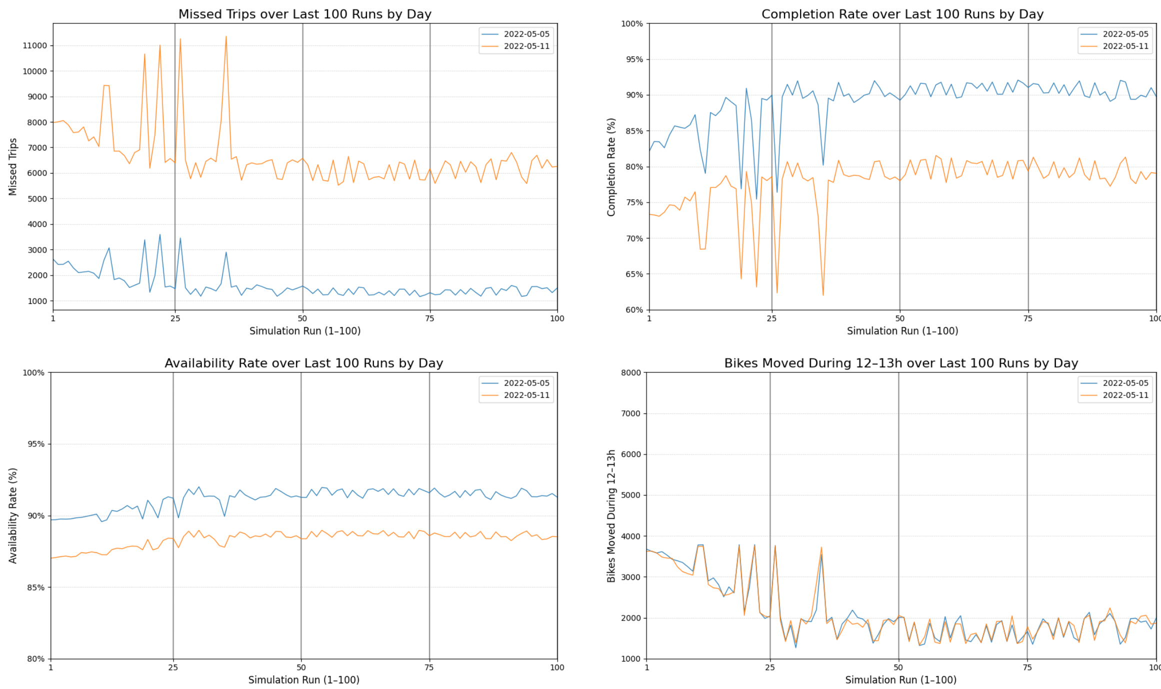

Table 4.1: Performance Evaluation for Scenario 1 (15 Bikes per Station)

The availability rate is steady for all the simulation runs for both days, but on May 5th, the availability is slightly better, between 90-92%. While on May 11th, the results are staying between

87-88%. This pattern suggests frequent reduction in the number of bikes, which is directly related to an increase in missed trips and decreased customer satisfaction.

Regarding rebalancing operations, Scenario 1 experiences a high movement of bikes initially, often close to 4,000 bikes moved, but as the simulations went on, the number of bikes transported decreased to between 1,500 and 2,000 bikes for both days. While this indicates adaptation, the high volume of bikes transferred still reflects significant operational costs.

In conclusion, Scenario 1 clearly highlights the limitations of starting the simulation with an initial bike count of 15 bikes per station. Despite the redistribution algorithm's capability to partially adapt over time, the results show constant operational inefficiencies, high missed trips, poor availability, and increased rebalancing costs.

4.2 Scenario 2: 20 Bikes per Station

In Scenario 2, each station began the simulation day with 20 bikes. On May 5th, missed trips already rapidly stabilized, remaining below 1,000 per simulation run, demonstrating that the DQN algorithm is learning effectively under normal demand (Table 4.2). In contrast, May 11th initially showed high unpredictability, with over 6,000 missed journeys. Despite a noticeable decrease in the following runs, missed trips stabilized at 4,000, showing that the initial allocation of 20 bikes per station is not enough to meet peak demand without significant redistribution efforts.

