



Modeling the Line Interruption Issue in a Railway Network

Luca Fronda¹, Riccardo Berta^{1(✉)}, Paolo Cesario^{2(✉)}, Alessandro De Gloria¹,
and Francesco Bellotti¹

¹ Department of Electrical, Electronic and Telecommunication Engineering (DITEN),
University of Genoa, Via Opera Pia 11a, 16145 Genova, Italy
s4047957@studenti.unige.it, {alessandro.degloria,
francesco.bellotti}@unige.it

² Si Consulting, Srl - Via Gavotti 5/6, 16128 Genova, Italy
paolo.cesario@siconsulting.biz

Abstract. Rail line interruptions are rare but very costly events, as they require a complete re-definition not only of the timetable of the trains, but also of their path, with major variations at least in the hit area. To the best of our knowledge, the literature is rich of documentation on timetable re-scheduling in case of delays and/or disruption of train lines, but without considering path deviations.

The Flatland initiative has published a 2D railway-world toolkit that allows developers devise and experiment with different solutions to deal with the train re-scheduling issue particularly, but not exclusively, through reinforcement learning (RL) with multi-agent path finding (MAPF). While the approach looks very promising to deal with the complexity of the issue, the platform still has some limitations in terms of the modeling of a realistic railway network scenario. This article proposes the integration of some key features in order to make Flatland a valuable platform for training RL agents for supporting decision making in real-world train re-scheduling.

Keywords: Multi agent path finding · Reinforcement learning · Flatland · Railway interruption

1 Introduction

Rail line interruptions (e.g., due to bad weather or accidents) are rare but very costly events, as they require a re-definition not only of the timetable of the trains, but also of their path, with major variations in the hit area, and, possibly, in the whole national network as well.

In case of large disruptions, railway traffic controllers must apply fast and proper measures to guarantee the train services and minimize delay propagation to the rest of the network. Currently, contingency plans are used to assist traffic controllers in dealing with disrupted traffic [1]. Contingency plans are referred to a specific disruption scenario and to a specific infrastructure, so they are static and limited. In order to address

the specificity of the current disruption case, controllers often have to make adjustments to the original contingency plans. Hence, an efficient and robust decision support system would be a very useful instrument in this context.

The Timetable Rescheduling Problem (TRP) concerns the redefinition of the timetable in case of disruptions or delays. This involves the arrival and departure times of trains, the platforms at which the train should stop and the order of the different trains. To the best of our knowledge, a main limit of the literature on this problem is that it does not consider the possibility of train path deviation, which is anyways a significant opportunity in meshed railway networks, as is the case of several countries, especially in Europe.

Integer Programming (IP) [2], Mixed Integer Programming (MIP) [3] and Alternative Graph (AG) [4] are the three most commonly used models in literature for TRPs in railway networks. Other methods include discrete event simulation [5, 6] and simulation [7]. To find optimal solutions with these models, several techniques have been studied, such as Branch & bound (B&B), heuristic approaches, metaheuristic approaches (including Generative Algorithms (Gas), Simulated Annealing (SA), Tabu Search (TS), Ant Colony Optimization (ACO), Evolutionary Algorithms (EAs)), standard solvers (including CPLEX, GLPK, etc.), etc.

The Flatland initiative, supported by the Swiss and German federal railways, represents a significant novelty in the area. Flatland provides an open source 2D railway toolkit to develop and compare solutions for the problem of efficiently managing dense traffic on complex railway networks. A challenge is held yearly, that addresses the vehicle rescheduling problem by providing a simplistic grid world environment where researchers and practitioners can develop and test solutions based on any methodological approach, with a particular attention to (distributed or centralized) multi-agent reinforcement learning (RL) [8].

We consider this approach to be very promising, because of the ability, demonstrated by RL, of managing complex issues, by learning through (simulated) experience [9]. Moreover, the support to distributed multi-agent path finding opens new perspectives for new transportation services based on dynamic needs. However, the Flatland platform is still limited in terms of modeling a realistic railway network scenario, particularly as the current environment only allows assigning targets that trains have to reach as soon as possible, avoiding any crash. On the other hand, a realistic re-scheduling problem requires at least the definition of the original train schedules and paths, that the system should try to respect as much as possible, also in case of interruptions.

The remainder of this paper is organized as follows. Section 2 presents a critical analysis of the literature concerning the train rescheduling problem. Section 3 presents the proposed extension to the Flatland platform. Section 4 draws the conclusions on the ongoing work.

2 Literature Review

Literature presents various models to address TRP. The IP model relies on binary decision variables, like the priority of two trains, the sequence of trains, and on integer variables, like the arrival and departure times, and the delays. Constraints in the railway network

management are generally expressed by equations and inequalities. In [2], an IP model considering part of the Netherland railway is defined, with a time window of 2 h and 282 trains. The objective is to minimize the arrival time of the passengers.

In the MIP model, the arrival and departure times and delays are considered as a continuous variable. A MIP model of the Dutch railway network, considering a time window of 75 min, is presented in Cavone et al. [3], with the objective to minimize the average and maximum arrival delays of trains.

