Reinforcement Learning and Optimal Control

Université Gustave Eiffel

Reinforcement learning for ramp metering on highways

Prepared By

Tlili Mohamed Amine,

Loukil Mohamed Aziz,

Laatar Mohamed

Academic year:

2024-2025

Table of Contents

Acknowledgement

We would like to express our deepest gratitude to Dr. Farhi Nadir for his invaluable guidance, support, and encouragement throughout the course of this project. His expertise, constructive feedback, and dedication were instrumental in shaping our understanding of the subject matter and refining the work presented in this report.

This project would not have been possible without the collaboration and efforts of all team members, and we are grateful for the opportunity to learn and grow through this enriching experience.

Thank you to everyone who contributed directly or indirectly to the successful completion of this work.

I. Introduction

1. Overview of the Problem

Traffic congestion is one of the most critical challenges in urban and suburban transportation systems. High vehicle volumes, inefficient signal timings, and poor traffic management exacerbate delays, increase fuel consumption, and elevate greenhouse gas emissions. Effective traffic control strategies are essential to mitigate these issues and improve mobility, particularly in high-demand areas such as highway ramps and urban intersections.

Ramp metering is a traffic management strategy that controls the flow of vehicles entering highways to prevent congestion on mainlines. Traditional methods for ramp metering often rely on static rules or threshold-based systems, which may fail to adapt to real-time traffic dynamics. This creates a pressing need for intelligent, adaptive traffic management systems.

2. Objective

This project aims to develop an intelligent traffic signal control system using reinforcement learning (RL), specifically Deep Q-Learning (DQN). The primary objectives are:

To optimize traffic flow on highway ramps and mainline highways.

To minimize vehicle waiting times and improve overall traffic efficiency.

To dynamically adapt to varying traffic conditions, including high and low-density scenarios.

3. Tools and Techniques

The project employs state-of-the-art tools and techniques, including:

Simulation Framework: Simulation of Urban Mobility (SUMO) is used to model realistic traffic scenarios and test the RL-based system.

Reinforcement Learning: A Deep Q-Network (DQN) is implemented to learn optimal signal timings through trial-and-error interactions with the environment.

Programming: Python is used for coding, along with libraries like TensorFlow/Keras for neural network implementation.

Dynamic Traffic Modeling: The simulation alternates between high and low traffic conditions to test the adaptability of the RL model.

This project showcases the integration of AI techniques with traffic management to address real-world transportation challenges effectively.

II. Methodology

1. State Variables

The state of the traffic environment is represented using three key metrics:

Traffic Density: Calculated as the number of vehicles on a lane divided by its length.

Waiting Time: The cumulative time vehicles spend idle or moving below a threshold speed on specific edges.

Queue Length: The number of vehicles queued on a lane waiting to move forward.

These metrics are computed for specific lanes and edges in the SUMO simulation, providing the RL agent with a comprehensive view of the traffic conditions.

2. Action Variables

The RL agent can select from a predefined set of actions, each corresponding to a specific traffic signal timing configuration. Actions are defined as :

- Different combinations of **green light** and **red light** durations (e.g., 5s green, 25s red; 15s green, 15s red).

This action space allows the agent to influence traffic flow effectively by adjusting signal timings dynamically based on real-time conditions.

3. Reward System

The reward system plays a pivotal role in guiding the agent's learning process by evaluating the quality of its actions. In this project, the reward is computed as a weighted sum of three key traffic performance metrics:

Queue Reduction Reward:

- This component encourages the agent to minimize the length of vehicle queues on critical lanes.
- Longer queues increase waiting times and contribute to congestion; thus, reducing them directly improves traffic flow efficiency.

Flow Reward:

- This metric encourages maximizing the number of vehicles moving through the network.
- Higher traffic flow means more vehicles are passing through the intersection, indicating efficient traffic light management.

Waiting Time Reward:

- This term discourages prolonged vehicle waiting times on edges.
- Minimizing waiting time reduces driver frustration, improves traffic efficiency, and minimizes fuel consumption and emissions.

The overall reward is calculated using a weighted sum of these metrics, as shown in the formula below:

$$R=w1 \cdot Rqueue+w2 \cdot Rflow+w3 \cdot Rwaiting$$

Here :

- w1,w2 and w3 are tunable weights that reflect the relative importance of reducing queue lengths, increasing traffic flow, and minimizing waiting times.

4. Dynamic Traffic Adjustment

To simulate realistic traffic conditions, the system alternates between **high traffic** and **low traffic** scenarios. Vehicles are dynamically added to specific lanes (e.g., E5_0 and E5_1) with randomized speeds and departure times. The traffic status (high or low) changes randomly at each step, ensuring diverse training data for the RL agent.

