A Two-Stage Learning-and-Optimization Framework for Real-Time Train Platforming

Md Tabish Haque, Sebastian Sehmisch, Christopher Szymula, and Nikola Bešinović

Chair of Railway Operations,
Technische Universität Dresden, 01069, Dresden, Germany
{md_tabish.haque}@tu-dresden.de

Abstract. The real-time train platforming problem determines feasible routes and platform allocations for trains during disruptions while limiting operational impacts. The problem is especially challenging in complex station areas, where dense layouts and high traffic volumes generate tightly coupled conflicts. Most existing work targets small-to-moderate disruptions, whereas large-scale station-area disruptions remain less studied. This paper proposes a two-stage learning and optimization framework that combines supervised learning with MILP-based decision-making. The framework follows a Predict Repair Optimize (PRO) paradigm: it first predicts stable routing decisions to reduce the problem, then repairs infeasibilities induced by incorrect fixations, and finally optimizes the resulting MILP. The approach is evaluated on real-world instances, demonstrating that PRO achieves significant run time savings while maintaining solution quality and enhancing incumbents in time-constrained hard instances. The results highlight the capability of ML-integrated approaches to enhance real-time decision-making support in railway operations.

Keywords: machine learning, optimization, railway, disruptions, real-time

1 Introduction

Real-time decision-making in railway operations, commonly known as online problems, entails the dynamic control of train movements in reaction to unforeseen incidents. Such incidents are generally classified as disturbances or disruptions [2]. Disturbance indicates a minor incident, such as track conflicts or primary train delays, that can be swiftly rectified with minimal deviation from the scheduled timetable; however, disruption entails significant incidents, including infrastructure obstructions, accidents, rolling stock malfunctions, or extended signal failures, necessitating comprehensive recovery efforts to reinstate normal operations. Effectively managing the latter requires real-time rescheduling and rerouting strategies in a dynamic environment. Station areas, particularly those characterized by high traffic density, intricate track layouts, and constrained capacity, present significant challenges for real-time rescheduling and heighten the risk of delay propagation throughout the network due to localized bottlenecks. Addressing these challenges necessitates comprehensive infrastructure modeling, analysis of rolling stock characteristics (running times), and assessment of signaling systems, all of which must be evaluated and optimized in real time.

Real-time rail traffic management issues (rRTMP) in station areas are usually referred to as the *real-time train platforming problem* (rTPP) or the *real-time station dispatching problem* (rSDP), which are specialized subsets of rRTMP concentrating on localized rescheduling and rerouting, particularly concerning platform allocation. Although the extensive rRTMP literature generally employs a network-wide perspective, representing stations as simplified nodes or capacity-constrained elements [10, 9, 17], there is a dearth of studies that focus on the operational intricacies particular to station areas. In the majority of rTPP studies, disturbances or disruptions are captured as delays in train arrivals—usually quantified at the home signal—without explicitly accounting for internal station disruptions [3, 10, 5], with the exception of [21], which addresses scheduled maintenance. Consequently, these studies implicitly presume that disruptions arise external to the station area or are due to a scheduled event, neglecting to consider temporary capacity reductions or operational limitations within the station itself. This study is motivated by the rTPP in the context of disruptions within the station area that make the original schedule unfeasible and result in a temporary decrease in operational capacity.

Research in rRTMP focuses on modeling and algorithmic techniques to improve computational efficiency. Solution methodologies can be broadly classified into three categories: decomposition methods, heuristics, and problem space reduction techniques [1]. Decomposition methods leverage problem structure to partition it into smaller, manageable subproblems [9, 10], with certain studies employing a two-stage framework in which offline route selection establishes a constrained set of viable train routes to facilitate real-time optimization [17, 18]. Heuristic and metaheuristic approaches have been utilized for local rerouting and rescheduling to produce rapid, near-optimal solutions [16, 20]. Moreover, techniques for problem space reduction, including the restriction of rescheduling and rerouting decisions [4, 12] or the incorporation of valid inequalities [14], were employed. We implement a two-stage learning-and-optimization framework for the rTPP that improves computational efficiency through systematic problem reduction while preserving solution quality. Operationally, the framework follows a Predict–Repair–Optimize (PRO) pipeline: a supervised learning model first predicts stable nominal routing decisions and fixes the corresponding binaries to reduce the MILP; a targeted repair step then releases critical fixations to restore feasibility; finally, the resulting model is optimized to produce the operating plan. The major contributions of the paper are as follows:

- In contrast to most rTPP studies that model disruptions only as arrival delays at the home signal, we explicitly represent *internal* station-area disruptions, including capacity reductions caused by these disruptions.
- We develop a Predict-Repair-Optimize (PRO) framework that couples ML-based variable fixation with a tailored feasibility-repair heuristic, enabling

reliable warm starts and accelerating MILP solution times without loss in solution quality.
- We evaluate the framework on a real-world station-area case study and demonstrate robust performance gains, highlighting the trade-off between computational speed and solution quality across diverse disruption scenarios.