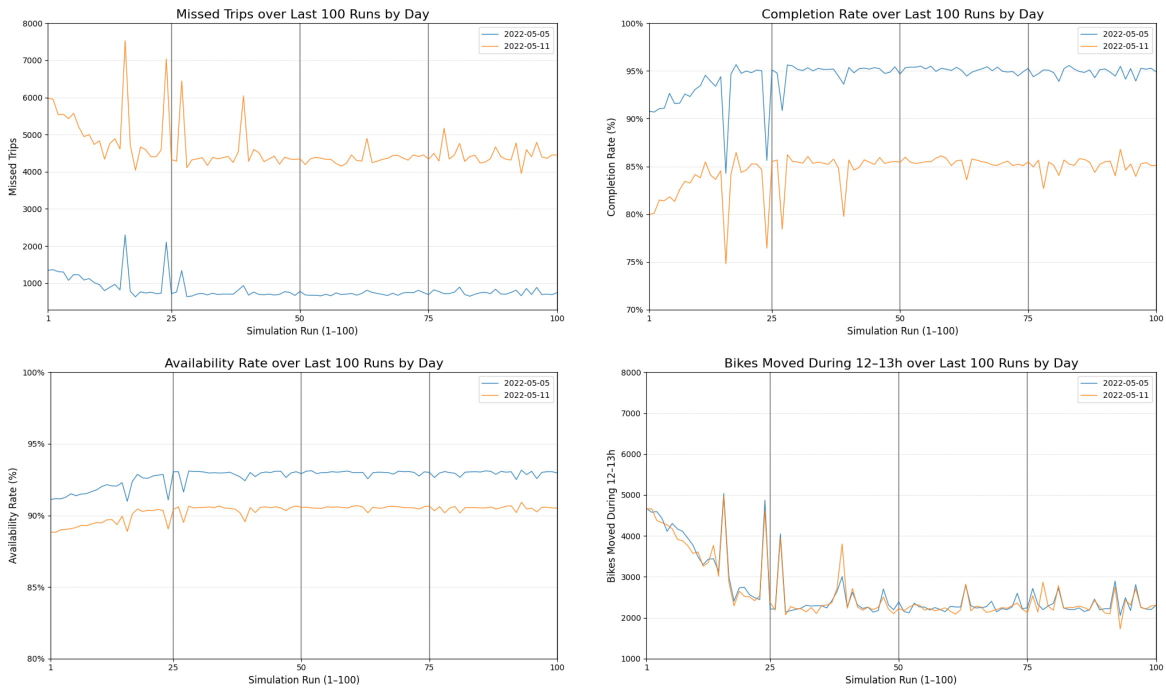

Table 4.2: Performance Evaluation for Scenario 2 (20 Bikes per Station)

Completion rate went over 90% on May 5th and increased to a stabilized level around 95%. At the same time, station availability rate stayed high, often increasing from 92% to 94% as simulations went on. This shows that a starting stock of 20 bikes per station can lead to a high trip completion and station availability rate. On the other hand, the system first started having trouble on May 11 due to increased demand. The completion rate is balanced between 75-85%, but it manages to keep it around 85% over time. In line with this, station availability started out low at about 88% and only slightly improved to about 90%. These limited improvements underlined the inventory constraints during periods of high demand.

In terms of the redistribution cost, both days experienced high redistribution volumes (5,000-6,000 bikes). However, with continuous simulation runs, the cost decreased to about 2,000 bikes per day, which reflected the successful learning of the system.

Scenario 2 effectively managed moderate demand, achieving high completion rates and stable availability. However, under peak-demand scenarios, the initial fleet size proved insufficient, reflected in a higher amount of missed trips and a lower station availability rate. Despite these challenges, the DQN system demonstrated robust learning capability.

4.3 Scenario 3: 30 Bikes per Station

The analysis shown in Table 4.3 of missed trips across 100 simulation runs between May 5th and May 11th shows a significant improvement compared to Scenario 2. On May 5th, the missed trips consistently remained below 1,000, stabilizing around 500 and 700 missed trips per day, which shows the improvement in the system's performance, with the increase in the initial bikes. On the busier days, May 11th, missed trips ranged between 1,500 and 2,500. While still higher than May 5th, this scenario outperforms the previous one, where the trips spiked above 4,000.