The AG is a generalization of the disjunctive graph and is used to model complex job shop scheduling problems (JSSPs). In [4], the Dutch national railway network is modeled, with a 1-hour time window and 679 trains. The objective of the model is to minimize the delay propagation to the entire network.

The discrete event models describe the railway system by the state of each train and the discrete events happening in the network at the given time. The dynamic process controls all the trains, determining the optimal velocities, orders, and connections for optimizing different objectives. In [5], the Madrid Regional Railway Network is simulated, exploiting real data provided by the Spanish national railway company, RENFE Operadora. In [6], the authors use a discrete event model to simulate three lines of the Indian railway. This work uses RL in order to minimize the propagation of the delays in the railway net.

A typical simulation model simulates the status of a railway system, but also forecasts the future status, also looking to resolve conflicts. Therefore, the rescheduling approaches can be integrated into the simulation model to support real-time dispatching. In [7], the authors propose a simulation of the Lucerne station area with 30 trains per hour.

Flatland is a two-dimensional simplified grid environment developed by the Swiss Federal Railway Company (SBB) and Deutsche Bahn (DB), that allows faster experimentation providing an easy interface. In the Swiss Railway, on a typical day of operations, more than 10,000 train runs are executed on a network of more than 13,000 switches and 32,000 signals. Almost 1.2 million passengers are transported on the railway network each day. Flatland is a high-performance simulator, which represents and simulates the railway infrastructure and the dynamics of train traffic. Flatland provides specific support to RL. [8] argues that the Flatland environment is a robust and valuable framework to investigate the rescheduling problem in railway networks.

However, Flatland presents some important simplifications with respect to a real railway environment, as stressed by [10]. Particularly relevant to the TRPs are the following:

- Trains cannot move backwards (which could happen in case of a sudden line interruption).
- Trains move with a constant velocity, based on the type of the train, but independent of the railway segment.
- Trains do not follow a timetable, with intermediate stations (to call at before reaching the target station) and arrival and departure times.

Table 1 synthesizes the features of the above cited papers. Mathematical models are the most used ones in literature. They require short computation times, but they grow up very quickly in terms of constraints as the network size and the number of trains

Table 1. Synthesis of the analyzed literature on TRP.

	Model	Model size	Time horizon	Solver	Computation time
[2]	IP	Part of Dutch railway	2 h	CPLex	30 s
[3]	MIP	Dutch railway	75 min	Two level heuristic	60 s
[4]	AG	Dutch railway	1 h	Branch and bound	Maximum of 90 s
[5]	Discrete event simulation	Madrid Regional Railway Network	20 h	Alternative greedy heuristic	~15.5 h
[6]	Discrete event simulation	Three Indian railway lines	72 h	Q-Learning (RL)	~11 min
[7]	Simulation	Lucerne station area	Not specified	Real-time rescheduling and train control approach	Not specified

increase. This is a limit for the scalability (e.g., a nation-wide network in an entire day). Moreover, mathematical models do not consider path deviations, which is necessary in case of disruptions in dense networks.

Simulation models are used in limited geographical areas, with limits in terms of scalability. In this case, attention should be given to [6], that uses a discrete event simulation in order to develop a RL algorithm. In this case, the computational time is limited and the network considered is larger with respect to the other simulation models. This hints in favor of RL as a promising tool to tackle the TRP.

3 Extending the Flatland Platform

From the literature analysis, we argue that the Flatland environment provides a significant opportunity for research, as it allows testing different ML solutions, particularly RL, that could be adequate to address, in a flexible and scalable way, the issue of train rescheduling in case of a rail line interruption, as it could learn from simulated experience.

However, as anticipated, even a realistic albeit simplified use case of train rescheduling requires some features that need to be integrated in Flatland. Particularly, we highlight the following:

- Need for allowing the user to specify generation of trains in the network also after the initial simulation time.
- Need for allowing the user to specify, for each train, such details as: stations to call at, expected arrival and departure schedule in each station, preferential routing. This information is mandatory for passenger's trains – apart from the case of line interruption, which may require the system to learn a different route.

- Need for adding the reverse train action, which is necessary for a train which is in a line segment that has been abruptly interrupted.
- Need for dynamic velocities, that vary depending on the type of train, on the line in which the train is traveling, and on the delay that the train has accumulated.
- Need for considering train run suppression (while keeping alive the train at a station until its end of daily service). Thus, trains should be considered as physical entities, not as independent source-sink services. This implies that the schedule involves the entire path of a day for a train. In this way, a train could avoid a run (i.e., a segment of its daily path), in order to avoid excessive total delay, but do a next one later in the day, starting from the station at which it was blocked.

After analyzing the Flatland source code, we argue that these additions should be addressable by properly extending the schedule generation (by subclassing *Schedule Generator*) and management modules (*rail_env*, *agent_utils*), and by writing proper RL reward/penalty functions, which last point is a standard use of Flatland.

Once implemented, these features should make Flatland environment able to deal with a real-world TRP. A simplified test case will be the one presented in Fig. 1. The use case involves a very simple double track railway network with five stations.