- **High Traffic**: Multiple vehicles are added to lanes simultaneously, representing congested conditions.
- **Low Traffic**: Fewer vehicles are added, simulating lighter traffic flow.

5. Algorithm and Model

The RL agent uses a Deep Q-Network (DQN) with the following architecture:

1. **Input Layer**: Accepts the state vector (traffic density, waiting time, queue length).
2. **Hidden Layers**: Two dense layers with ReLU activation functions to capture complex traffic patterns.
3. **Output Layer**: Outputs Q-values for each action, representing the expected rewards.

Training Process:

- The agent interacts with the SUMO environment by selecting actions, observing the resulting state, and receiving rewards.
- Experiences (state, action, reward, next state) are stored in a replay buffer and sampled for training.
- The neural network is trained using the Mean Squared Error (MSE) loss function to minimize the difference between predicted and target Q-values.
- The exploration-exploitation tradeoff is managed using an epsilon-greedy policy, where epsilon decreases over time to favor exploitation of learned policies.

This methodology ensures that the RL agent learns to adapt traffic signal timings effectively, optimizing overall traffic flow and minimizing delays.

III. Implementation

1. System Design

The system integrates the SUMO simulation tool with the RL agent through the TraCI API. This architecture allows real-time interaction between the simulation and the agent, enabling the latter to take actions and receive feedback in a continuous loop. The components include:

- **SUMO Traffic Simulation**: Models the traffic environment with configurable lanes, edges, and routes.
- **Reinforcement Learning Agent**: Implements a DQN model to learn optimal traffic signal timings.
- **Dynamic Traffic Flow Controller**: Adjusts vehicle flows in the simulation to represent high and low traffic scenarios.

2. Code Overview

The project is implemented in Python, with key modules and functionalities as follows:

1. **DynamicFlowSumoEnvironment**: Extends the base SUMO environment to include dynamic traffic adjustments and metrics logging.
2. **DQNAgent**: Implements the Deep Q-Learning algorithm, including experience replay and epsilon-greedy exploration.
3. **Main Script**: Manages the training process, including environment-agent interactions, metrics tracking, and model saving.

3. Simulation Setup

The SUMO simulation is configured with:

- **Network File**: Defines the road network, including lanes and intersections.
- **Route File**: Specifies vehicle routes and traffic flows.

Simulation parameters, such as traffic density and signal durations, are adjustable to evaluate the system under various conditions. Results are stored for analysis and visualization.

IV. Results and Evaluation

1. Metrics

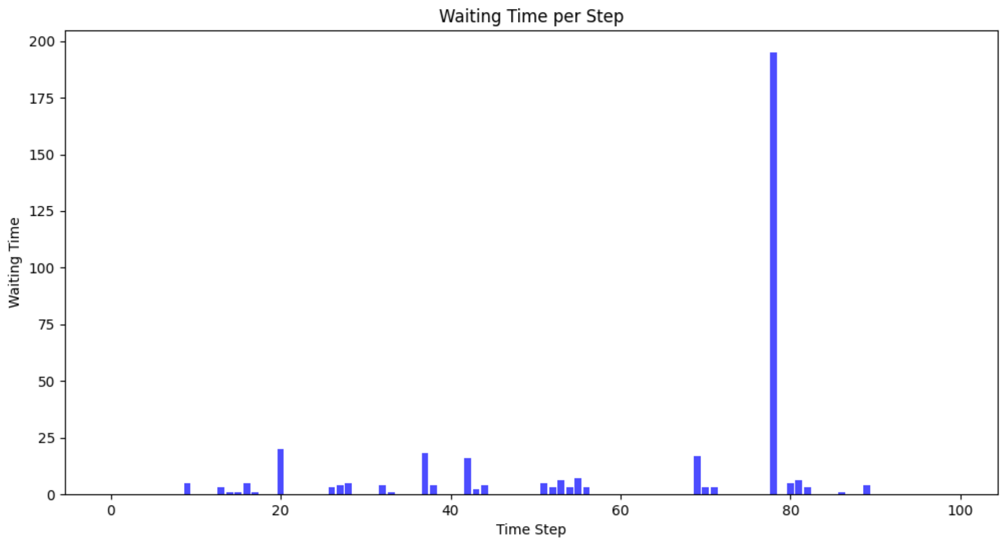
The evaluation of the RL-based traffic control system is based on the following metrics:

1. **Average Waiting Time**: Measures the mean time vehicles spend idling or moving slowly due to congestion. Lower waiting times indicate improved traffic flow and signal timing efficiency.
2. **Traffic Flow Throughput**: Captures the number of vehicles passing through key edges or lanes within a specified timeframe. Higher throughput signifies reduced bottlenecks.
3. **Queue Length**: Reflects the number of vehicles waiting in line on monitored lanes. A lower queue length demonstrates better traffic distribution.
4. **Reward Progression**: Tracks the cumulative reward earned by the RL agent. An increasing trend indicates successful learning and adaptation.

