Integrated Reinforcement Learning and Optimization for Railway Timetable Rescheduling

Zhang, Hengkai; Liu, Xiaoyu; Sun, Dingshan; Dabiri, Azita; De Schutter, Bart

DOI
10.1016/j.ifacol.2024.07.358

Publication date
2024

Document Version
Final published version

Published in
IFAC-PapersOnline

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-10 (2024) 310–315

Integrated Reinforcement Learning and Optimization for Railway Timetable Rescheduling

Hengkai Zhang * **Xiaoyu Liu** * **Dingshan Sun** * **Azita Dabiri** * **Bart De Schutter** *

* *Delft Center for Systems and Control, Delft University of Technology, 2628CD Delft, The Netherlands (email: hengkai.zhang0@gmail.com, {x.liu-20,d.sun-1,a.dabiri,b.deschutter}@tudelft.nl)*

Abstract: The railway timetable rescheduling problem is regarded as an efficient way to handle disturbances. Typically, it is tackled using a mixed integer linear programming (MILP) formulation. In this paper, an algorithm that combines both reinforcement learning and optimization is proposed to solve the railway timetable rescheduling problem. Specifically, a value-based reinforcement learning algorithm is implemented to determine the independent integer variables of the MILP problem. Then, the values of all the integer variables can be derived from these independent integer variables. With the solution for the integer variables, the MILP problem can be transformed into a linear programming problem, which can be solved efficiently. The simulation results show that the proposed method can reduce passenger delays compared with the baseline, while also reducing the solution time.

Keywords: Railway timetable rescheduling, mixed-integer linear programming, reinforcement learning

1. INTRODUCTION

Railway plays an important role in the modern transportation system. In many countries, railway transport is extensive and includes daily commuting, long-distance travel, and cargo transportation. However, the regular operation of trains is easily affected by disturbances or disruptions. There could be many reasons for railway disturbances and disruptions, such as extreme weather, worker strikes, and system failures. In essence, timetable rescheduling can manage disturbances without impeding the capacity of the railway network. However, disruptions carry a greater impact than disturbances, potentially rendering the original timetable unfeasible and causing significant delays. Railway disturbances and disruptions normally require real-time timetable rescheduling to reduce initial delays and to prevent further propagation of these delays. Therefore, developing effective rescheduling algorithms is critical to the operation of railway systems. In this paper, we investigate the timetable rescheduling approach for disturbances. An efficient rescheduling algorithm should minimize the influence of disturbances.

Various problem formulations have been introduced in the literature to address the railway timetable rescheduling problem (Fang et al., 2015). These problem formulations include mixed logical dynamical systems (Liu et al., 2023a), alternative graphs (D'Ariano et al., 2007), discrete event models (Wang et al., 2015), and event-activity networks (Zhu and Goverde, 2021). Most of these formulations result in a mixed-integer nonlinear programming (MINLP) problem or a mixed-integer linear programming (MILP) problem (D'Ariano et al., 2007; Luan et al., 2018). To solve the resulting mixed-integer programming problems, many studies have considered optimization-based approaches (Liu et al., 2023b; Luan et al., 2018; Yin et al., 2022; Liu et al., 2024). The main advantage of using optimization-based approaches is that they are easy to implement and straightforward. However, the main disadvantage is that most optimization-based methods cannot meet the time requirements for large railway timetable rescheduling problems.

Reinforcement learning (RL) is a machine learning technique where an agent learns how to map states to actions, to maximize a numerical reward signal (Sutton and Barto, 2018). Specifically, at each time step, the agent first measures the state of the environment and then takes an action according to its current policy. The agent then may receive a reward from the environment. The learning target of the agent is to learn an appropriate strategy so that it can maximize the accumulated reward. Because of its negligible online computation time and ability to address uncertainties, RL has been successfully applied in the research of transportation systems, including urban traffic management problems (Sun et al., 2023, 2024), power systems (Fu et al., 2024), and cooperative train control problems (Wang et al., 2023).