2 Problem Description

This study examines the rTPP, which entails the rescheduling and rerouting of trains within a station area during disruptions. The station area comprises several essential components, including track segments, switches, crossovers, platform tracks, and signaling systems, as depicted in Figure 1. Home signals delineate the outer limit of the station area, governing train entry and exit while establishing the transitions to and from mainline tracks. The infrastructure network is represented as a graph $G^I = (N, A)$, where nodes N denote track segments and directed arcs A signify direct connections between them. Each train is directed along a series of nodes and arcs between specified entry and exit points, conceptually categorized into two components: the inbound and outbound routes, as illustrated in Figures 1. This is termed the mesoscopic level of detail, as route-specific speed limits are not taken into account.

Disruptions are modeled as the inaccessibility of particular nodes or arcs within the station area. We specifically tackle internal disruptions, including temporary track obstructions or platform unavailability (nodes) and signal malfunctions (arcs), which diminish effective capacity and significantly impact routing and scheduling decisions. The objective is to develop a conflict-free, viable operational plan for the affected trains during disruptions, taking into account infrastructure limitations and operational constraints. We proposed a two-stage Predict–Repair–Optimize (PRO) framework, consisting of an offline phase and an online phase, to achieve high-quality solutions in real-time; refer to Figure 2.

3 Predict-Repair-Optimize Framework

This study utilizes the mixed integer linear programming (MILP)-based rTPP established in [6] as the foundational optimization model. The model simultaneously calculates modified arrival and departure times, dwell durations, waiting intervals, platform allocations, and routing choices for each train, facilitating synchronized rescheduling and rerouting during disruptions. This section first introduces a generalized rTPP formulation and subsequently elucidates the interaction of each stage of the proposed PRO framework with this model during

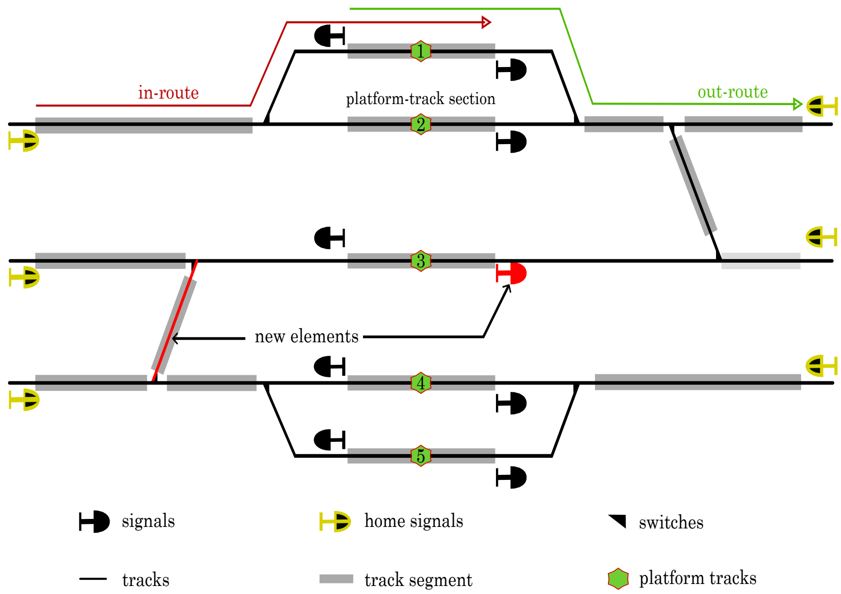

Fig. 1. A schematic of station area and elements

both offline and online phases.

