

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/334327730>

Flatland Challenge: Multi Agent Reinforcement Learning on Trains.

Preprint · July 2019

DOI: 10.13140/RG.2.2.24477.05601

CITATIONS

0

READS

153

8 authors, including:



Erik Nygren

Plattform für Forschung und Innovation at SBB

5 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)



Giacomo Spigler

Tilburg University

15 PUBLICATIONS 137 CITATIONS

[SEE PROFILE](#)

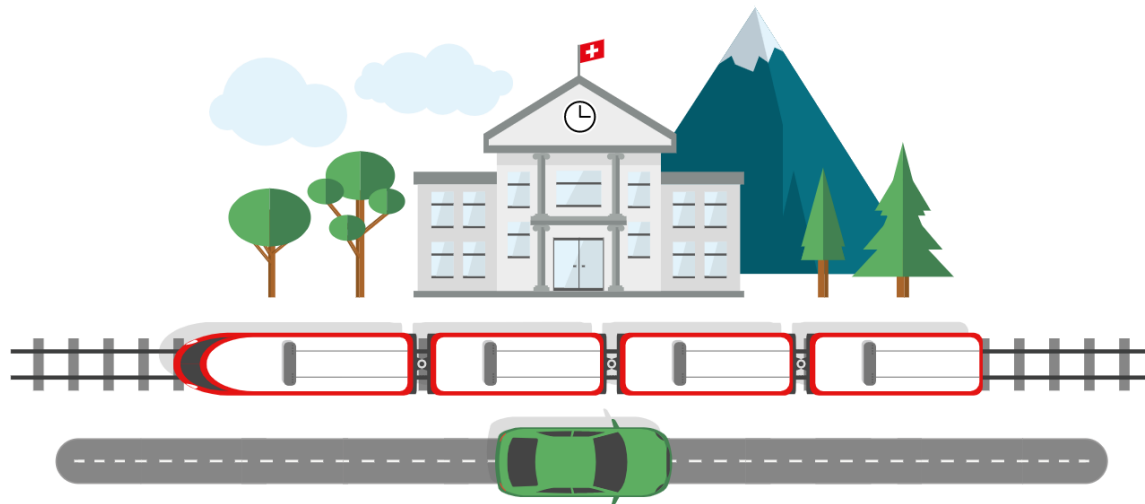
Some of the authors of this publication are also working on these related projects:



Multi-Agent Reinforcement Learning for Railways Dispatching [View project](#)



Temporal Singularity [View project](#)



FLATLAND

Erik Nygren, Adrian Egli, Giacomo Spigler, Mattias Ljungström, Jeremy Watson, Christian Eichenberger, Guillaume Mollard, Sharada Mohanty

Link to the Challenge <https://www.aicrowd.com/challenges/flatland-challenge>
Link to the library : <https://gitlab.aicrowd.com/flatland/flatland>

The key question we want to answer here is: How can trains learn to automatically coordinate among themselves, so that there are minimal delays in large train networks ?

Abstract

The Flatland Challenge is a competition to foster progress in multi-agent reinforcement learning for any [re-scheduling problem \(RSP\)](#). The challenge addresses a real-world problem faced by many transportation and logistics companies around the world (such as the Swiss Federal Railways, SBB. Different tasks related to RSP on a simplified 2D multi-agent railway simulation must be solved. Your contribution may shape the way modern traffic management systems (TMS) are implemented not only in railway but also in other areas of transportation and logistics. This will be the first of a series of challenges related to re-scheduling and complex transportation systems.

Background

The Swiss Federal Railways (SBB) operate the densest mixed railway traffic in the world. SBB maintain and operate the biggest railway infrastructure in Switzerland. Today, there are more than 10,000 trains running each day, being routed over 13,000 switches and controlled by more than 32,000 signals. Each day 1.2 million passengers and almost half of Switzerland's volume of transported goods are transported on this railway network. Due to the growing demand for mobility, SBB needs to increase the transportation capacity of the network by approximately 30% in the future.

The increase in transport capacity can be achieved through different measures, such as [denser train schedules, investments in new infrastructure, and/or investments in new rolling stock](#). However, SBB currently lack suitable technologies and tools to quantitatively assess these different measures.