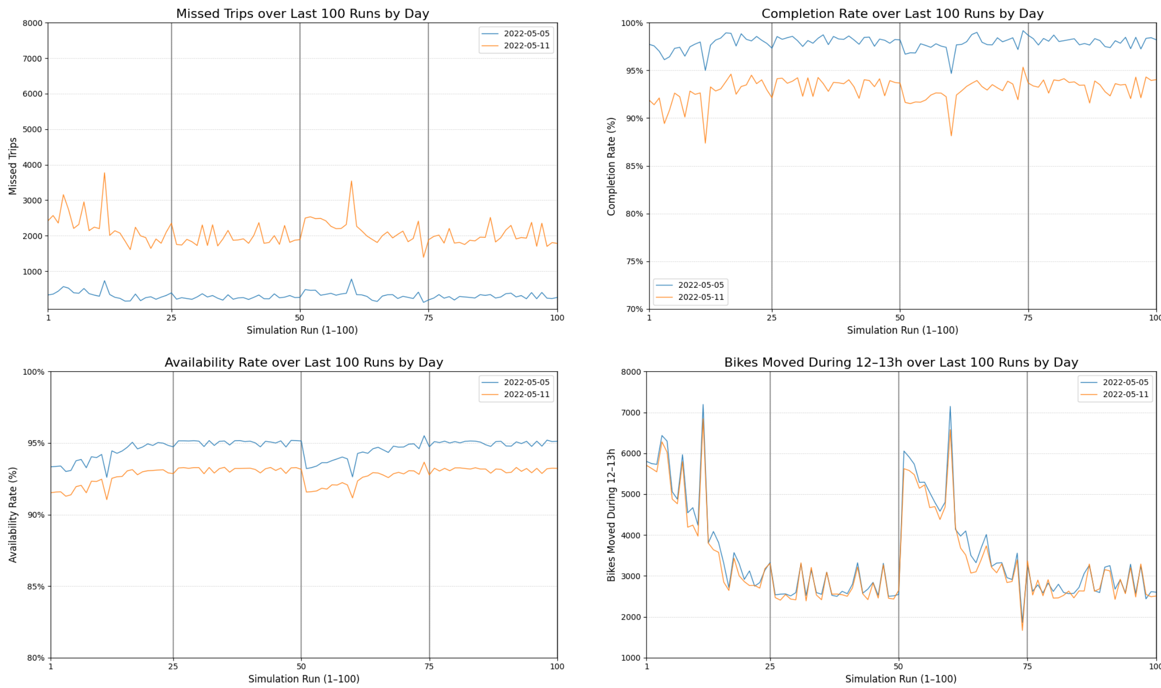

Table 4.3: Performance Evaluation for Scenario 3 (30 Bikes per Station)

The trip completion rate reflects the improvement in missed trips. For May 5th, the completion rate remained consistently high, averaging around 97-98% with rate dips slightly below. On May 11th, the completion rate is between 92-94%, significantly improved compared to Scenario 2, where the completion rate is often around 85%. Thus, starting with 30 bikes per station gives stability and reliability even on high-demand days.

The station's average availability across simulations remained consistently strong. For both days, the results remain almost similar, with a slight difference in the values, where on May 5th, availability goes around 94-94%, indicating excellent station bike-stock management. On May 11th, they are varying between 91% and 93%, which demonstrates well-managed inventory despite customer demand.

The number of bikes moved during the midday redistribution (12-13 hours) was originally very variable but eventually stabilized after 25 and 75 simulation runs, resulting in a decrease in bike cost to a range of around 2,000-3,000 bikes per day on both days. Compared to the scenario with 20 bikes per station, this scenario requires fewer redistributions once stabilized, which suggests a beneficial trade-off.

Overall, starting each station with 30 bikes strikes a good balance, drastically reducing missed trips, increasing completion rate, maintaining high availability, and simultaneously lowering redistribution cost compared to earlier scenarios. These results clearly demonstrate the value of increasing inventories to better handle intense demand, especially on days like May 11th.

4.4 Scenario 4: 40 Bikes per Station

The scenario with 40 bikes per station displays a significant improvement in the system's ability to handle user demand. In Table 4.4 on May 5th, the number of missed trips is consistently low, often less than 400, indicating the system's outstanding performance for normal conditions. Even on the highest-demand day (May 11th), missed trips remained approximately between 800- 1,000 per day, far lower compared to previous scenarios. This indicates a robust and stable redistribution policy, achieved by the model.

Correspondingly, the completion rate remained high throughout the simulated runs. For both days, the results are all above 95%, and even for May 5th, they are very close to 100%, indicating strong system stability. The minor changes seen across the repeated runs highlight the robustness of the redistribution strategy when the system is first configured with the number of bikes per station.