There are also a few studies that use RL to solve the railway timetable rescheduling problem. Most literature considers a traffic controller as the agent that directly takes actions to control the railway system, which is regarded as the environment in RL. To find the best possible action in this RL setting, most existing research uses value-based, model-free RL algorithms, including Q-learning (Šemrov

10.1016/j.ifacol.2024.07.358

et al., 2016; Khadilkar, 2019; Zhu et al., 2020) and deep Q-network (DQN) (Ning et al., 2019). The RL-based railway timetable rescheduling approach typically can be divided into the offline training part and the online implementation part, and the online part can be executed efficiently thereby realizing real-time control. However, the optimality of solutions obtained by the learning-based approach cannot be guaranteed. Furthermore, the method used for learning the value function significantly influences the performance of the corresponding approach (Tang et al., 2022). Meanwhile, constraint satisfaction and the convergence of the approach can also be potential drawbacks of RL-based approaches.

This paper focuses on integrating RL and optimization to address railway disturbances, and the main contribution of the paper is twofold: (1) we develop a method combining both RL and optimization for the railway timetable rescheduling problem, where the integer variables are obtained by RL and then the continuous variables are obtained by solving a linear programming problem; (2) we are the first to apply a double deep Q network (DDQN) to train the agent for the timetable rescheduling problem, where a fully connected input layer and a ReLU layer for nonlinear terms are developed for the Q network.

The rest of the paper is organized as follows [1] . In Section 2, the formulation of the railway timetable rescheduling problem is presented. Section 3 introduces the proposed RL-based railway timetable rescheduling method. In Section 4, a case study is presented to illustrate the performance of the developed approach. Section 5 concludes this paper.

2. RAILWAY TIMETABLE RESCHEDULING PROBLEM

This section starts with defining the decision variables, followed by the timetable rescheduling model, and the resulting MILP problem for railway timetable rescheduling.

2.1 Decision Variables

Reordering and retiming are typically applied when disturbances occur. To account for that, continuous variables are defined to consider the operation times of the train, and the integer variables are introduced to consider train orders. Specifically, the continuous variables are defined as follows:

- $a_{i,s}$: arrival time of train i at station s;
- $d_{i,s}$: departure time of train i from station s;
- $\tau_{i,s}$: dwelling time of train i in station s;
- $r_{i,u,s}$: running time of train i between station u and station s.

Then, the integer variables consist of the departure and arrival orders. The departure order is defined as follows:

$$\delta^{\text{dep}}_{i,j,s} = \begin{cases} 1, & \text{if } d_{i,s} - d_{j,s} > 0; \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where $\delta^{\text{dep}}_{i,j,s}$ represents the departure order between train i and j at station s. Similarly, the arrival order is defined as

$$\delta^{\text{arr}}_{i,j,s} = \begin{cases} 1, & \text{if } a_{i,s} - a_{j,s} > 0; \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $\delta^{\text{arr}}_{i,j,s}$ represents the arrival order between train i and j at station s.

2.2 Railway Timetable Rescheduling Model

In what follows several constraints governing the decision variables in Section 2.1 are explained. For train i in the railway system, the train operation should satisfy the dwelling constraint and running time constraints:

$$d_{i,s} = a_{i,s} + \tau_{i,s}, \tag{3}$$

$$a_{i,s} = d_{i,s^-_i} + r_{i,s^-_i,s}, \tag{4}$$

where s^-_i represents the preceding station of station s on the route of train i. The dwelling time $\tau_{i,s}$ and running time $r_{i,u,s}$ should satisfy

$$\tau_{i,s} \geq \tau^{\min}_{i,s}, \tag{5}$$

$$r_{i,u,s} \geq r^{\min}_{i,u,s}, \tag{6}$$

where $\tau^{\min}_{i,s}$ is the minimum dwelling time of train i at station s, and $r^{\min}_{i,u,s}$ is the minimum running time of train i between station u and s.

In principle, the train cannot depart earlier than the originally scheduled departure time:

$$d_{i,s} \geq D_{i,s}, \tag{7}$$

$$a_{i,s} \geq A_{i,s}, \tag{8}$$

where $D_{i,s}$ and $A_{i,s}$ represent the original departure time and arrival time of train i at station s.