A promising solution to this dilemma is a complete railway simulation that efficiently evaluates the consequences of infrastructure changes or schedule adaptations for network stability and traffic flow. A complete railway simulation consists of a full dynamical physics simulation as well as an automated traffic management system.

***Flatland:** This image illustrates an early draft of the environment visualization. The core task of this challenge is to manage and maintain railway traffic on complex scenarios in complex networks.*

The research group at SBB has developed a high-performance simulator which simulates the dynamics of train traffic as well as the railway infrastructure. Different approaches for [automated traffic management systems \(TMS\)](#) are currently under investigation. The role of the traffic management system is to select routes for all trains and decide on their priorities at switches in order to optimize traffic flow across the network.

At the core of this challenge lies the general vehicle re-scheduling problem (VRSP) proposed by Li, Mirchandani and Borenstein in 2007:

The vehicle rescheduling problem (VRSP) arises when a previously assigned trip is disrupted. A traffic accident, a medical emergency, or a breakdown of a vehicle are examples of possible disruptions that demand the rescheduling of vehicle trips. The VRSP can be approached as a dynamic version of the classical vehicle scheduling problem (VSP) where assignments are generated dynamically.

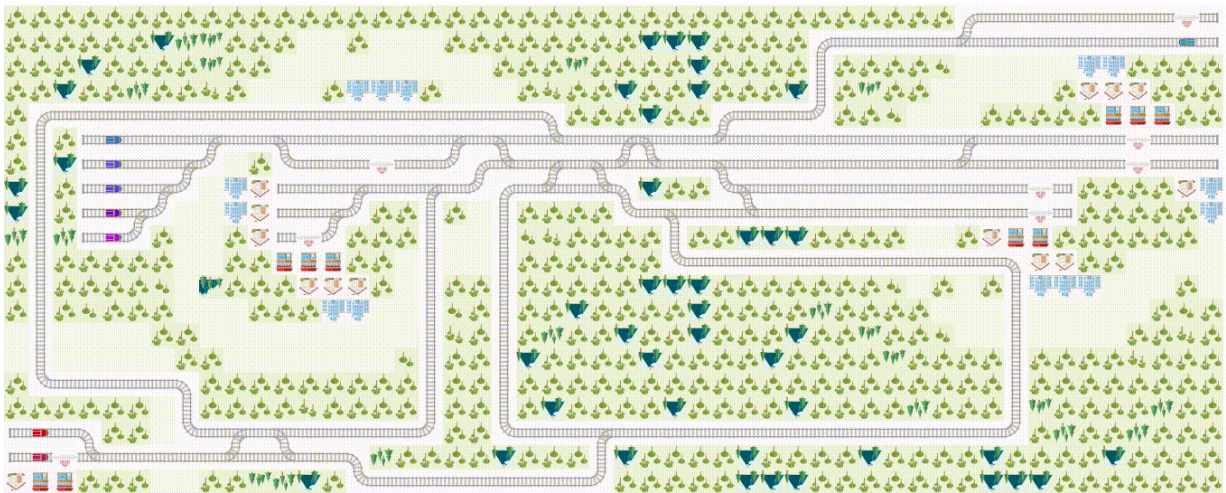
The “Flatland” Competition aims to address the vehicle rescheduling problem by providing a simplistic grid world environment and allowing for diverse solution approaches. The challenge is open to any methodological approach, e.g. from the domain of reinforcement learning or of operations research.

The problems are formulated as a 2D grid environment with restricted transitions between neighboring cells to represent railway networks. On the 2D grid, multiple agents with different objectives must collaborate to maximize global reward. There is a range of tasks with increasing difficulty that need to be solved as explained in the coming sections.

Tasks

The challenge requires your creativity and savviness. In 3 submission rounds with increasing difficulty, you can prove that you have what it takes. We invite you to enter the race with your unique solution and to win great prizes - at the same time solving one of the key challenges in the world of transportation!

Here is a teaser of what we expect you to do:



Your overall goal is to make all agents (trains) arrive at their target destination with a minimal travel time. In other words, we want to minimize the time steps (or wait time) that it takes for each agent in the group to reach its destination.

Let's say in a scenario with n -agents, the travel time is measured by the collected amount of timesteps all the agents have until the n -th agent arrives at its destination.