The availability rate was likewise continuously great in this situation, showing average results around 96% for May 5th and slightly over 94% for May 11th. The reduced variability ensures high bike availability for users even in peak demand periods.

The rebalancing cost, calculated as the number of bikes moved between 12:00 and 1:00 PM, initially began higher, with over 5,000 bikes, but quickly stabilized between 1,500 and 3,000 bikes per run, resulting in low operational costs across simulations. The decreasing trend and the stabilization demonstrate that the DQN learns efficiently over time, minimizing unnecessary bike movements.

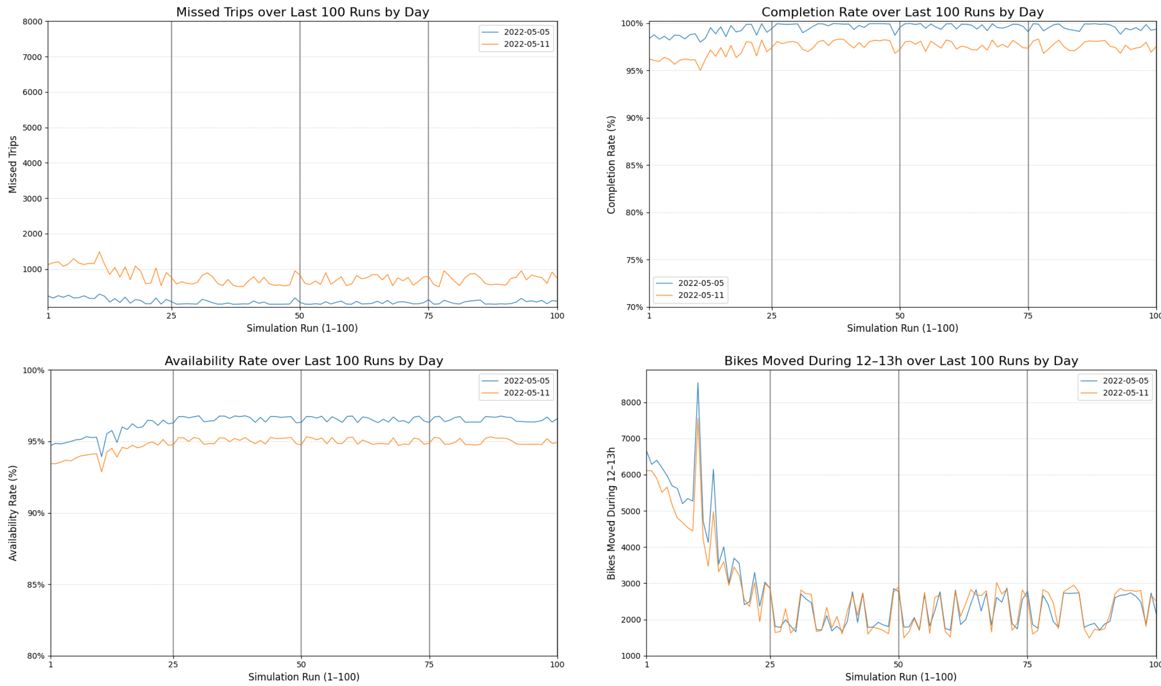

Table 4.4: Performance Evaluation for Scenario 4 (40 Bikes per Station)

Overall, initializing stations with 40 bikes achieves an effective combination, significantly improving the system's performance in terms of low missed-trip counts, high trip completion rate, stable bike availability, and reasonable rebalancing costs.

4.5 Scenario 5: 50 Bikes per Station

The missed trips graph shows a significant reduction in missed trips for both days, May 5th and May 11th, compared to scenarios with lower initial inventories (Table 4.5). The number of missed journeys swings around a lower baseline, where the numbers go below 1,000 for both days, and for May 5th, even under 300 missed trips per day. These improvements suggest that a higher starting inventory considerably lowers the risk of stations running out of bikes, which is essential for the system.

Completion rates for both days are noticeably high, approaching or achieving close to 100% on May 5th and on May 11th remaining under 97%. The improved availability of bikes at the start contributes to these higher completion rate percentages, showing that riders can typically start and finish their trips successfully.

Station availability rates stay continuously high throughout the simulations, maintaining levels above 95% for both days, with May 5th showing higher availability, increasing to 97%. This consistent availability highlights the system's effectiveness, ensuring that bikes are accessible to users while avoiding unnecessary oversupply or shortage.