To describe the headway constraint of trains departing from a shared station, the parameter $\beta_{i,j,s}$ for the departure of two trains is defined as follows:

$$\beta_{i,j,s} = \begin{cases} 1, & \text{if train } i \text{ and } j \text{ use the same track} \\ & \text{for their departure at station } s; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

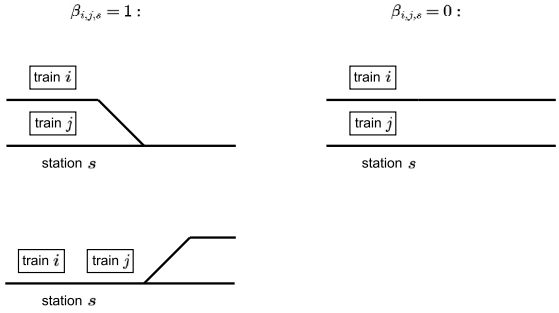

Fig. 1. Illustration of departure situation parameter $\beta_{i,j,s}$

Similarly, as shown in Fig. 2, the parameter $\theta_{i,j,s}$ for the arrival of two trains is defined as follows:

$$\theta_{i,j,s} = \begin{cases} 1, & \text{if train } i \text{ and } j \text{ use the same track} \\ & \text{for their arrival at station } s; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Then, the departure-departure headway between trains i and j at station s should satisfy:

$$d_{i,s} - d_{j,s} \geq h^{\min} - M(2 - \beta_{i,j,s} - \delta^{\text{dep}}_{i,j,s}), \tag{11}$$

[1] This paper is the compact version of the master thesis of Zhang (2023).

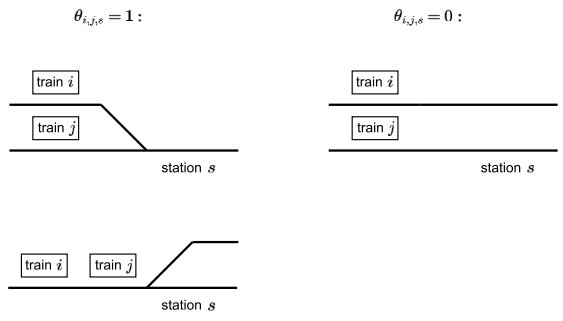

Fig. 2. Illustration of arrival situation parameter $\theta_{i,j,s}$

where $H^{\min}$ denotes the minimum headway, and M represents a large positive constant. Eq. (11) indicates that only when $\beta_{i,j,s}=1$ and $\delta_{i,j,s}=1$, the departure headway constraints should be considered; otherwise (11) holds automatically.

For the consistency of the departure order, we have

$$\delta^{\text{dep}}_{i,j,s}+\delta^{\text{dep}}_{j,i,s}=1. \tag{12}$$

Similar to (11) and (12), the arrival-arrival headway constraints are defined as follows:

$$a_{i,s}-a_{j,s}\geq h^{\min}-M(2-\theta_{i,j,s}-\delta^{\text{arr}}_{i,j,s}), \tag{13}$$
$$\delta^{\text{arr}}_{i,j,s}+\delta^{\text{arr}}_{j,i,s}=1. \tag{14}$$

For the station with more than one platform, reordering can be applied, and we have

$$\sum_{j\in\mathcal{T}(s)\setminus\{i\}}(\delta^{\text{arr}}_{i,j,s}-\delta^{\text{dep}}_{i,j,s})\leq\xi_s-1, \tag{15}$$

where $\mathcal{T}(s)$ is the set of all trains that pass the station s, and ξ_s is a parameter indicating the total number of platforms in station s. Eq. (15) represents that at most ξ_s-1 trains can change their order with train i at station s with ξ_s platforms.

Furthermore, the arrival orders and departure orders between two connected stations should satisfy

$$\beta_{i,j,s^-_i}\theta_{i,j,s}\delta^{\text{arr}}_{i,j,s}=\beta_{i,j,s^-_i}\theta_{i,j,s}\delta^{\text{dep}}_{i,j,s^-_i}. \tag{16}$$

Constraint (16) indicates that if train i and train j use the same track to travel from station s^-_i to station s, their arrival order $\delta^{\text{arr}}_{i,j,s}$ must be as the same as their departure order $\delta^{\text{dep}}_{i,j,s^-_i}$.