1. Can you design the best-performing agent?

Design the best-performing agent. At the more basic levels, the agents may achieve their goals using ad-hoc decisions. But as difficulty increases from round to round, the agents have to be able to plan ahead, i.e. with increasing difficulty, planning becomes more relevant!

2. Can you design the best observation?

As a participant, you have the choice. You can either work with the three base observations that we prepared or better, design an improved observation yourself. If you do the latter, then share your observation and you will have chances of winning

the Community Contribution Prize (see Prizes). These are the three base observation that we prepared:

- Global Observation: The whole scene is observed
- Local Grid Observation: A local grid around the agent is observed
- Tree Observation: The agent can observe its navigable path to some predefined depth.

Sounds complicated? Do not despair, the next sections will provide you with more useful information about these rounds!

Rounds

There will be 3 rounds in the challenge. The first one (round 0) is a beta round and serves as an introduction to get familiar with Flatland (as well as bug fixing). Rounds 1 and 2 pose the actual problems to be solved. Submissions are only accepted for Round 1 and Round 2, both rounds will contribute to the final ranking.

Round 0: Learn to navigate (Beta Round): A single agent has to navigate from a freely chosen starting point to a freely chosen target destination on a random infrastructure. It is, in other words, a relatively simple shortest path problem.

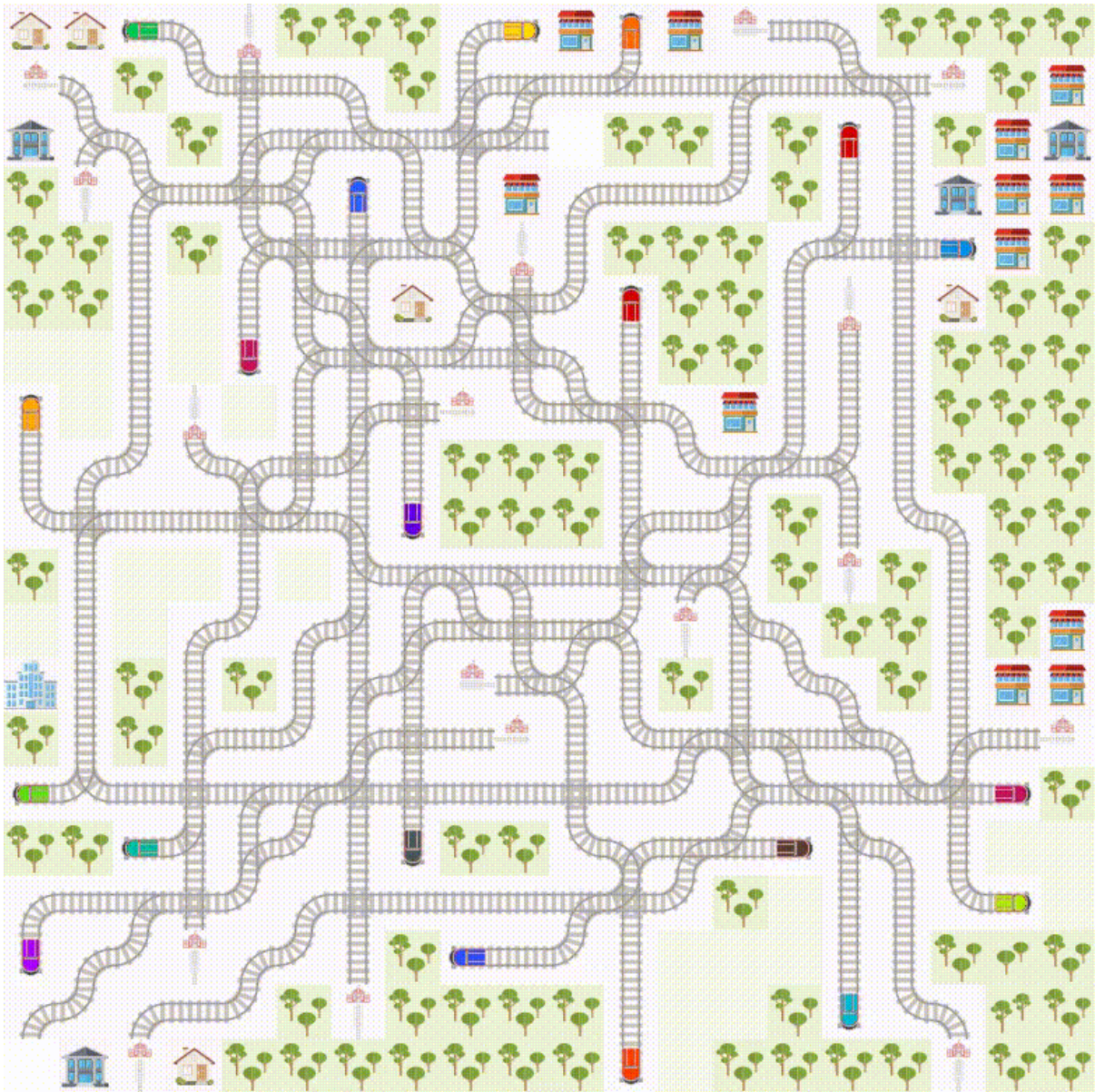
There will be no uploading possibility, no ranking, nor any prizes to be gained in this round- but the collected insights make it all worth it!