The number of bikes moved during the midday rebalancing period first increases, but after 25 simulation runs, it settles into a constant range. This behavior demonstrated that the multi-agent DQN effectively learns to manage bike redistribution, balancing between moving enough bikes to reduce missed trips and minimizing unnecessary relocation cost.

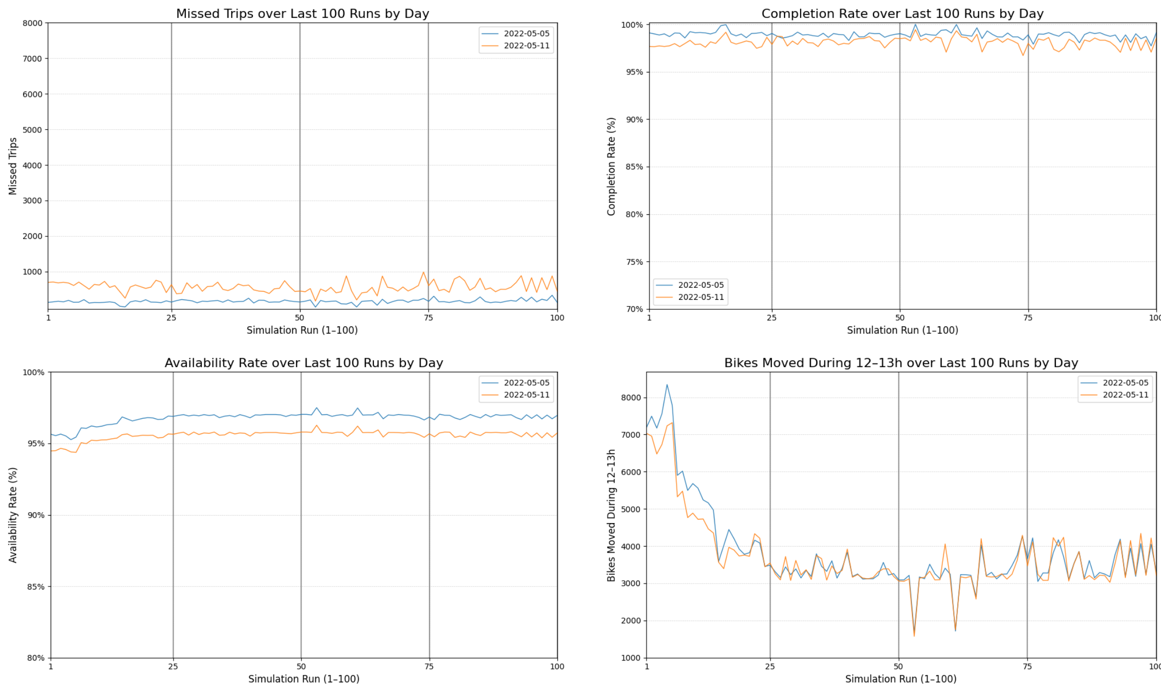

Table 4.5: Performance Evaluation for Scenario 5 (50 Bikes per Station)

This analysis shows that Scenario 5, with 50 bikes initially per station, achieves the best operational balance among the tested configurations. It successfully reduces missed trips while increasing trip completion and availability rate, although at the cost of higher rebalancing activities. The results highlight the conflict between initial stock levels and operational efficiency, underscoring the need for data- driven optimization in real- world bike- sharing systems.

4.6 Comparative Analysis

This section compares five initial inventory scenarios, introduced earlier, varying only in the number of bikes assigned per station at the start of the simulation day. Key metrics are examined, like missed trips, trip completion rate, availability rate, and operational cost across the system, to understand their operational trade-offs and benefits. The analysis helps to understand how initial bike allocation impacts system efficiency and user experience under different demand conditions.

The scenario with the lowest beginning inventory of 15 bikes per station regularly has the most missed trips, particularly on May 11th, which represents the highest demand day of the whole year. Missed trips in this setup often exceed 7,000 and approach 11,000, showing significant under-stocking issues and a failure to meet user demand effectively. In contrast, the scenario with 50 starter bikes has the lowest missed trips count, often remaining below 1,000 even during peak demand periods, indicating a much enhanced ability to fulfill user requests.