2.3 The MILP Problem Formulation

In this paper, the objective is to minimize delays for all passengers at all stations. The objective function of the railway timetable rescheduling problem is formulated as follows:

$$J=\sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{T}}p_{i,s}(a_{i,s}-A_{i,s}), \tag{17}$$

where $\mathcal{S}$ and $\mathcal{T}$ are sets of all stations and all trains respectively, $p_{i,s}$ represents the number of passengers on train i with the destination of station s, $A_{i,s}$ is the original arrival time of train i at station s from the pre-defined timetable. In this paper, the value of $p_{i,s}$ is assumed to be known, which in general can be estimated based on the historical data.

The railway timetable rescheduling problem is then given as

$$\begin{aligned}
\min\quad & J:=\sum_{s\in\mathcal{S}}\sum_{i\in\mathcal{T}}p_{i,s}(a_{i,s}-A_{i,s})\\
\text{s.t.}\quad & d_{i,s}=a_{i,s}+\tau_{i,s},\\
& a_{i,s}=d_{i,s^-_i}+r_{i,s^-_i,s},\\
& d_{i,s}\geq D_{i,s},\\
& a_{i,s}\geq A_{i,s},\\
& r_{i,u,s}\geq r^{\min}_{i,u,s},\\
& \tau_{i,s}\geq\tau^{\min}_{i,s},\\
& d_{i,s}-d_{j,s}\geq h^{\min}-M(2-\beta_{i,j,s}-\delta^{\text{dep}}_{i,j,s}),\\
& \delta^{\text{dep}}_{i,j,s}+\delta^{\text{dep}}_{j,i,s}=1,\\
& a_{i,s}-a_{j,s}\geq h^{\min}-M(2-\theta_{i,j,s}-\delta^{\text{arr}}_{i,j,s}),\\
& \delta^{\text{arr}}_{i,j,s}+\delta^{\text{arr}}_{j,i,s}=1,\\
& \sum_{j\in\mathcal{T}(s)\setminus\{i\}}(\delta^{\text{arr}}_{i,j,s}-\delta^{\text{dep}}_{i,j,s})\leq\xi_s-1,\\
& \beta_{i,j,s^-_i}\theta_{i,j,s}\delta^{\text{arr}}_{i,j,s}=\beta_{i,j,s^-_i}\theta_{i,j,s}\delta^{\text{dep}}_{i,j,s^-_i},\\
& i,j\in\mathcal{T},\ s\in\mathcal{S}.
\end{aligned} \tag{18}$$

The optimization problem in (18) is an MILP problem.

3. RL-BASED RAILWAY TIMETABLE RESCHEDULING

In this section, the underlying setting to be used in RL is introduced. Then, the RL-based algorithm is developed for timetable rescheduling.

3.1 Environment Settings

In this paper, the environment is a combination of the railway network and the MILP problem. Specifically, at each time step, the RL agent makes decisions based on the real-time state of the railway system. After receiving the integer solutions from the RL agents, the MILP problem is transformed into a linear programming problem. After the linear programming problem is solved, the complete solution, i.e. the new timetable, will be applied to the railway network until the next time step.

Environment Update In this paper, the railway system is formulated as an event-triggered and time-based system. When a train i arrives at station s, the railway timetable rescheduling problem is triggered if the following condition holds:

$$(r_{i,s}+a_{i,s}\geq R_{i,s}+A_{i,s})\wedge(\xi_s>1). \tag{19}$$

The first part of (19) describes that train i is delayed when it arrives at station s, and the second part indicates that reordering is possible when station s has more than one platform.

State To build the connection between consecutive steps, a set $\mathcal{T}^0$ is introduced to include the most recently departed trains on each track from the initial station s_0. The state representation of the RL environment is given as follows:

$$\Lambda=(\boldsymbol{p},\boldsymbol{a}_{s_0},\boldsymbol{\delta}_{s_0},\boldsymbol{\mu},\mathcal{T}^0), \tag{20}$$

where $\boldsymbol{p}$ is the vector representing all variables related to the number of passengers, $\boldsymbol{a}_{s_0}$ is the vector denoting

the arrival times of all trains at the initial station s_0, $\boldsymbol{\delta}_{s_0}$ denotes the vector of train orders at the initial station s_0, and $\boldsymbol{\mu}$ represents the existing delays.