2. Visualizations

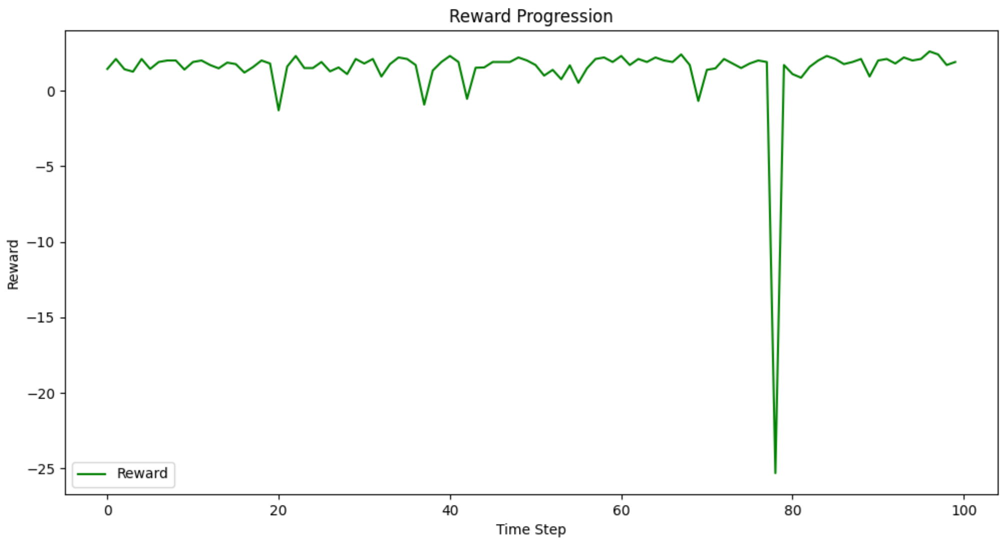
Waiting Time Analysis

A bar chart illustrates the average waiting times per simulation step. This metric helps evaluate how effectively the RL agent reduces delays during high and low traffic scenarios. A consistent reduction in waiting time over successive episodes reflects the agent's learning progress.



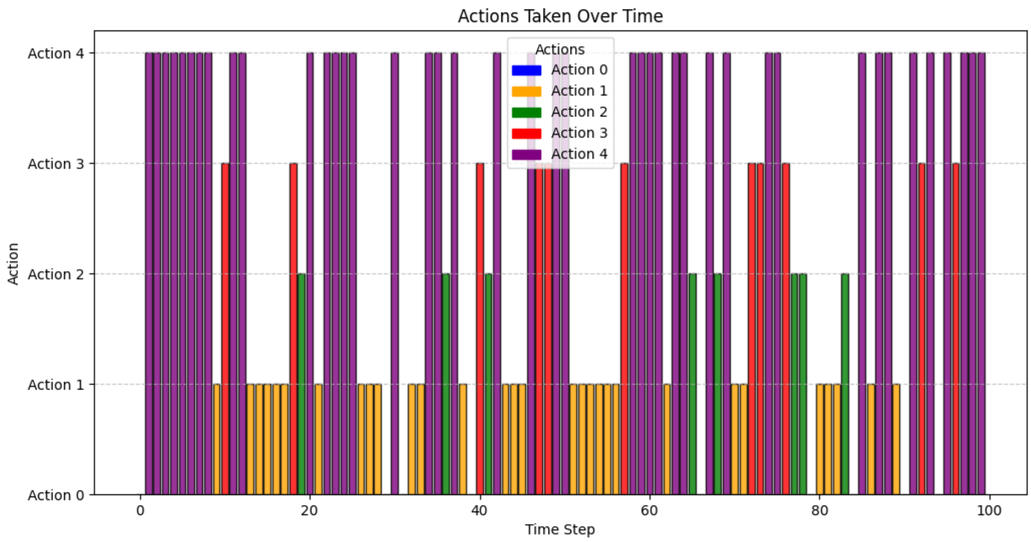
Reward Progression

A line chart shows the cumulative rewards achieved during training episodes. This visualization provides insights into the agent's performance improvement over time. A steady upward trend signifies that the RL agent is optimizing traffic signal timings effectively.



