The Fifth International Conference on Railway Technology
RAILWAYS 2022
22–25 August 2022
Montpellier, France

Proceedings of the Fifth International Conference on
Railway Technology:
Research, Development and Maintenance
Edited by J. Pombo
Civil-Comp Conferences, Volume 1, Paper 23.5
Civil-Comp Press, Edinburgh, United Kingdom, 2022, doi: 10.4203/ccc.1.23.5

Multi-Agent Deep Reinforcement Learning (MADRL) for Solving Real-Time Railway Rescheduling Problem

B. Kővári[1], I.F. Lövétei[1], Sz. Aradi[1] and T. Bécsi[1]

[1]Budapest University of Technology and Economics, Budapest, Hungary

Abstract

The real-time railway rescheduling problem is a challenging task since several factors have to be considered when a train deviates from the initial timetable. Nowadays, the problem is solved by human operators, which is safe but not optimal. This paper proposes a novel state representation for the introduced control problem that enables the efficient utilization of Multi-Agent Deep Reinforcement Learning. To support our claim, a proof of concept network is implemented, and the performance of the trained agent is evaluated. The results show that our approach enables fast convergence and excellent performance, while the representation has the potential for solving much more complex networks.

Keywords: Rescheduling, Railway Traffic, Multi-Agent Deep Reinforcement Learning, Markov Decision Process.

1 Introduction

The realization of automated rail traffic control is a step forward to efficient capacity utilization to ensure the maximum number of running trains through the network. Besides the costs of a newly built infrastructure, highly developed controlling algorithms provide a cheap alternative solution under disturbed traffic situations. Usually, in most countries, trains run based on a predefined timetable that contains the arrival and departure times for every station (with exact platforms). Every train has a train number that identifies a given train directly. The automated route setting systems use this number - stored in a database with every necessary information - to provide route setting orders to the interlocking systems that ensure safety. These technical systems may tolerate a few minutes late - if the predefined path before a train is still free -, but in a case of a disturbance or a significant late, human operators have to make the right decision. This process means - usually - a rerouting and reordering task under railway company regulations to minimize the total secondary delay on the network. Human operators use their skills and experiences to find a feasible solution. They have to modify the timetable, the crossings, train allocations, the timetable of the connecting lines - if it is necessary. In a modern railway system, the human operators in the Operational Control Centres supervise the traffic and may remotely control the interlocking systems in possession of all the necessary information.

Artificial Intelligence may help to solve these situations automatically. This technology may provide a fast solution for each train to find a conflict-free route through the network. MCTS based algorithms may find the best-first solution in a tree-based representation of the control problem [1]. Simple Q-learning algorithms also find the solution in a simplified infrastructure [2]. The concrete model of railway infrastructure - due to its complexity - makes the solution of the rescheduling task complicated to handle. In a Flatland model, the environment is a simplified grid representation [3]. In this article, a similar two grid model is used to show the performance in a simple station with two platforms, eight sections – signed by S – containing two switches, (see Figure 1.) as proof of our new concept.

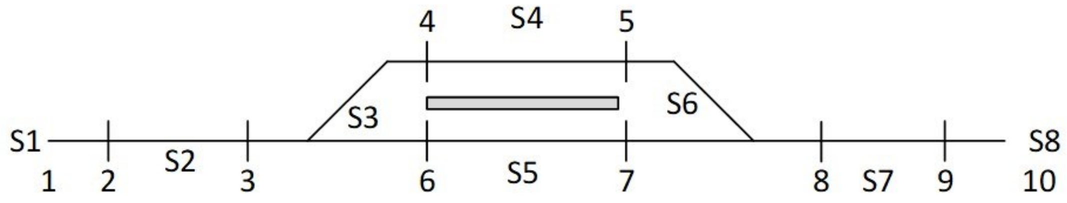

Figure 1: Modelled network.

2 Methods

Deep Reinforcement Learning (DRL) earned the researchers' attention in recent years for its remarkable results in several challenging domains such as Go [4], Chess [5], Videogames [6], Robot control [7], and autonomous driving [8]. The mathematical framework of DRL is constructed as a Markov Decision Process (MDP) <S, A, T, R>, where the agent interacts with an environment in an online manner (Figure 2.) and generates its training samples through these interactions. This aspect is one of the essential features that distinguish Reinforcement Learning from other fields of Machine Learning. The agent goal is to maximize the amount of the gathered reward during the episodes composed by single interactions.