Action The action of the RL agent is the independent integer variable of the MILP problem. There are two main reasons for not selecting all integer variables as the action. First, the action space may become too large to allow for efficient learning. Second, the action space may not be fully feasible, which will cause further problems with reward design and environment updates.

According to the layout of the railway network, we define the following preprocessing principles to prune some binary variables and find independent integer variables:

- The order between two trains on the same track cannot be changed.
- At the intersection point where multiple tracks merge into one track, the order between the trains that have already passed this intersection and trains that have not yet passed cannot be changed.
- Conjugate orders are always dependent on each other according to (12) and (14).

The main aim of the preprocessing is to reduce the solution space by removing infeasible solutions, thereby ensuring constraint satisfaction in the RL approach and improving solution efficiency.

Reward Function For the reward function, the negative value of the objective function given in (17) is used as follows:

$$r = -K \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{T}} p_{i,s}(a_{i,s} - A_{i,s}), \tag{21}$$

where K is a scaling constant.

3.2 Reinforcement Learning Algorithm

As a variant of the DQN method (Mnih et al., 2015), the double deep Q network (DDQN) uses two different networks to select and evaluate the action in order to address the problem of overestimation. The DDQN approach (van Hasselt et al., 2016) is implemented to train the agent, where the target network is used to evaluate and the online network is applied to select the action. In this context, DDQN is generally considered superior to DQN due to its ability to mitigate overestimation bias by decoupling action selection and evaluation using two separate neural networks (van Hasselt et al., 2016).

The Q network consists of an input layer, two hidden layers, and one output layer, and the structure of the neural network structure is presented in Fig. 3.

The target employed by the DQN approach is defined as follows:

$$y_t = \Psi_{t+1} + \gamma \max_{\omega} \hat{Q}(\Lambda_{t+1}, \omega; \boldsymbol{\eta_t^-}), \tag{22}$$

where $\boldsymbol{\eta_t^-}$ represents the parameters of the target Q network for DQN, Λ_{t+1} denotes the state at step $t+1$, and ω represents the action defining all independent variables. For a clear comparison, the target equation can be rewritten as

$$y_t = \Psi_{t+1} + \gamma \hat{Q}(\Lambda_{t+1}, \arg\max_{\omega} \hat{Q}(\Lambda_{t+1}, \omega; \boldsymbol{\eta_t^-}); \boldsymbol{\eta_t^-}). \tag{23}$$

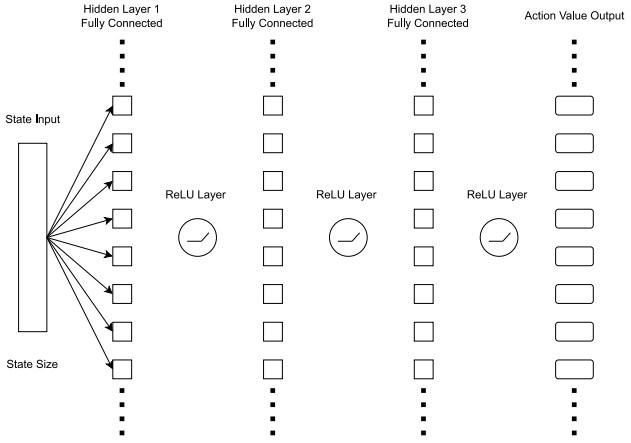

Fig. 3. Structure of the Q-network

The DDQN approach uses two different convolutional neural networks to select and evaluate the action. In this context, the existing online network of the DQN, represented by Q in this paper, is chosen to select the action, and the target network, represented by $\hat{Q}$ in this paper, is chosen to evaluate. Therefore, the target equation could be written as

$$y_t = \Psi_{t+1} + \gamma \hat{Q}(\Lambda_{t+1}, \arg\max_{\omega} Q(\Lambda_{t+1}, \omega, \boldsymbol{\eta_t}); \boldsymbol{\eta_t^-}), \tag{24}$$

where two networks for selection and evaluation in the DDQN approach are not fully decoupled, since the target network $\hat{Q}$ remains a periodic copy from the online network Q.