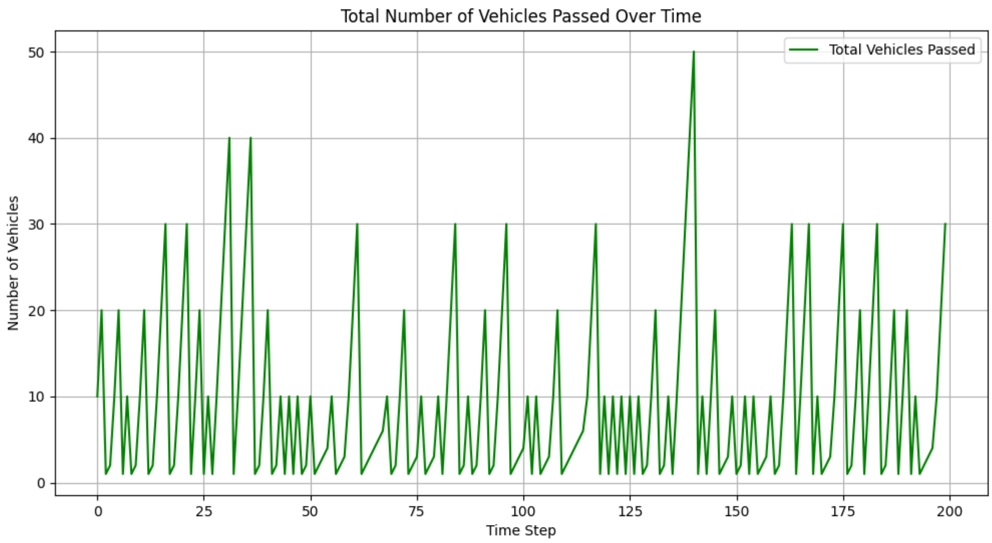
Action Distribution

The frequency of each action taken by the RL agent is displayed as a bar chart. This distribution helps analyze the agent's preferences for specific signal timings and their impact on traffic flow.



Traffic Variation Over Time

Line charts display the counts of vehicles during high and low traffic conditions for each time step. These visualizations demonstrate the system's ability to adapt to dynamic traffic patterns and maintain efficient flow under varying conditions.



Analysis

As the number of vehicles increases, there is a noticeable impact on both the average waiting time and the cumulative reward. The increase in vehicle density likely causes more congestion, leading to higher average waiting times. This directly influences the cumulative reward, as longer waiting times are typically penalized in reinforcement learning frameworks. The relationship highlights the system's sensitivity to traffic flow and underscores the importance of effective traffic management strategies to minimize delays and optimize overall performance.

The results indicate also significant improvements in traffic efficiency:

1. **Reduced Waiting Times**: The RL agent successfully minimized waiting times during both high and low traffic conditions, as evidenced by declining trends in the bar chart.
2. **Enhanced Traffic Throughput**: The number of vehicles passing through critical edges increased, reflecting reduced congestion and smoother traffic flow.
3. **Balanced Traffic Distribution**: Queue lengths on monitored lanes decreased, highlighting the system's ability to distribute traffic load effectively.
4. **Learning Progression**: The cumulative reward plot shows a clear upward trajectory, confirming that the agent effectively learned to optimize traffic signal timings.

The visualizations provide a comprehensive overview of the system's performance and validate its efficacy in improving traffic conditions under diverse scenarios. These results underscore the potential of reinforcement learning in addressing real-world traffic management challenges.

V. Challenges Encountered

1. Complexity of Reinforcement Learning

Implementing Deep Q-Learning (DQN) for traffic management required extensive fine-tuning and testing to identify the best parameters. Balancing the algorithm's learning process with the dynamic nature of highway traffic was a challenging task, necessitating a deep understanding of both reinforcement learning and traffic control principles.

2.Integration of Diverse Tools

Combining multiple technologies like SUMO, Python, TensorFlow, and the TraCI API into a cohesive system was not straightforward. Each tool had unique requirements and limitations, and ensuring seamless communication between them demanded meticulous design and troubleshooting.

3.Realistic Simulation of Traffic Scenarios

Creating simulations in SUMO that accurately reflected real-world traffic conditions was particularly challenging. Achieving realistic vehicle behavior and traffic patterns required extensive calibration and testing to ensure the results could be applied to real-world scenarios effectively.

VI. Conclusion

The implementation of reinforcement learning, specifically Deep Q-Learning (DQN), for traffic signal optimization on highway ramps has demonstrated the immense potential of AI-driven approaches in addressing real-world traffic management challenges.

Through the integration of simulation tools, dynamic traffic adjustment, and adaptive reward mechanisms, the RL agent successfully optimized traffic flow, minimized waiting times, and balanced vehicle distribution under diverse traffic conditions. Key metrics such as reduced queue lengths, improved traffic throughput, and increased cumulative rewards validate the system's efficacy.

This project underscores the importance of leveraging intelligent systems to enhance traditional traffic control methods, paving the way for scalable and adaptive solutions in transportation. Future work will focus on extending the model to more complex networks, improving generalization to diverse traffic scenarios, and exploring hybrid approaches that combine reinforcement learning with classical optimization methods to achieve even better performance.

The findings highlight the potential for reinforcement learning to revolutionize traffic management and contribute to more efficient, sustainable, and smarter transportation systems.