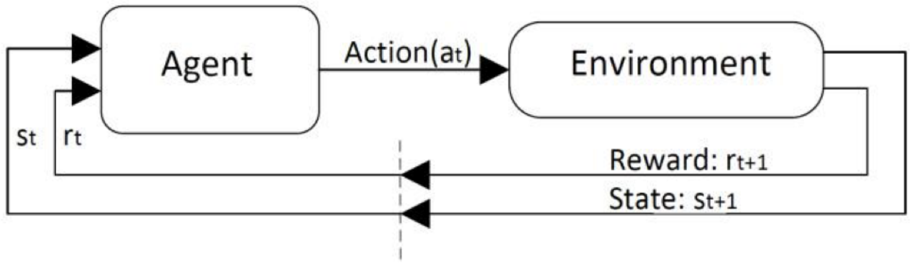

Figure 2: The Reinforcement Learning training loop.

In this paper, we utilize Multi-Agent Deep Reinforcement Learning (MADRL), in which multiple independent learners try to solve the given control problem. In the independent learner approach, the individual agents interpret the other agents as part of the environment. The mathematical framework also changes to Markov Games (MG) <S, N, A, T, R>. We constructed the state representation to be compatible with this approach. The core of the idea is that the network is interpreted as a simple grid word and the control problem as solving a maze, hence finding the shortest path to the destination. However, the problem is more complicated since there could be other trains in the network, which have to be considered. Consequently, the proposed state representation tackles these challenges. Since the network is a 2D grid world, Convolution Neural Networks (CNN) are utilized. Five channels describe the environment state for each agent.

- Channel 1: One-hot encoded version of the network shows the walls and free paths.
- Channel 2: Agent position in the grid-word.
- Channel 3: The agent's destination.
- Channel 4: Agents' positions that travel in the opposite direction.
- Channel 5: Agents' positions that travel in the same direction.

Thanks to this information, every agent is aware of the strategic aspect of the given scenario and can make a decision that does not cause conflicts in the network. There are five possible actions in the environment, making a step into one of the four directions, and the last one is waiting.

The rewarding is fully cooperative. Hence the agents get a positive reward if and only if all the agents arrive at their destination without causing any conflicts.
The utilized method for training the neural network is the Deep Q-Network algorithm that uses the Bellman Equation (1) as an update rule:

$$Q(s_t, a_t; \theta_t) = r_{t+1} + \gamma \max_a Q(s_{t+1}; a_t; \theta_t^-) \tag{1}$$

3 Results

Every episode starts from an initial setup in the training phase. The agents' positions and destinations are chosen randomly to provide the algorithm with diverse training samples. Every episode has a maximum number of steps under which it must be solved. During training, the episode terminates if the agents reach their destination, and when it chooses an action that triggers a conflict or is invalid, which means it does not step on the free path. In such a case, the agent position does not change.

The decision-making for the agents is handled as a batch. Hence the neural network predicts the actions based on the agents' states; then, the environment receives the decisions for every agent in the network, executing them sequentially. The convergence of the algorithm is shown in Figure 3. This figure supports our claim that the representation is well-suited for the control problem since the convergence is fast and stable. The algorithm solves almost every episode at the end of the training process.

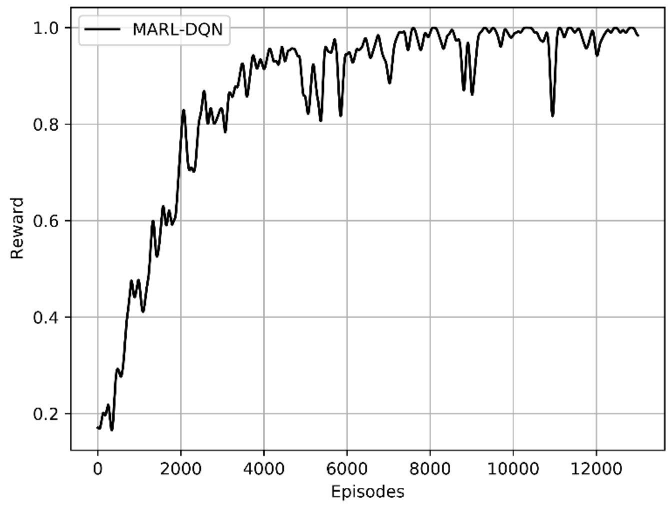

Figure 3: The convergence of the algorithm.

The performance assessment of the trained neural network is conducted in the same manner as the training process but with different random seeds to make sure that the given scenarios are truly unseen for the agent. The agent solved 100% of the provided episode in the evaluation phase. The detailed evaluation is shown in Figure 4. It shows the frequency of the number of steps needed for the algorithm to solve the test episodes. In one step, every agent in the network makes a decision. We also monitored the frequency of choosing the waiting action, which is more informative about the performance since it is only worth choosing if it is strategically required to avoid potentially conflicting scenarios. In the same 1000 episodes, the agents avoid the waiting action in 97.7% of the cases. In contrast, they used them once in 1.3% and twice 1% of the cases.