The procedure of the RL-based algorithm is illustrated in Fig. 4.

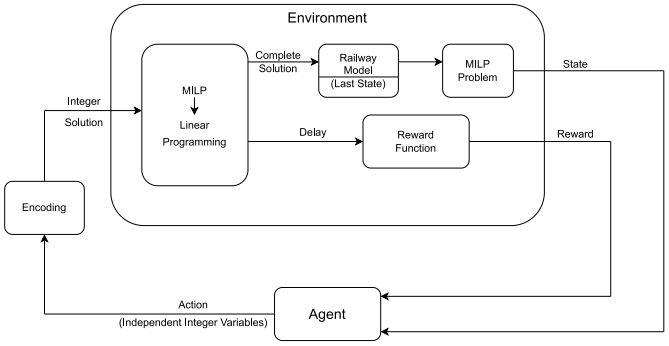

Fig. 4. Procedure for solving the timetable rescheduling problem

4. CASE STUDY

In this section, we conduct a small-scale case study to illustrate the effectiveness of the developed framework. We assume that passenger demands are known so that we can demonstrate the performance of the developed approach under deterministic conditions.

Part of the Dutch railway network from Utrecht (Ut) to 's-Hertogenbosch (Ht) is used for the simulation. The layout of the railway network is shown in Fig. 5. The simulation is conducted through MATLAB R2022b on a MacBook Pro 14.2 2017 with 3.1 GHz Intel i5 CPU and 8 GB RAM.

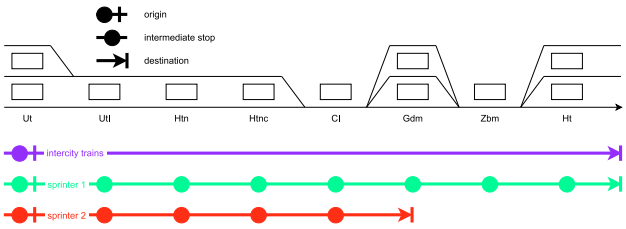

Fig. 5. Layout of the railway network

Optimization problems in the case are solved by using Gurobi Optimizer v10.0.1rc0 (mac64[x86]).

For the timetable, five trains are considered every 30 minutes. Among these trains, three are Intercity trains, and two are Sprinter trains. From Fig. 5, it can be seen that there are two separate tracks from Station Ut to Station Htnc, and thus we do not need to consider the train order from Station Ut to Station Htnc. In this context, the initial station s_0 of the RL agent is set to be Station Htnc.

The line parameters are defined as follows. For all departure parameters $\beta_{i,j,s} = 1$ and for all arrival parameters $\theta_{i,j,s} = 1$. The minimum safety headway is set to $h^{\min} = 90s$. Only Station Htnc and Station Gdm have two platforms, for all other stations $\xi_s = 1$. In this paper, a margin of 7% is accepted for the running time and dwelling time. These parameters are calculated as follows:

$$r^{\min}_{i,u,s} = 0.93R_{i,u,s} \qquad \tau^{\min}_{i,s} = 0.93\Gamma_{i,s},$$

where $R_{i,u,s}$ represents the original running time of train i between station u and s from the timetable, and $\Gamma_{i,s}$ is the original dwelling time of train i at station s.

For every step, five trains are considered. From the layout of the railway network, it is clear that reordering is only possible at Station Htnc and Station Gdm. For Station Htnc, since there are two tracks, 3 independent binary variables could determine the orders of these four trains. Similarly, for Station Gdm, there are five trains and still two platforms, 4 independent binary variables are considered. In total, there are seven independent binary variables for the railway timetable rescheduling problem at one single step. Therefore, the Q network consists of the following layers: the first layer is a fully connected input layer with an output dimension of 256, followed by a ReLU layer for non-linearity. After that, two fully connected layers of 256×256 with ReLU activation after each of them are added. In the end, the input dimension of the output layer is also 256. For the reward function, the constant K is set to be 10^{-4}. Specifically, the rest of parameters for training the RL agent are given in Table 1.