$$
\begin{aligned}
\text{(P):} \quad & \min_{x} \ c^T x && (1)\\
\text{s.t.} \quad & A_1 x = b_1 && (2)\\
& A_2 x \leq b_2 && (3)\\
& A_3 x \geq b_3 && (4)\\
& x_i \in Z \quad \text{for } i \in \mathcal{I} && (5)\\
& x_i \in \{0,1\} \quad \text{for } i \in \mathcal{B} && (6)\\
& x_i \in R \quad \text{for } i \in \mathcal{C} && (7)
\end{aligned}
$$

The objective function (1) seeks to minimize the cumulative weighted effect of disruptions, encompassing cancellations, delays, and reroutings (i.e., alterations in routes and platform assignments). The constraints (2) ensure that each train is allocated precisely one viable inbound and outbound route through the station. Constraints (3) stipulate that only one train may occupy a track segment at any given moment. Constraints (5) impose minimum headway requirements between successive train movements on the identical track segment. The constraints (6)–(8) delineate the domain for the decision variables. The decision vector is represented as $x \in R^n$, accompanied by constraint matrices $A_1 \in R^{m_1 \times n}$, $A_2 \in R^{m_2 \times n}$, and corresponding right-hand side vectors $b_1 \in R^{m_1}$, $b_2 \in R^{m_2}$, where $m = m_1 + m_2$. Consequently, it can be concisely expressed as $Ax\{\geq, =, \leq\}b$, where $A = A_1 \cup A_2 \cup A_3$ and $b = b_1 \cup b_2 \cup b_3$.

Prediction step. This step comprises an offline training stage and an online inference stage, as illustrated in Figure 2. We train supervised machine learning models, known as *scopers*, offline on disruption scenarios and their associated high-quality solutions. The *scoper* acquires knowledge from attributes defining each disruption scenario (e.g., duration, number of impacted trains, or directly affected routes) and binary route classifications. In a disruption scenario, the *scoper* is trained to predict the stability of nominal route utilization, specifically whether a route designated in the nominal timetable is anticipated to remain consistent for all trains typically assigned to this route. The ML models facilitate generalization across various timetables and disruption scenarios by deriving the label set from the infrastructure layout instead of a specific timetable. The rescheduling solutions derived from the MILP model (P) are utilized to extract route-stability labels, which are subsequently combined with engineered features to create a labeled, scenario-level dataset. The dataset is randomly divided into a training set (80%, encompassing fitting and tuning) and a hold-out test set (20%) to assess out-of-sample predictive performance. We utilize logistic regression (LR) and random forest (RF) as exemplars of multi-label classification models [8, 15]; comprehensive training specifics (cross-validation, feature selection, and hyperparameter optimization) are detailed in [19].

During the online stage, the trained *scoper* is utilized on unencountered disruption scenarios to forecast probability estimates for each standard route. Routes are classified as stable or unstable using a default threshold of 0.5 [11]. According to the nominal timetable, these predictions are subsequently converted into variable fixations within the MILP model (P): for routes deemed stable, all corresponding route-assignment binaries are fixed to their nominal timetable values, whereas variables associated with unstable routes retain flexibility for the solver. This results in a simplified problem (P_{red}) derived by assigning a subset of binaries x_i for $i \in \mathcal{B}' \subseteq \mathcal{B}$ to their predicted values $\hat{x}_i \in \{0, 1\}$. Route assignments are explicitly defined in the nominal timetable, allowing route-level predictions to be directly correlated with train-level routing decisions, thereby significantly decreasing the problem size.

$$(\mathrm{P_{red}}) : \min_x \quad c^\top x \quad \text{s.t.} \quad Ax\{\geq, =, \leq\}b, \quad x_i = \hat{x}_i \quad \forall i \in \mathcal{B}' \tag{8}$$

As scoping solely constrains spatial decisions (route selections), the MILP retains the flexibility to modify temporal variables such as arrival and departure times, which can frequently accommodate minor prediction inaccuracies. This flexibility is, however, constrained by temporal limitations (maximum dwell and waiting periods). Thus, an erroneous fixation may render the reduced model (P_{red}) infeasible, indicating that the constraint system $Ax\{\geq, =, \leq\}b$ cannot be satisfied with the fixed binaries. Consequently, we implement a repair step that selectively removes a minimal subset of these fixations to restore feasibility while maintaining the original model structure.

