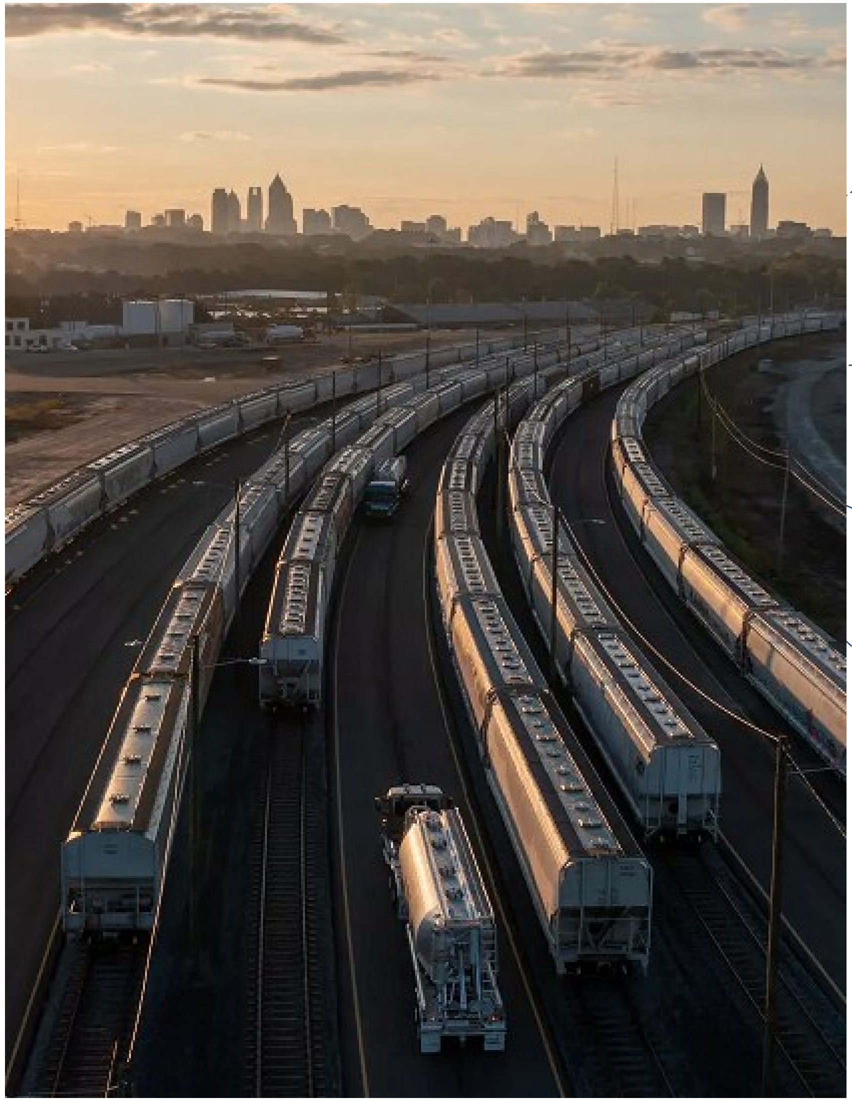
RL-based Rolling Stock Rostering (consider maintenance)

Ruifan Tang
PhD Student in Institute for Transport Studies
University of Leeds

Railway Problem and Motivation



Objective of the PoC
Explore the feasibility of applying RL approaches
Into Rolling Stock maintenance routing problem

Constraints / Requirements
Minimize the overall usage of rolling stock units
Consider the accumulated travelling mileage

Main Issues and Challenges
Data Availability & Acquisition
NP-hard problem if there are too many services to fulfill

Key Performance Indicators
Computation Resources
Usage of minimum rolling stock units

Proof-of-Concept as a Benchmark

AI Application
Adversarial Search

AI Related Disciplines
Reinforcement Learning

AI Techniques
Semi-supervised - Agent Based Modelling
Supervised – Deep Q-Network Learning
Self-Training / Optimization

Inspiring Solutions
Typical maintenance routing (heuristics)
TSP (Travelling Salesman Problem)
CVRP (Capacitated Vehicle Routing Problem)
Intelligent Rolling Stock Rostering

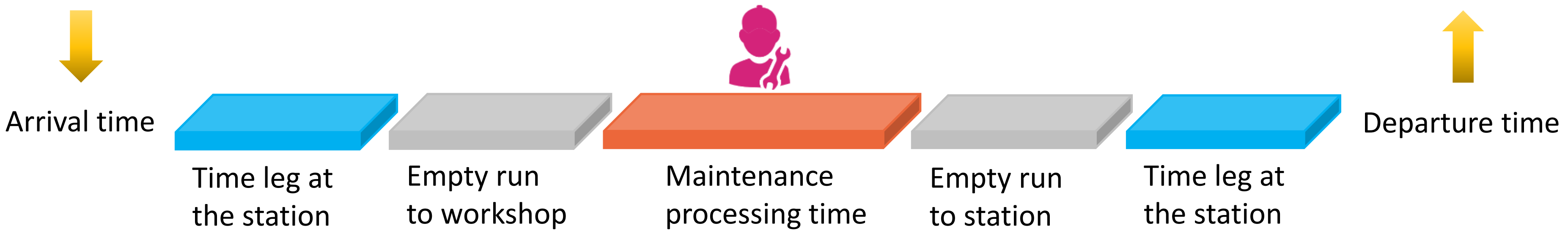
Datasets
TransPennine Express
Real-world timetable data
Synthetic Data