Check out this simple [introduction to training](#) to get started with your own training on flatland.

The beta round starts on the 1st of July 2019 and ends on the 14th of July 2019

Round 1: Avoid conflicts: We pick-up the same problem from the previous round and turn it into a multi-agent problem. This means, multiple agents have to find their ways to their respective target destinations. In this scenario you are likely to encounter resource conflicts when two or more agents simultaneously plan to occupy the same section of infrastructure. Thus, the agents have to learn to avoid conflicts and find feasible solutions. By timely submitting your solution and adhering to the participation rules (see here) you are automatically eligible for the Contribution Price & Best Agent Price. Good luck!

Round 1 will open on Monday, 15th of July and close on Sunday, 29th of September 2019, 12 PM. UTC +1



Round 2: Optimize train traffic: In reality, not all trains can go at the same speed. In round 2 we introduce additional complexity to the multi-agent-problem of round 1 by letting the trains have different speeds! This means that a good solution not only avoids/resolves conflicts, but also optimizes by taking into account that slower agents can slow down the faster ones. The price is reserved for the winner who submits the the solution with the minimal cumulated travel time for all agent. By submitting your solution timely and adhering to the participation rules (see [here](#)) you are automatically eligible for the Contribution Price & Best Agent Price. Good luck!

More details, as well as a clip for this round will follow

Round 2 will open mid-August and close on Sunday, 1st of December 2019, 12 PM. UTC +1

Environment

There are a few important basic elements and notions specific to this challenge that you should be aware of before diving into the “Lets get started” section.

Agent

Flatland is a discrete time simulation, that means that all actions performed happen with a constant time step. At each step, the agents can choose an action. The term agent is defined as an entity that can move within the grid and must solve tasks - these agents are, who would have thought, trains. A train does basically two things: wait or go into a particular direction. Depending on the train type (e.g. freight train or passenger train), they have different max. speeds - this fact will first become relevant in Round 2. An agent can move in any arbitrary direction (if the environment permits it) and transition from one cell to the next. If the agent chooses a valid action, the corresponding transition will be executed and the agent’s position and orientation is updated. Each agent has its individual start and target.



Agent at start:

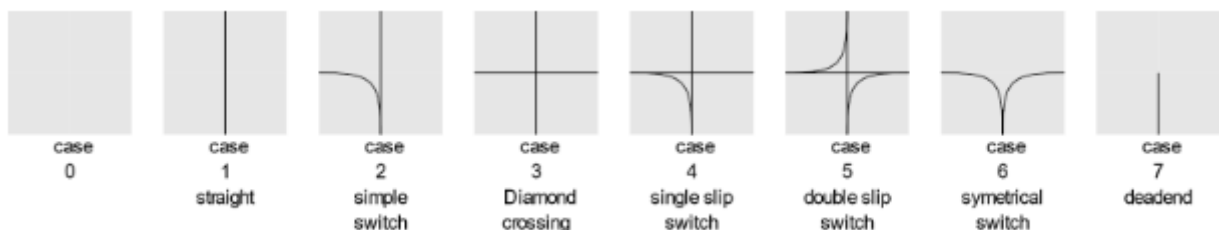


Target Destination:

The cell where the agent is located at must have enough capacity to hold the agent on (thus a “blank” or already reserved cell is impossible). Every agent reserves exact one capacity or resource and since the capacity of a cell is maximal one, it can never hold more than one agent. The different cell-types are introduced in the next section.