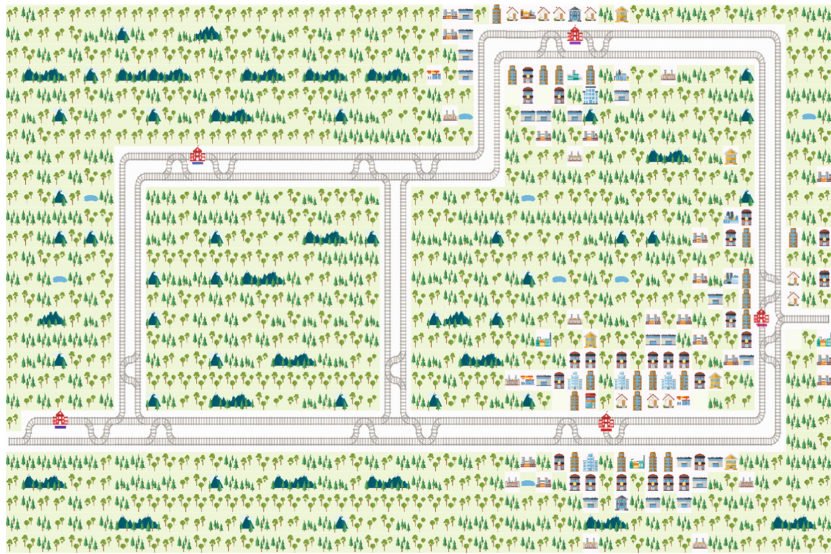


Fig. 1. A simple use case generated with the flatland environment, with five train stations (the red houses on the rail). The red cross represents a two-track disruption in the network.

The train agents should be trained on the whole network and daily activity, with random delays and interruptions, minimizing an overall cost function summing the daily delays and number of missed stations in case of deviation of a train from its ordinary path. It will be interesting to see whether additional information such as location of the

disruption should be provided by the environment to the agents in order to improve the overall performance.

4 Conclusions and Future Work

The problem of dealing with the TRP in case of disruption of a line is serious, but, to the best of our knowledge, has not yet been addressed in literature through a model considering the possibility of deviating trains from their original path in a meshed railway network. We argue that the problem could be tackled through a novel approach, based on RL, so that the train agents autonomously learn how to minimize a cost given by total delay and number of missed stations.

This paper has presented the extensions we are implementing on the open-source Flatland toolkit in order to make it able to better support decision making in real-world TRP cases. The classical research questions to be addressed in an upgraded model will concern the training time needed and accuracy achieved by RL path finding multiple agents, together with the ability of transferring learning in case of different railroad network configurations and, overall, interruption locations or delay states.

From the modeling point of view, a significant further extension will concern modeling train capacities and, overall, inter-station passenger flow demands.

Our focus is currently on a high-level railway network, considering only the connections among the cities, not the connections between a station and its railway depot(s). However, Flatland is a very flexible tool that can be used for this aspect as well. It is anyway of preliminary importance to see whether, once Flatland is extended, it is possible to train multiple RL agents to minimize the overall delay in case of an interruption in a railroad network.

Acknowledgments. This work was also supported by operative program Por FSE Regione Liguria 2014–2020 (Grant Agreement RLOF18ASSRIC).

References

1. Chu, F., Oetting, A.: Modeling capacity consumption considering disruption program characteristics and the transition phase to steady operations during disruptions. *J. Rail Transp. Plan. Manag.* **3**, 54–67 (2013)
2. Dollevoet, T., Huisman, D., Schmidt, M., Schöbel, A.: Delay management with rerouting of passengers. *Transp. Sci.* **46**(1), 74–89 (2012)
3. Cavone, G., Blenkers, L., van den Boom, T., Dotoli, M., Seatzu, C., De Schutter, B.: Railway disruption: a bi-level rescheduling algorithm. In: 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), pp. 54–59 (2019), <https://doi.org/10.1109/CoDIT.2019.8820380>
4. Kecman, P., Corman, F., D’Ariano, A., Goverde, R.M.P.: Rescheduling models for railway traffic management in large-scale networks. *Public Transp.* **5**(1/2), 95–123 (2013)
5. Almodóvar, M., García-Ródenas, R.: On-line reschedule optimization for passenger railways in case of emergencies. *Comput. Oper. Res.* **40**(3), 725–736 (2013)

6. Khadilkar, H.: A scalable reinforcement learning algorithm for scheduling railway lines. *IEEE Trans. Intell. Transp. Syst.* **20**(2), 727–736 (2019). <https://doi.org/10.1109/TITS.2018.2829165>
7. Lüthi, M., Medeossi, G., Nash, A.: Evaluation of an integrated realtime rescheduling and train control system for heavily used areas. In: *Proceedings of the IAROR International Seminar on Railway Operations Modelling and Analysis*, pp. 1–14 (2007)
8. Mohanty, S., et al.: Flatland-RL: multi-agent reinforcement learning on trains. arXiv preprint [arXiv:2012.05893](https://arxiv.org/abs/2012.05893) (2020)
9. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*. MIT Press, Cambridge (2015)
10. Wälter, J.: Existing and novel approaches to the vehicle rescheduling problem (VRSP). Master's thesis, HSR Hochschule für Technik Rapperswil (2020)