The trip completion rate follows a complementary trend, improving progressively with increased initial bike stock. The 15-bike scenario struggles to keep completion rates over 75-80% on the high-demand day, while the 50-bike scenario consistently achieves near-perfect completion rates on May 5th and around 97% on May 11th. Intermediate scenarios, such as those with 20, 30, and 40 bikes, display steady increases in trip completion, demonstrating the positive effect of higher starting stock on system reliability.

Station availability rates also correspond with the initial quantity of bikes. The 15 and 20 bike scenarios have lower availability, ranging from 88- 93%, showing regular shortages or overcrowding. In contrast, the 30, 40, and 50 bike scenarios maintain robust availability, typically above 90%, which supports improved user satisfaction.

The rebalancing cost, measured as the number of bikes relocated during the midday redistribution, indicates a significant factor connected to the original inventory size. While scenarios with fewer bikes per station, suck as the 15 bike configuration, result in lower number of bikes moved (1,500 - 2,000), this represents a disproportionately high operational cost in relation to the total fleet size. As the initial inventory increases, the total number of bikes in the system grows proportionally, which naturally raises the absolute number of bikes transported during the redistribution. For example, the 50 bike scenario involves moving up to 5,000 bikes during redistribution periods. However, over time, the DQN-based redistribution stabilizes its rebalancing activities across all scenarios, proving its ability to learn efficient redistribution techniques while balancing operating costs and service quality.

4.7 Summary

In conclusion, the complete study of five various initial inventory scenarios demonstrates the critical influence of starting bike count on the operational performance and user satisfaction of the system. Lower initial stocks, like 15 or 20 bikes per station, result in higher missed trip rates and poorer trip completion and availability percentages, especially during peak demand periods. However, higher inventories, ranging from 30 to 50 bikes per station, show improvements in service reliability, trip completion rate, and station availability. These improvements result in an increase in rebalancing activities in absolute terms. However, this must be interpreted in relation to the increased fleet size, where redistribution becomes more effective. The results underscore the inherent trade-offs between operating cost and user service quality, to carefully find the right balance. This balance provides both efficient resource management and high levels of customer satisfaction, demonstrating the effectiveness of the multi-agent DQN-based redistribution approach.

Chapter 5

Conclusions

The primary goal of this project is to successfully integrate a multi-agent Deep Q-Network algorithm into a bike-sharing system in order to maximize on-demand bike redistribution during off-peak hours. A redistribution framework based on reinforcement learning is developed to achieve the goal, and it is tested in five different scenarios with varying initial bike stocks per station (15, 20, 30, 40, and 50 bikes).

The simulations show that the multi-agent DQN algorithm learns to adapt over time, consistently improving the system performance with continuous training across all scenarios. This adaptability confirms the algorithm's capacity to handle redistribution challenges in the environment.

Within the tested scenarios, starting with 30 and 40 bikes per station provides the optimal balance between system reliability and operational cost. These scenarios demonstrated fewer missed trips, higher trip completion rates, and robust station availability, proving that the algorithm efficiently relocates bicycles to meet demand. An appropriate starting fleet size avoids frequent stockouts and rebalancing costs associated with extremely high inventory levels, which results in slight improvements in these mid-range scenarios.

Overall, the multi-agent DQN framework integration shows promising results, with the algorithm continuously learning to optimize bike redistribution and adapt to demand conditions. The study highlights how useful reinforcement learning is in managing actual urban mobility systems, while underscoring the importance of adjusting bike allocation to maximize both user satisfaction and operational efficiency.

5.1 Main Contribution

This work makes a significant contribution to the field of intelligent urban mobility by demonstrating how multi-agent reinforcement learning can be effectively applied to the bike redistribution problem. A significant component of the project is the creation of a multi-agent Deep Q-Network model that treats each bike station as an independent agent.

The study also provides a complete simulation environment based on actual demand data from Madrid's bike-sharing system. By analyzing system performance under various scenarios, the project highlights important insights into how fleet size and redistribution policies affect operational outcomes.

Another contribution is the creation of a well-balanced reward function that pushes agents to reduce missed trips, avoid wasteful bike movements, and maintain an appropriate inventory level. This encourages efficient redistribution strategies that improve the overall service quality.

Finally, the project provides practical information regarding the optimal fleet sizing, showing an excellent balance between service reliability and cost efficiency, specifically around 30 to 40 bikes per station. These results contribute to the practical implementation of reinforcement learning in urban bike-sharing systems.

5.2 Future Work

This study effectively integrates a multi-agent DQN algorithm for bike redistribution and shows that it works across various initial inventory scenarios, but there are still a number of areas that could use improvement.