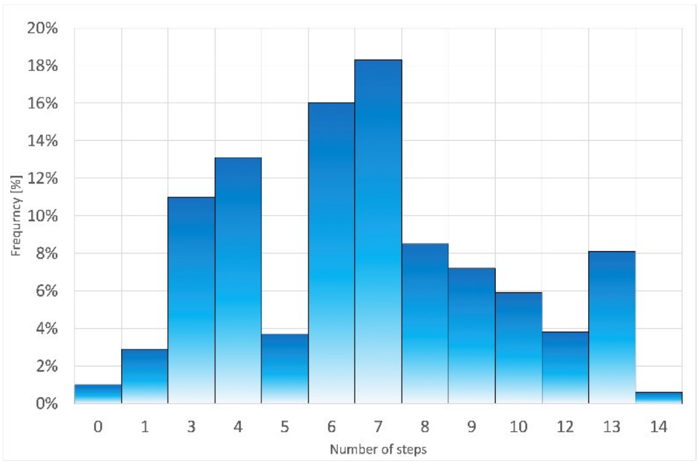

Figure 4: Detailed evaluation.

These results show that the proposed representation enables quick and stable learning of the control problem with solid performance.

4 Conclusions and Contributions

This paper presented a Multi-Agent Deep Reinforcement Learning based solution for the real-time railway rescheduling problem, which was demonstrated on a simple railway station as proof of our new method. A novel state representation was proposed that supports stable and fast learning that clearly benefits the trained agent's performance. By assessing the proposed state representation, it can be seen that it carries a tremendous potential of generalization and scalability thanks to its carefully chosen channels. First of all, thanks to Channel 1, it is possible to change the network architecture during training which will end up a solution that can be flexible from the aspect of the network. Secondly, Channel 3 ensures that the agent does not learn to go to the same destination but go wherever we choose. Finally, Channel 2,4,5 carries all the required information to avoid conflicts in the given network.

In the future, our goal is to evaluate the performance of the representation with much more complex networks and to complement the introduced grid word interpretation of the network with features that bring it closer to a real railway network from a simulation standpoint. We also plan to experiment with different reward strategies that are not fully cooperative but contain competitive aspects of the problem. Finally, we would like to focus our interest on possible communication and negotiation techniques, since in such a control problem, it can benefit the agent to know each other's intentions.

The paper's main contribution is two-folded. First of all, a MARL-based approach is utilized for solving the real-time railway rescheduling problem. Secondly, a novel state representation is proposed to efficiently use MARL techniques for such control tasks where the primary purpose is to avoid deadlocks and unnecessary waiting.

Acknowledgements

Supported by the ÚNKP-21-3 and ÚNKP-21-5 New National Excellence Program of the Ministry for Innovation and Technology from the Source of the National Research, Development and Innovation Fund.

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Acadamy of Sciences.

References

[1] I.F. Lövétei, B. Kővári, T. Bécsi, "MCTS Based Approach for Solvong Real-time Railway Rescheduling Problem", Periodica Polytechnica Transportation Engineering, 49(3), pp. 283-291, 2021.

[2] D. Smerov, R. Marsetic, M. Zura, L. Todorovski, A. Srdic, A, "Reinforcement learning approach for train rescheduling on a single-track railway", Transportation Research Part B, 86(2016) pp. 250-267.

[3] S. Mohanty, E. Nygren, F. Laurent, M. Schneider, C. Scheller, N. Bhattacharya, J. Watson, A. Egli, C. Eichenberger, C. Baumberger, G. Vienken, I. Sturm, G. Sartoretti, G. Spigler, "Flatland-RL: Multi-Agent Reinforcement Learning on Trains", arXiv preprint arXiv: 2012.05893.

[4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, A., ... & D. Hassabis, "Mastering the game of go without human knowledge", Nature, 550(7676), 354-359.

[5] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, ... & D. Silver, „Mastering atari, go, chess and shogi by planning with a learned model", Nature, 588(7839), 604-609.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, ... & D. Hassabis, "Human-level control through deep reinforcement learning", Nature, 518(7540), 529-533.

[7] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, "Continuous control with deep reinforcement learning", arXiv preprint arXiv:1509.02971.

[8] Á. Fehér, Sz. Aradi, T. Bécsi, "Hierarchical Evasive Path Planning Using Reinforcement Learning and Model Predictive Control", IEEE Access, 8, 187470-187482.