Research Project Documentation: Reinforcement Learning-Based Adaptive Traffic Signal Control

This documentation provides a comprehensive overview of the research project that developed and evaluated a reinforcement learning-based traffic signal control system for the Hitech City Junction in Hyderabad.

Project Overview

The research addresses the critical problem of urban traffic congestion by developing an intelligent traffic signal control system that adapts to real-time traffic conditions. Traditional fixed-time traffic signals operate on preconfigured schedules and cannot respond to fluctuating traffic patterns, resulting in excessive delays, fuel consumption, and congestion. This project proposes a Deep Q-Network (DQN) based reinforcement learning solution that learns optimal signal timing strategies through interaction with the traffic environment.

Key Innovation: The system treats traffic signal control as a Markov Decision Process (MDP), where an RL agent observes traffic states and learns to minimize total delay through optimal signal phase transitions[1].

Technical Architecture

Simulation Environment

The project utilizes SUMO (Simulation of Urban Mobility) as the primary simulation platform, integrated with real-world data from OpenStreetMap to ensure realistic modeling of the Hitech City Junction. The simulation environment includes:

- **Network Construction**: Base road network created using Java OpenStreetMap Editor (JOSM)
- **Data Processing**: OpenStreetMap data converted to SUMO-compatible format using netconvert utility
- **Simulation Parameters**: 3600 simulated seconds (1 hour) runtime with TraCI Python interface for real-time control

State Representation

The RL agent observes traffic states through a tuple of vehicle queue lengths on lanes controlled by each traffic light:

```
state = (q_len_lane1, q_len_lane2, ..., q_len_laneN)
```

Action Space

The system implements a discrete action space with two possible actions[1]:

- **Action 0**: Maintain current traffic light phase
- **Action 1**: Switch to next traffic phase

Reward Function

The reward mechanism is designed to minimize vehicle waiting times:

```
reward = -sum(queue_lengths)
```

This negative reward structure penalizes long queues and idle green phases, promoting efficient vehicle movement[1].

Implementation Details

Deep Q-Network Configuration

The DQN implementation uses the following hyperparameters[1]:

- **Learning Rate (α)**: 0.1
- **Discount Factor (γ)**: 0.9
- **Exploration Rate (ε)**: 0.1 (epsilon-greedy strategy)

Q-Learning Update Rule

The Q-values are updated using the standard Bellman equation:

```
Q[state][action] += α * (reward + γ * max(Q[next_state]) - Q[state][action])
```

Controller Integration

Both baseline and RL controllers were implemented as separate Python modules using the TraCI API, enabling dynamic interaction with the SUMO simulation environment.

Performance Results

The experimental evaluation demonstrates significant improvements of the DQN-based controller over traditional fixed-time systems[1]:

Metric	Fixed-Time Controller	DQN Controller	Improvement
Average Waiting Time	~81,000s	~68,000s	~16% reduction
Average Travel Time	~640s	~530s	~17% reduction
Teleportation Events	~340	~290	~15% reduction
Throughput	~1180 vehicles	~1180 vehicles	Maintained

The results show that the RL-based system reduces average waiting times and queue lengths by up to 13-15% while maintaining consistent throughput.

System Requirements

Software Dependencies

- **SUMO v1.23.1**: Traffic simulation platform
- **Python**: For TraCI interface and controller implementation
- **Java OpenStreetMap Editor (JOSM)**: For network construction
- **OpenStreetMap**: Real-world geographic data source[1]

Hardware Considerations

The system requires computational resources for:

- Real-time Q-learning updates
- Traffic state observation and processing
- Simulation environment execution

Deployment Considerations

Advantages

- **Adaptive Control**: Responds dynamically to changing traffic conditions
- **Data-Driven**: Learns optimal policies through experience
- **Scalable**: Can be extended to multiple intersections
- **Performance**: Significant improvements in key traffic metrics[1]

Limitations

The research identifies several challenges for real-world deployment:

- **Generalization**: May struggle with anomalies like accidents or sensor failures
- **Sim-to-Real Gap**: Simulations don't fully capture real-world complexity
- **Deployment Costs**: High computational requirements and integration challenges with legacy systems

Future Work and Extensions

The project establishes a foundation for intelligent traffic signal systems with potential extensions including:

- **Multi-Agent Coordination**: Extending to network-level control across multiple intersections
- **Robustness Enhancement**: Improving handling of unexpected traffic scenarios
- **Real-World Validation**: Field testing and deployment studies
- **Integration**: Compatibility with existing traffic management infrastructure

Research Contributions

This project makes three primary contributions to the field of intelligent transportation systems[1]:

1. **Realistic Modeling**: SUMO-based simulation of Hitech City junction using OpenStreetMap data
2. **Custom RL Formulation**: DQN-based approach with tailored state, action, and reward design
3. **Performance Validation**: Comprehensive evaluation demonstrating significant improvements over conventional methods

The research advances the practical deployment of AI-driven traffic signal systems in metropolitan environments, providing a data-driven alternative to traditional fixed-time control systems.

Vishwas Chakilam – 22951A67H9

G.V. Pavan Gadepalli – 22951A67G7

G.V. Aravind Deepak – 22951A67G6

⁂