Developments / Implementations
Creating the RS routing/rostering environment
Designing the Q-Learning Algorithm as needed
Writing the training loop

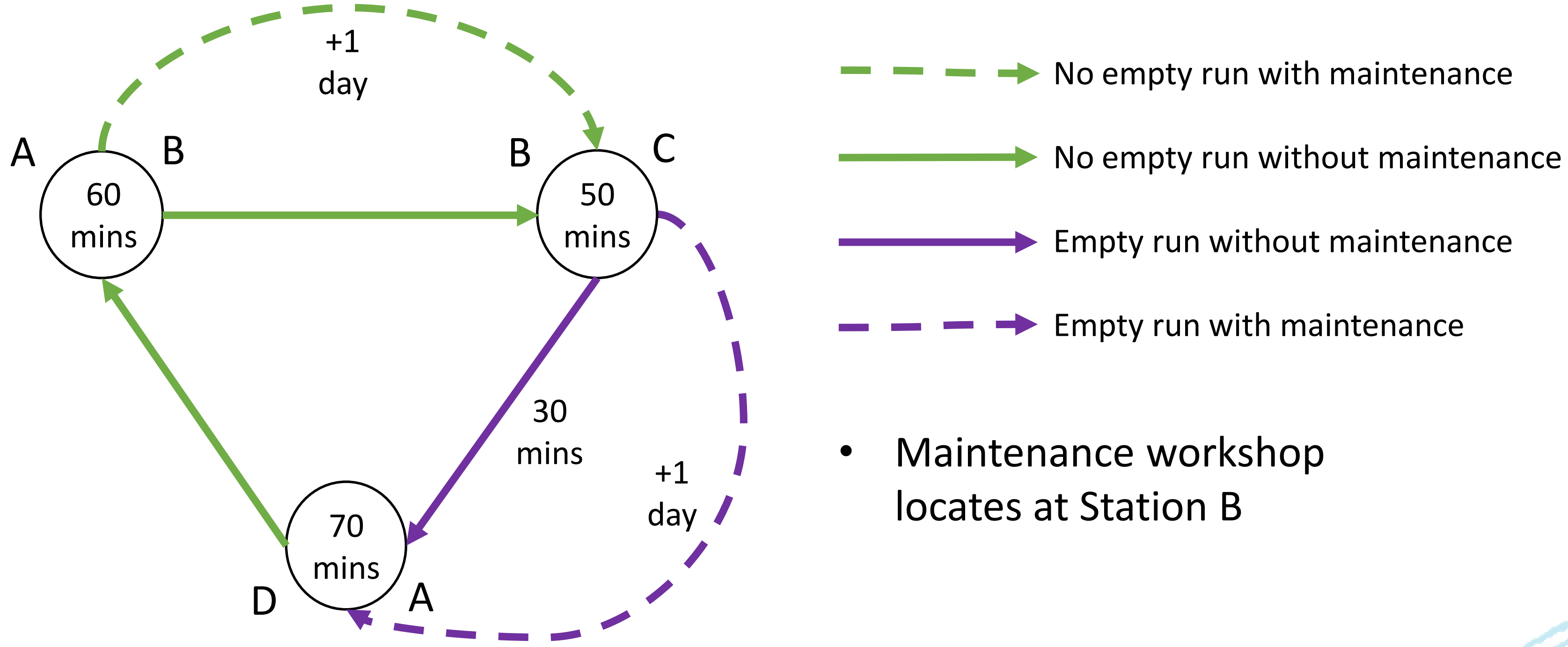
Exploited Software and Framework
Keras-RL: DQNAgent
Scipy, Numpy, Pandas

Hardware Requirements
Google Colaboratory
GPU(s) with CUDA cores and 16GB System RAM