Table 1. Some parameters of training RL agent

Parameter	Notation	Value
Size of Layer 1	-	State Size × 256
Size of Layer 2, 3	-	256 × 256
Size of Output Layer	-	256 × Action Size
Experience Buffer Length	N	10000
Batch Size	-	128
Learning Rate	ζ	0.001
Initial Exploration Rate	ϵ_0	0.8
Epsilon Decay Rate	-	0.001
Discount Factor	τ	0.99
Regularization Factor	-	0.0001
Reward Scaling Constant	K	10^{-4}

In this case study, the influence of multiple extra delays in Station Htns is studied. Every new train may get an extra delay in the new step. The delay item $\boldsymbol{\mu}$ is a five-dimensional vector represented by $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3, \mu_4, \mu_5)$, where μ_i denotes the delay of train i. The probability of adding extra delay for each train is 50%. The exact values of these delays are given in Table 2, where $U(0,1)$ represents a uniform distribution between 0 and 1.

Table 2. Delays of trains for case study

Delay	Initial Step	Other Steps
μ_1	0	$d_{1,\text{Htnc}} - D_{1,\text{Htnc}}$
μ_2	$8 + 4 \cdot U(0,1)$	$1 + 2 \cdot U(0,1)$
μ_3	$4 + 4 \cdot U(0,1)$	$2 + U(0,1)$
μ_4	$5 + 3 \cdot U(0,1)$	$2 + 2 \cdot U(0,1)$
μ_5	$5 + 4 \cdot U(0,1)$	$1 + U(0,1)$

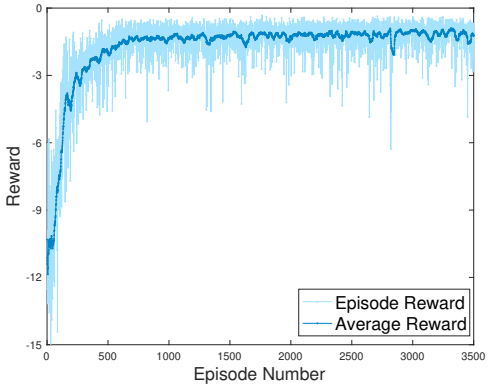

Fig. 6. Episode reward during training process

The average reward during the training process is given in Fig. 6. It can be seen that the agent learned very fast in the first 500 epochs. After about 800 episodes, the learning curve has become relatively stable. The average reward after 3000 epochs is -1.1636. The test results are given in Table 3.

Table 3. Simulation results for railway timetable scheduling

Method	Delay	Improvement	CPU Time
Baseline (FIFO)	38751 min	-	41.3 s
Optimization	7139 min	81.95%	12.5 s
RL-based Method	9582 min	75.08%	8.2 s

Compared with the baseline, both approaches make a significant improvement. Specifically, compared with the baseline, the performance of the RL-based method increases by 75.08%, while the improvement of the optimization-based approach is about 81.95%. Although the RL-based method underperforms the optimization-based method, the gap between these two approaches is small. For one epoch, the RL-based method takes about 8.2 s on average for execution, while the local optimization-based method takes about 12.5 s. Therefore, the RL-based approach reduces execution time by around 34.8%, and hence can be considered as a viable approach to mitigate passenger delays while also reducing solution time.

5. CONCLUSIONS

In this paper, an integrated algorithm that combines RL and linear programming has been developed for the railway timetable rescheduling problem. A double DQN approach with a fully connected input layer and a ReLU layer is applied to train the agent in finding the integer variables. Given these variables, optimization is then used to solve the resulting LP in order to find the remaining continuous time variables of the problem. Simulation results indicate that the developed approach can help to reduce passenger delays while the solution time is reduced.

Topics for future research include extending the integrated approach to large-scale railway networks, developing a more elaborated approach that considers more detailed passenger demands, and performing an in-depth assessment for more complex case studies.

ACKNOWLEDGEMENTS

This work is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101018826 - CLariNet). The work of Xiaoyu Liu was supported by the China Scholarship Council.

REFERENCES

D'Ariano, A., Pacciarelli, D., and Pranzo, M. (2007). A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2), 643–657.

Fang, W., Yang, S., and Yao, X. (2015). A survey on problem models and solution approaches to rescheduling in railway networks. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), 2997–3016.

