Train Network Optimization using Reinforcement Learning: A Generic Environment for Schedule Management

Umar Aslam

January 2, 2025

Abstract

We present a generic reinforcement learning environment designed for train network optimization and real-time schedule management. The environment provides a flexible framework capable of handling various network topologies, operational constraints, and optimization objectives. The system incorporates delay recovery mechanisms, conflict resolution, and schedule adherence monitoring. This paper details the mathematical foundations, implementation architecture, and visualization capabilities of the environment, demonstrating its utility for both research and practical applications in railway operations.

Contents

1 Introduction

Railway networks represent complex systems where multiple objectives - punctuality, energy efficiency, passenger comfort, and operational feasibility - must be balanced. Traditional optimization approaches often struggle with the real-time nature of railway operations, where decisions must be made quickly in response to delays and disruptions. This work introduces a reinforcement learning environment that addresses these challenges through a flexible, modular approach.

2 Problem Formulation

2.1 Network Model

Consider a railway network represented by a directed graph $G = (V, E)$, where:

- $V = \{v_1, ..., v_n\}$ represents the set of stations
- $E = \{e_{ij}\}$ represents the track sections between stations

Each edge e_{ij} is characterized by:

$$e_{ij} = (r_{ij}^{min}, r_{ij}^{max}, h_{ij}) \tag{1}$$

where:

- r_{ij}^{min}: Minimum running time between stations i and j
- r_{ij}^{max}: Maximum allowable running time
- h_{ij}: Minimum headway time on section (i, j)

2.2 Train Dynamics

For each train k, the movement between consecutive stations i and j is governed by:

$$t_j^{arr,k} = t_i^{dep,k} + r_{ij}^k \tag{2}$$

$$t_j^{dep,k} = t_j^{arr,k} + d_j^k \tag{3}$$

where:

- $t_j^{arr,k}$: Arrival time of train k at station j
- $t_i^{dep,k}$: Departure time of train k from station i
- r_{ij}^k: Running time of train k between stations i and j
- d_j^k: Dwell time of train k at station j

2.3 Delay Recovery Model

The environment incorporates a delay recovery mechanism through two key parameters:

$$\beta_w \in [0,1] : \text{waiting time recovery factor} \tag{4}$$

$$\beta_s \in [0,1] : \text{running time recovery factor} \tag{5}$$

For a delayed train, the actual running time $\tilde{r}_{ij}^k$ is:

$$\tilde{r}_{ij}^k = r_{ij}^{min} + (1 - \beta_s)\delta_{ij}^k + \epsilon_{ij}^k \tag{6}$$

where:

- δ_{ij}^k: Initial delay on section (i, j)
- ϵ_{ij}^k: Random disturbance term

3 Reinforcement Learning Framework

3.1 State Space

The state space $\mathcal{S}$ combines multiple components:

$$\mathcal{S} = \mathcal{S}_{loc} \times \mathcal{S}_{time} \times \mathcal{S}_{nbr} \times \mathcal{S}_{win} \times \mathcal{S}_{viol} \tag{7}$$

where:

- $\mathcal{S}_{loc} \in \{0,1\}^n$: One-hot encoded current station
- $\mathcal{S}_{time} \in [0,1]$: Normalized time
- $\mathcal{S}_{nbr} \in \mathbb{R}^{4m}$: Nearby train positions (m trains)
- $\mathcal{S}_{win} \in \mathbb{R}^2$: Departure time window
- $\mathcal{S}_{viol} \in \{0,1\}$: Constraint violation flag

3.2 Action Space

The action space $\mathcal{A}$ consists of two continuous variables:

$$\mathcal{A} = [0,1]^2 \tag{8}$$

An action $a = (a_r, a_d)$ determines:

$$r_{actual} = r_{min} + (1 - a_r)(r_{max} - r_{min}) \tag{9}$$

$$d_{actual} = d_{min} + (1 - a_d)(d_{max} - d_{min}) \tag{10}$$

3.3 Constraints

3.3.1 Headway Constraints

For consecutive trains k and l:

$$t_j^{arr,k} - t_j^{arr,l} \geq h_{min} \quad \forall j \in V \tag{11}$$

$$t_j^{dep,k} - t_j^{dep,l} \geq h_{min} \quad \forall j \in V \tag{12}$$

3.3.2 Capacity Constraints

For each station j with capacity C_j:

$$\sum_{k \in T_j(t)} 1 \leq C_j \quad \forall t \tag{13}$$

where $T_j(t)$ is the set of trains at station j at time t.

3.4 Reward Function

The reward function combines multiple objectives:

$$R(s, a, s') = w_1 R_{delay} + w_2 R_{energy} + w_3 R_{comfort} + w_4 R_{feasibility} \tag{14}$$

Component definitions:

$$R_{delay} = -\sum_{k,j} |t_j^{act,k} - t_j^{plan,k}| \tag{15}$$

$$R_{energy} = -c_1(a_r + a_d) \tag{16}$$

$$R_{comfort} = -c_2|\Delta a_r| - c_3|\Delta a_d| \tag{17}$$

$$R_{feasibility} = -c_4 \sum_{v \in V} p_v \tag{18}$$

where:

- w_i: Weight factors
- c_i: Scaling constants
- V: Set of constraint violations
- p_v: Penalty for violation v

4 Implementation Architecture

4.1 Environment Structure

Algorithm 1 Environment Step Function

```
Require: Action $a = (a_r, a_d)$
1: Calculate actual running and dwell times
2: Check feasibility of proposed times
3: if feasible then
4:     Update train position and times
5:     Calculate reward components
6:     Update state
7: else
8:     Apply infeasibility penalty
9:     Restore previous state
10: end if
11: return new state, reward, done, info
```

4.2 Key Components

- State Manager: Handles state updates and transformations
- Constraint Checker: Validates operational constraints
- Delay Generator: Simulates random disturbances
- Schedule Tracker: Monitors schedule adherence
- Visualization Engine: Generates time-space diagrams

5 Visualization Tools

The environment includes built-in visualization capabilities:

5.1 Time-Space Diagrams

- Planned vs. actual trajectories
- Conflict detection markers
- Station occupancy visualization
- Delay propagation analysis

5.2 Performance Metrics

- Punctuality statistics
- Energy consumption profiles

- Constraint violation analysis
- Delay recovery effectiveness

6 Experimental Results

[Include your specific experimental results, graphs, and analysis here]

7 Conclusion

This paper presented a comprehensive reinforcement learning environment for train network optimization. The environment's key features include:

- Flexible network topology support
- Realistic delay recovery modeling
- Comprehensive constraint handling
- Built-in visualization tools
- Modular architecture for extensions

8 Future Work

Potential extensions include:

- Multi-agent optimization capabilities
- Integration of passenger flow models
- Energy consumption optimization
- Real-time rescheduling strategies
- Integration with existing railway management systems