Grid World

As you know by now, the Flatland environment consists of cells that are arranged in a simulation grid. These cells have a tile type - for a railway specific problem, 8 basic tile types can be defined. These tile types determine where the agent can be located and how the agent can move through the cell. Here is a quick overview of the tile types:



Of course, there are more possibilities, as these tiles can be rotated in steps of 90° and mirrored along the North-South and East-West axis - but the principal idea remains the same. To get an intuition, let us now discuss four interesting cases in more detail.

Railway Network Fact: Every time an agent approaches a switch, a navigation choice has to be made. In Flatland (like in reality) a maximum of two options is available. There does not exist a switch with three or more options.

Element	Description
	Straight: This tile represents a single passage. While on the tile, the agents can't take any navigation decisions, but only decide to either continue, i.e. passing to the next connected tile or wait.
	Simple Switch: The switch in this tile forces agents who arrive from one direction (in this case from the West) to make a navigation choice (to turn North or go straight). Navigation in either direction is equally costly. Agents coming from any other direction do not have a navigation choice.
	Double Slip Switch: : In this case, we have a crossing with two switches accessible from all four directions. Thus, in every case, the agent has a navigation choice. The agents can only change directions through the switches, the crossing alone doesn't permit direction changes.
	Dead end: : If an agent occupies this cell, only stop or backward motion is possible.

Important to note:

- Due to the dynamics of train traffic, each transition probability is symmetric in this environment. This means that neighboring cells will always have the same transition probability, regardless of direction of movement.
- Each cell is exclusive and can only be occupied by one agent at any given time.

Getting Started

You should now be equipped with the needed background knowledge to get started with the challenge. Check out the [Starter Kit](#) for a description on the technical set up and tips on how to get started.

[Here](#) is the public respository containing all the code you need to participate in this challenge.

Prizes

Your problem solutions mean something to us - hence prizes with a total value of 30k CHF (approx. 30k USD) are reserved for those with the best submissions. You can excel in two categories: The best solution category and the community price category. Within both those categories your submission is individually ranked taking into account your performance in Round 1 and Round 2. Make sure to check the participation rules before you start. Only submissions conforming to our rules have a chance of winning the prizes.

Best Solution Prize: Won by the participants with the best performing submission on our test set. Both of your rankings from the Round 1 and Round 2 are taken into account. Check the leader board on this site regularly for the latest information on your ranking.

The top three submissions in this category will be awarded the following cash prizes (in Swiss Francs):

CHF 7'500.- for first prize

CHF 5'000.- for second prize

CHF 2'500.- for third prize

Community Contributions Prize: Awarded to the person/group who makes the biggest contribution to the community - done through generating new observations and sharing them with the community.

The top submission in this category will be awarded the following cash prize (in Swiss Francs): **CHF 5'000.-**

In addition, we will hand-pick and award up to five (5) travel grants to the Applied Machine Learning Days 2019 in Lausanne, Switzerland. Participants with promising solutions may be invited to present their solutions at SBB in Bern, Switzerland.

Note: It is possible for a participant to win in both categories

Rules

The following rules apply to all participants:

- Participants are allowed at most 5 submissions per day.
- When evaluating individual solutions directly via the REST-API, a limit of one REST-call per minute shall be observed.
- The results achieved by the solver must be reproducible. If there are randomized portions of your approach, be sure to include seeds to make the runs repeatable.
- In case of conflicts, the decision of the Organizers will be final and binding.
- Organizers reserve the right to make changes to the rules and timeline.
- Violation of the rules or other unfair activity may result in disqualification.

More legal details, such as eligibility criteria are [here](#)

Contact

For Challenge-related questions (technical and/or content questions):

- Gitter Channel : <https://gitter.im/AIcrowd-HQ/flatland-rl>
- Technical Issues : Please use the [issue tracker](#) in the public repository
- Discussion Forum : <https://discourse.aicrowd.com/>

We strongly encourage you to use the public channels mentioned above for communications between the participants and the organizers. But in case look for a direct communication channel, feel free to reach out to us at :

- **mohanty [at] aicrowd.com**
- **erik.nygren [at] sbb.ch**

For press inquiries Please contact SBB Media Relations at press@sbb.ch