Fu, J., Sun, D., Peyghami, S., and Blaabjerg, F. (2024). A novel reinforcement-learning-based compensation strategy for dmpc-based day-ahead energy management of shipboard power systems. *IEEE Transactions on Smart Grid*. doi:10.1109/TSG.2024.3382213.

Khadilkar, H. (2019). A scalable reinforcement learning algorithm for scheduling railway lines. *IEEE Transactions on Intelligent Transportation Systems*, 20(2), 727–736.

Liu, X., Dabiri, A., Wang, Y., and De Schutter, B. (2023a). Modeling and efficient passenger-oriented control for urban rail transit networks. *IEEE Transactions on Intelligent Transportation Systems*, 24(3), 3325–3338.

Liu, X., Dabiri, A., Wang, Y., and De Schutter, B. (2024). Real-time train scheduling with uncertain passenger flows: A scenario-based distributed model predictive control approach. *IEEE Transactions on Intelligent Transportation Systems*, 25(5), 4219–4232.

Liu, X., Dabiri, A., Xun, J., and De Schutter, B. (2023b). Bi-level model predictive control for metro networks: Integration of timetables, passenger flows, and train speed profiles. *Transportation Research Part E: Logistics and Transportation Review*, 180, 103339.

Luan, X., Wang, Y., De Schutter, B., Meng, L., Lodewijks, G., and Corman, F. (2018). Integration of real-time traffic management and train control for rail networks-part 1: Optimization problems and solution approaches. *Transportation Research Part B: Methodological*, 115, 41–71.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

Ning, L., Li, Y., Zhou, M., Song, H., and Dong, H. (2019). A deep reinforcement learning approach to high-speed train timetable rescheduling under disturbances. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 3469–3474.

Šemrov, D., Marsetič, R., Žura, M., Todorovski, L., and Srdic, A. (2016). Reinforcement learning approach for train rescheduling on a single-track railway. *Transportation Research Part B: Methodological*, 86, 250–267.

Sun, D., Jamshidnejad, A., and De Schutter, B. (2023). Adaptive parameterized control for coordinated traffic management using reinforcement learning. *IFAC-PapersOnLine*, 56(2), 5463–5468.

Sun, D., Jamshidnejad, A., and De Schutter, B. (2024). A novel framework combining MPC and deep reinforcement learning with application to freeway traffic control. *IEEE Transactions on Intelligent Transportation Systems*. doi:10.1109/TITS.2023.3342651.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement Learning: An Introduction*. MIT press.

Tang, R., De Donato, L., Besinović, N., Flammini, F., Goverde, R.M., Lin, Z., Liu, R., Tang, T., Vittorini, V., and Wang, Z. (2022). A literature review of artificial intelligence applications in railway systems. *Transportation Research Part C: Emerging Technologies*, 140, 103679.

van Hasselt, H., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2094–2100.

Wang, X., D'Ariano, A., Su, S., and Tang, T. (2023). Cooperative train control during the power supply shortage in metro system: A multi-agent reinforcement learning approach. *Transportation Research Part B: Methodological*, 170, 244–278.

Wang, Y., Tang, T., Ning, B., van den Boom, T.J.J., and De Schutter, B. (2015). Passenger-demands-oriented train scheduling for an urban rail transit network. *Transportation Research Part C: Emerging Technologies*, 60, 1–23.

Yin, J., Yang, L., D'Ariano, A., Tang, T., and Gao, Z. (2022). Integrated backup rolling stock allocation and timetable rescheduling with uncertain time-variant passenger demand under disruptive events. *INFORMS Journal on Computing*, 34(6), 3234–3258.

Zhang, H. (2023). Reinforcement learning based real-time railway timetable scheduling. URL `http://resolver.tudelft.nl/uuid:dd52dd98-ad0e-4c11-9659-3a3dfb5fd6cd`.

Zhu, Y. and Goverde, R.M. (2021). Dynamic railway timetable rescheduling for multiple connected disruptions. *Transportation Research Part C: Emerging Technologies*, 125, 103080.

Zhu, Y., Wang, H., and Goverde, R.M. (2020). Reinforcement learning in railway timetable rescheduling. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–6.