Reparation step. To address infeasibility in the reduced problem ($\mathrm{P_{red}}$), we initially construct a repair subproblem that relaxes the minimal subset of previously assigned binary variables. An auxiliary variable is introduced: binary

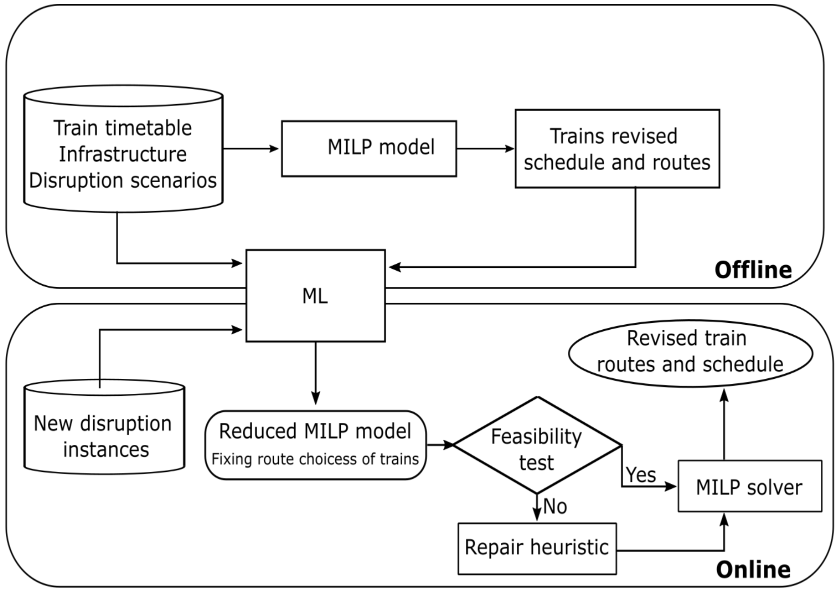

Fig. 2. Overview of the PRO framework for rTPP

deviation indicators $d_i \in \{0,1\}$ for $i \in I_v \cap \mathcal{B}$, which denote alterations in fixed binary assignments. The repair subproblem can be formulated as follows:

$$\text{(Repair): min} \quad \sum_{i \in I_v \cap \mathcal{B}} d_i \tag{9}$$

$$\text{s.t.} \quad \sum_{i \in I_v} A_{ji} x_i = b_j \qquad \forall j \in \mathcal{V} \tag{10}$$

$$\sum_{i \in I_v} A_{ji} x_i \leq b_j \qquad \forall j \in \mathcal{V} \tag{11}$$

$$\sum_{i \in I_v} A_{ji} x_i \geq b_j \qquad \forall j \in \mathcal{V} \tag{12}$$

$$d_i \geq x_i - \hat{x}_i, \quad d_i \geq \hat{x}_i - x_i \qquad \forall i \in I_v \cap \mathcal{B} \tag{13}$$

$$x_i \in R, \xi_j \geq 0 \qquad \forall i \in I_v, j \in \mathcal{V} \tag{14}$$

$$x_i \in Z \qquad \forall i \in I_v \cap \mathcal{I} \tag{15}$$

$$x_i \in \{0,1\}, d_i \in \{0,1\} \qquad \forall i \in I_v \cap \mathcal{B} \tag{16}$$

The objective function (9) seeks to minimize the total deviations d_i. Imposing penalties on binary deviations allows the model to regain feasibility with minimal changes, facilitating a structured trade-off. Constraints (10) – (12) address the relaxation of violated constraints, contingent upon their initial form $(\geq, =, \leq)$. The constraints (13) delineate the deviation indicator d_i. Addressing the subproblem (Repair) reinstates feasibility to $\mathrm{P}_{\mathrm{red}}$ by identifying violated constraints and carefully modifying the corresponding binary variables.

Optimization step. Upon restoring feasibility through the reparation step, we implement a focused post-repair improvement heuristic that selectively relaxes a limited subset of previously resolved routing decisions to augment solution quality. We begin by solving the reduced rTPP (P_{red}) using a standard MILP solver to obtain an incumbent solution. From this incumbent, we identify trains that continue to experience significant disruption impacts (e.g., cancellations or delays) and broaden the set using their adjacent entities in the precomputed conflict graph, thereby encompassing interacting trains that may indirectly hinder improvements. In this localized conflict area, we subsequently relax a limited subset of routing decisions (inbound, outbound, or both), determined by scoping, that possess the highest conflict weight. The resulting partially unfixed model is re-solved, allowing the solver to explore better rerouting options while keeping the released decision space tightly controlled.