Adding real-time dynamic demand data is a promising direction to increase the accuracy and responsiveness of the system. Future work could expand the current model to incorporate real-time data streams like traffic disruptions, weather, and specific events. These factors often cause unpredictable swings in bike demand that are not captured by the historical demand patterns used in the current study.

Another crucial next step is to move from simulation to real-world implementation through field deployment and pilot testing. Collaborating with an operational bike-sharing provider to deploy the multi-agent DQN system in a real-world setting can validate simulation results under actual operational constraints. When confronted with real user behavior and infrastructure challenges, pilot testing would provide invaluable insights into the system's performance. Furthermore, the algorithm may be improved with the gathered feedback during deployment.

Additionally, future studies might examine strategies for station expansion and adaptive bike capacities. A more extensive approach to system optimization would include mechanisms in the simulations to modify station locations and capacities dynamically. This includes adding, removing, and relocating stations to meet higher or lower demand in certain locations across the city. This would improve operational effectiveness and service quality.

Furthermore, some limitations of the data used in this study provide opportunities for future work. The absence of an exact total bike count and detailed station capacity information led to making assumptions, which may affect the accuracy of the metrics. Future research might focus on gathering more precise data or developing methods for dynamically determining these factors.

In order to support more innovative and sustainable urban mobility solutions, these future scenarios collectively seek to improve the intelligent bike-sharing system's scalability, resilience, and practical applicability.

References

[1] E. Alaoui and S. Tekouabou. “Intelligent management of bike sharing in smart cities using machine learning and Internet of Things”. In: *Sustainable Cities and Society* (2021). URL: https://www.sciencedirect.com/science/article/abs/pii/S2210670720309161.

[2] V. Albuquerque, M. Sales Dias, and F. Bação. “Machine Learning Approaches to Bike-Sharing Systems: A Systematic Literature Review”. In: *ISPRS International Journal of Geo-Information* 10.2 (2021), p. 62. URL: https://www.mdpi.com/2220-9964/10/2/62.

[3] J. Bae et al. “A data-driven agent-based simulation of the public bicycle-sharing system in Sejong city”. In: *Simulation Modelling Practice and Theory* (2024). URL: https://www.sciencedirect.com/science/article/abs/pii/S1569190X23001387.

[4] N. Boufidis et al. “Development of a station-level demand prediction and visualization tool to support bike-sharing systems’ operators”. In: *Transportation Research Procedia* (2020). URL: https://www.sciencedirect.com/science/article/pii/S2352146520302593.

[5] D. Bravo et al. “Can user data improve Bike Sharing Systems demand forecasting?” In: *IEEE Transactions on Intelligent Transportation Systems* (2021). URL: https://ieeexplore.ieee.org/document/9435497.

[6] Bycyklen. *Rent a bike in Copenhagen*. Official website available at https://bycyklen.dk/rent-bike-copenhagen/. 2025.

[7] D. Chemla, F. Meunier, and R. Wolfler Calvo. “Bike sharing systems: Solving the static rebalancing problem”. In: *Discrete Optimization* 10 (2013), pp. 120–146. URL: https://www.sciencedirect.com/science/article/pii/S1572528612000771.

[8] Yiheng Chi. “Self-Organized Bike Redistribution in Urban City”. In: *Open Journal of Modelling and Simulation* 6.3 (2018), pp. 93–101. URL: https://www.scirp.org/journal/paperinformation?paperid=87622.

[9] F. Chiariotti et al. “A Dynamic Approach to Rebalancing Bike-Sharing Systems”. In: *Sensors* 18.2 (2018), p. 512. URL: https://www.mdpi.com/1424-8220/18/2/512.

[10] M. Cipriano, L. Colomba, and P. Garza. “A Data-Driven Based Dynamic Rebalancing Methodology for Bike Sharing Systems”. In: *Applied Sciences* 11.15 (2021), p. 6967. URL: https://www.mdpi.com/2076-3417/11/15/6967.

[11] European Commission. *Energy Union and Climate Action: Driving Europe’s Transition to a Low-Carbon Economy*. Press Release available at https://ec.europa.eu/commission/presscorner/detail/en/ip_16_2545. 2016.

[12] C. Contardo, C. Morency, and Louis-Martin Rousseau. *Balancing a Dynamic Public Bike-Sharing System*. Tech. rep. CIRRELT - Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation, 2012. URL: https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2012-09.pdf.