Problem scenarios

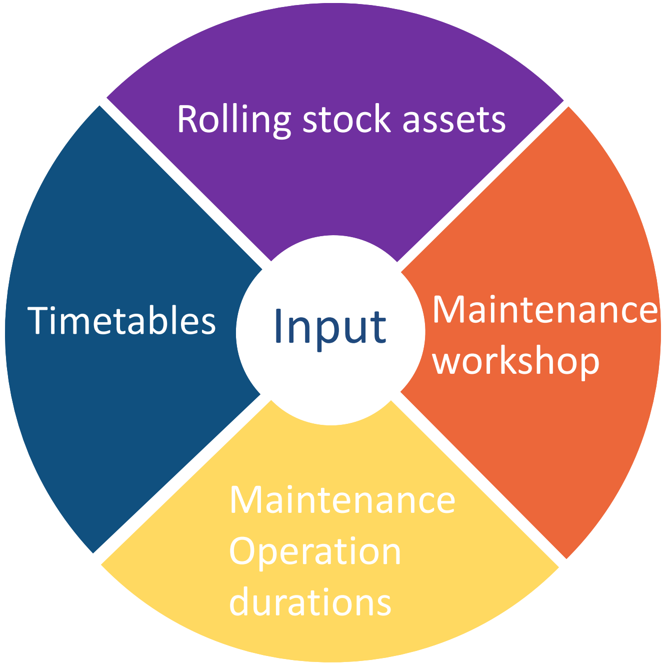


Illustrative Example



- Maintenance workshop locates at Station B

Approach: A Modular Architecture



Objective: Rostering with the minimum costs of RS units (consider maintenance activities)

Rostering

Each train service represents as a node

Arcs denotes the train movements between two consecutive services

Maintenance

Categories of Arcs :
Empty run with/without maintenance
No empty run with/without maintenance

Reinforcement Learning module

Agent
Environment
States
Actions
Reward

Approach: reinforcement learning components

The settings of reinforcement learning module

Fundamental settings

- Set the **Agents** as the RS units, keep the number of agent as less as possible

- Maintain several **Q-matrixes** for each arc category - there are multiple arcs between each station pair

- **Objective** constraints: Minimum overall costs of RS units & number of empty runs

States maintained

- Accumulated kilometres (AK) as the **AK-state** (For each RS units)

- **Maintenance-state**: When the AK over a threshold, i.e., maximum kms for maintenance, set the next **action** with a penalty value: Fail

Other settings

- **Reward**: the necessary costs (measured with time) between two consecutive services, including empty run time + station waiting time + maintenance time (if any)

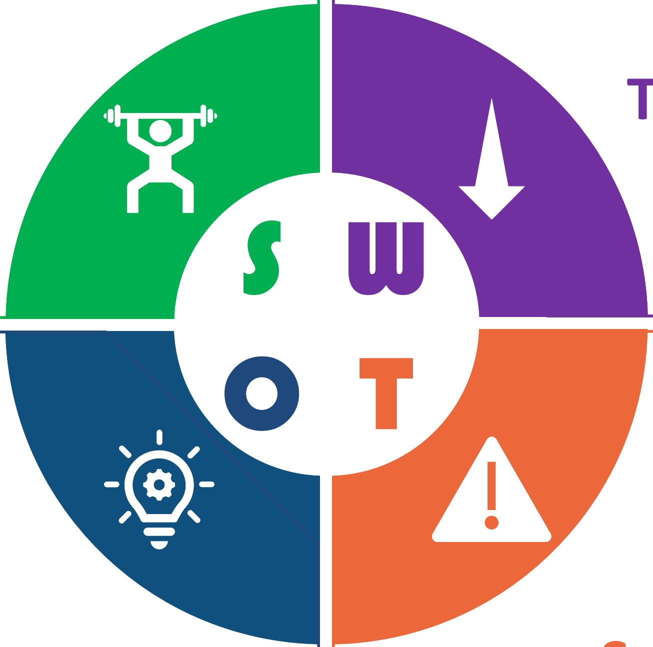
- **Environment**: the different train services to deliver with different origins & destinations, and the next station where to navigate

- **Actions**: at each location, the decisions to make are "which service do I chose to deliver next"

SWOT Analysis of the Investigated Approach

Adaptability
Optimization
Learning from experience
Flexibility and Scalability
Handling uncertainty
Trade-off exploration

Data requirements
Training time and complexity
Lack of interpretability



Enhanced efficiency
Cost reduction
Friendly to dynamic environment

Safety concerns
Sensitivity to unseen senarios
Integration challenges

Recommendations

Multi-objective optimization

Additional objective functions such as energy consumption, Operational resilience

Integration with predictive maintenance

Condition monitoring
Failure prediction

Transfer learning and generalization

Reduce the training time & resource requirements

Thank you for your attention!

RAILS
Roadmaps for AI Integration in raiL Sector

- ✓ *Deliverable D3.1: WP3 Report on case studies and analysis of transferability from other sectors (safety and automation)*
- *Deliverable D3.2: WP3 Report on AI approaches and models*
- ✓ *Deliverable D3.3: WP3 Report on experimentation, analysis, and discussion of results*
- *Deliverable D3.4: WP3 Report on identification of future innovation needs and recommendations for improvements*

Available at: https://rails-project.eu/downloads/deliverables/