Learning to Solve Stochastic Multi-Agent Path Finding

Guillaume Dalle, Axel Parmentier

CERMICS, Ecole des Ponts, Marne-la-Vallée, France
{guillaume.dalle,axel.parmentier}@enpc.fr

Mots-clés : *multi-agent path finding, vehicle rescheduling, online stochastic optimization, machine learning for operations research.*

1 Introduction

In large transportation networks, real-time traffic management is essential to minimize disruptions and maximize punctuality. This is especially true for railway systems, where delays can easily propagate from one train to the next due to infrastructure constraints. We propose novel algorithms to tackle this problem, using the Flatland challenge as a testing ground.

2 The Flatland challenge

The AIcrowd Flatland challenge [2] is an international competition in which participants must solve the vehicle rescheduling problem in a simple simulation environment. This environment was designed in collaboration with three national railway companies (SNCF, DB and SBB) to reflect the main features of railway traffic dynamics.

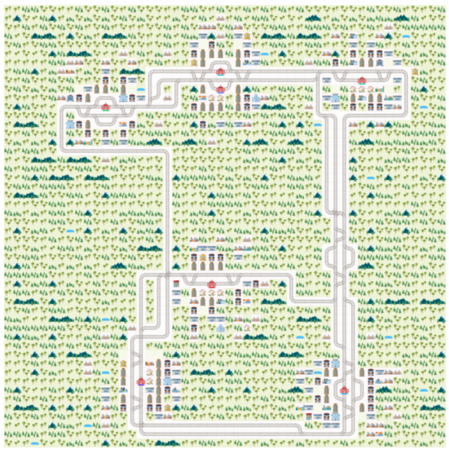

FIG. 1: Example Flatland network

The goal of the challenge is to route multiple trains through a network (such as the one in Figure 1) from their origin to their destination. This requires coordination: indeed, two trains cannot simultaneously occupy the same vertex or cross the same edge in opposite directions (no double tracks). The objective function is given by the sum of all delays, with additional penalties for trip cancellations.

Not only is it necessary to plan the route of every train in advance, these routes must also be adapted in real time as the simulation includes random malfunctions. Such incidents can unexpectedly freeze a train for multiple time steps, forcing the others to change course in order to prevent unnecessary slowdowns or even deadlocks. This calls for an algorithm fast enough to be run after each disturbance.

3 Multi-Agent Path Finding

While the Flatland challenge is designed to encourage contributions in the field of Machine Learning (ML), more traditional submissions using Operations Research (OR) have consistently won the competition so far. Indeed, the best algorithms, such as the 2020 winner [1], take advantage of a large body of optimization research dedicated to Multi-Agent Path Finding (MAPF).

This combinatorial problem consists in finding paths on a graph for a set of agents such that no two paths conflict [5]. Most algorithms rely on the observation that without conflict constraints, MAPF decomposes into a set of independent shortest path problems, which can be solved very efficiently using the A* algorithm. Therefore, the collective research effort focuses on ways to use shortest path algorithms by preventing conflicts beforehand or repairing them afterwards.

4 Our solution approach

Many of the most efficient approaches are heuristics where some decisions are made arbitrarily: for instance, how to define a priority ranking on the trains, or how to select which conflict to resolve first when repairing an infeasible solution. We propose a more principled way to make these choices: by leveraging recent progress on ML for OR [3], we can feed the main features of the environment (graph structure, agent distribution, destinations and deadlines) to a statistical predictor, which will then give us hints on the right way to tune our algorithm.

To build a dataset, we simply use the Flatland simulator to generate many different instances. Quite surprisingly, we do not need to precompute exact solutions for them: the heuristics we build upon would probably not be able to find such solutions anyway. Our goal is to guide these heuristics towards their own best possible solution, which is why we simply repurpose the objective function of the problem as a learning loss.

Since our predictor is trained with optimization as the end goal, we expect that the performance of our ML-enhanced heuristics will exceed the results obtained with manual tuning. In particular, designing a model that captures delay propagation phenomena would lead to considerable improvements.

In conclusion, applying our method to greedy routines such as prioritized planning [4] may make them precise enough to solve complex instances to near-optimality. And crucially, this would not affect their speed since the learning phase happens offline. In a real-time setting where fast computations are essential, such a mix of ML and OR could open new avenues for reactive and sound railway traffic management.

References

[1] Jiaoyang Li et al. "Scalable Rail Planning and Replanning: Winning the 2020 Flatland Challenge". In: *Proceedings of the International Conference on Automated Planning and Scheduling* 31 (May 17, 2021), pp. 477–485. ISSN: 2334-0843. URL: `https://ojs.aaai.org/index.php/ICAPS/article/view/15994` (visited on 06/25/2021).

[2] Sharada Mohanty et al. *Flatland-RL : Multi-Agent Reinforcement Learning on Trains*. Dec. 11, 2020. arXiv: `2012.05893 [cs]`. URL: `http://arxiv.org/abs/2012.05893` (visited on 01/07/2021).

[3] Axel Parmentier. "Learning to Approximate Industrial Problems by Operations Research Classic Problems". In: *Operations Research* (Apr. 16, 2021). ISSN: 0030-364X. DOI: `10/gj3t75`. URL: `http://pubsonline.informs.org/doi/abs/10.1287/opre.2020.2094` (visited on 05/18/2021).

[4] David Silver. "Cooperative Pathfinding". In: *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. AIIDE'05. Marina del Rey, California: AAAI Press, June 1, 2005, pp. 117–122.

[5] Roni Stern et al. "Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks". In: *Twelfth Annual Symposium on Combinatorial Search*. Twelfth Annual Symposium on Combinatorial Search. July 1, 2019. URL: `https://www.aaai.org/ocs/index.php/SOCS/SOCS19/paper/view/18341` (visited on 08/25/2021).