[13] Paul DeMaio. *The Meddin Bike-sharing World Map*. The Bike-Sharing World Map available at https://bikesharingworldmap.com/index.php#/all/2.7/0/51.5/. 2007.

[14] The Guardian. *Story of cities #30: how this Amsterdam inventor gave bike-sharing to the world*. Article available at https://www.theguardian.com/cities/2016/apr/26/story-cities-amsterdam-bike-share-scheme. 2016.

[15] Earth Policy Institute. *Bike-Sharing Programs Hit the Streets in Over 500 Cities Worldwide*. Article available at https://earthpolicyinstitute.wordpress.com/2013/04/25/bike-sharing-programs-hit-the-streets-in-over-500-cities-worldwide/. 2013.

[16] P. Jing et al. “Agent-Based Simulation of Autonomous Vehicles: A Systematic Literature Review”. In: *IEEE Access* 8 (2020), pp. 79089–79103. URL: https://ieeexplore.ieee.org/document/9078053.

[17] Kaggle. *Madrid Bike Sharing Dataset*. Usage data for Madrid’s public bike sharing service "BiciMad" in 2022. Available at https://www.kaggle.com/datasets/carlosborrajo/madrid-bike-sharing-dataset. 2022.

[18] P. Leclaire and F. Couffin. “Method for Static Rebalancing of a Bike Sharing System”. In: *16th IFAC Symposium on Information Control Problems in Manufacturing*. Vol. 51. IFAC, June 2018, pp. 1561–1566. URL: https://www.sciencedirect.com/science/article/pii/S2405896318313983.

[19] G. Li et al. “Towards smart transportation system: A case study on the rebalancing problem of bike sharing system based on reinforcement learning”. In: *Journal of Organizational and End User Computing* (2021). URL: https://www.igi-global.com/gateway/article/full-text-pdf/276375.

[20] X. Liu and K. Pelechrinis. “Excess Demand Prediction for Bike Sharing Systems”. In: *PLoS ONE* 16.6 (2021). URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0252894.

[21] Á. Lozano et al. “Multi-Agent System for Demand Prediction and Trip Visualization in Bike Sharing Systems”. In: *Applied Sciences* 8.1 (2018), p. 67. URL: https://www.mdpi.com/2076-3417/8/1/67.

[22] M. Lu et al. “Improving the sustainability of integrated transportation system with bike-sharing: A spatial agent-based approach”. In: *Sustainable Cities and Society* (2018). URL: https://www.sciencedirect.com/science/article/abs/pii/S2210670718304700.

[23] A. Maleki et al. “Developing a supervised learning-based simulation method as a decision support tool for rebalancing problems in bike-sharing systems”. In: *Expert Systems with Applications* (2023). URL: https://www.sciencedirect.com/science/article/abs/pii/S0957417423014859.

[24] T. Mátrai and J. Tóth. “Comparative Assessment of Public Bike Sharing Systems”. In: *Transportation Research Procedia* 14 (2016), pp. 2344–2351. URL: https://www.sciencedirect.com/science/article/pii/S2352146516302678?via%3Dihub.

[25] Patrick R. McMullen. “A Markov Simulation Approach to Balancing Bike-Sharing Systems”. In: *American Journal of Operational Research* 9.1 (2019), pp. 12–17. URL: https://scholar.google.com/scholar_lookup?title=A%20markov%20simulation%20approach%20to%20balancing%20bike-sharing%20systems&author=P.%20McMullen&publication_year=2019&pages=12-17.

[26] J. Pucher and R. Buehler. “Comparative Assessment of Public Bike Sharing Systems”. In: *Transport Reviews* 37.6 (2017), pp. 689–694. URL: https://www.tandfonline.com/doi/full/10.1080/01441647.2017.1340234?scroll=top&needAccess=true.

[27] Z. Shi et al. “Multi-Platform dynamic game and operation of hybrid Bike-Sharing systems based on reinforcement learning”. In: *Transportation Research Part E: Logistics and Transportation Review* (2024). URL: %5Curl%7Bhttps://www.sciencedirect.com/science/article/abs/pii/S1366554523003629%7D.

[28] F. Soriguera, V. Casado, and E. Jiménez. “A simulation model for public bike-sharing systems”. In: *Transportation Research Procedia* (2018). URL: https://www.sciencedirect.com/science/article/pii/S2352146518302412.