4 Computational results

This section presents the details of the investigated case study and analysis of results obtained from computational experiments. The selected railway station for this study is a medium-sized station in Germany, featuring eight platforms and up to 125 routes. The train schedule analyzed corresponds to a six-hour window (5:00 AM–11:00 AM), encompassing over 250 train movements. We randomly select 21 infrastructure components (approximately 12% of the station's components) for disruptions, comprising platform unavailability and switch failures, dispersed throughout the station area. A total of 1,342 scenarios were generated using the MILP-based instance-generation method, comprising 1,070 training instances and 272 test instances. During dataset construction, each training instance was solved with an hour time limit; only solutions within a 2% optimality gap were used for labeling. In total, 662 features were engineered to describe each disruption scenario. After feature selection, the LR and RF models were trained using 537 and 337 features, respectively.

4.1 Prediction performance

The prediction performance of the trained LR and RF models is evaluated on the test set before deploying them as *scopers* in the reduction framework. This offers an initial indication of how effectively the predictions may improve the solution quality of the MILP model solving the reduced problem instances. We measure performance via the confusion matrix (juxtaposing the predicted with the actual route labels for each scenario) [7], as each prediction outcome has a different operational impact.

True positives (TP) are correctly identified stable nominal routes; fixing the associated binaries is therefore safe and reduces problem size. True negatives (TN) correctly flag routes as nominally unstable; these variables remain free, so they do not directly generate speed-ups, but they also do not compromise feasibility. Misclassifications are more critical. False positives (FP)— actual unstable routes predicted as stable—lead to incorrect fixations that may block necessary rerouting, worsen solution quality, or even cause infeasibility. False negatives (FN) occur when actual stable routes are not fixed; this mainly reduces potential computational gains, while feasibility and optimality are typically preserved because the solver can still choose the nominal route. This creates a clear trade-off: more aggressive fixations can speed up computation but raise the risk of false positives, whereas conservative fixations better protect solution quality at the cost of smaller speed-ups.

We focus on sample precision as the primary metric, measuring the proportion of TP with respect to all predicted positives (TP + FP), averaged across the test instances. This is because FP are the most detrimental outcome in our reduction setting: an incorrectly fixed route can eliminate essential rerouting options and may trigger infeasibility. Table 1 shows that RF consistently outperforms LR with reference to precision. Beyond its higher precision, RF more frequently produces clean predictions—either matching the full label set or, more importantly, avoiding false positives altogether. This behavior is particularly valuable for safe problem reduction, as it enables more confident fixations while limiting the risk of over-restricting the MILP.

Table 1. Prediction performance of Random Forest vs. Logistic Regression ($n = 272$)

	Random Forest	Logistic Regression
Precision	0.9809	0.9757
Instances with correct Predictions only	20/272 (7.35%)	15/272 (5.51%)
Instances without False Positives	73/272 (26.84%)	47/272 (17.28%)

4.2 Computational Speed-Up

Table 2 presents a subset of instances, labeled I1–I8, where each label encodes three key features: "dXX" denotes the disruption duration in minutes; "p" or "offp" indicates whether the disruption occurs during peak or off-peak hours within the station area. To keep real-time criteria, all instances were solved within a time limit of 180 seconds [13].

Table 2 compares CPLEX with the proposed PRO framework in terms of runtime and objective value. For the moderate disruption instances (I2, I3, I5), PRO preserves the same objective value as CPLEX while achieving substantial speed-ups, indicating that the learned scoping and subsequent repair do not compromise solution quality when the disruption is still "recoverable" without extensive change in operational measures (cancel, delays, or rerouting). In the easier instances (e.g., I1), both approaches already solve quickly, and PRO offers limited additional benefit–an expected outcome when the baseline problem is

Table 2. Performance comparison between CPLEX and PRO method

Instances	CPLEX		PRO		Change (%)	
	Time (s)	Obj	Time (s)	Obj	Time	Obj
I1_d30_offp	10	187.75	10.3	187.75	+3.0	0
I2_d30_p	35	26	15.2	26	-25.7	0
I3_d45_offp	9.9	11	4.2	11	-57.4	0
I4_d45_p	180	255	180	85	0	-66.6
I5_d50_offp	11.2	19.75	7.62	19.75	-31.9	0
I6_d55_p	180	75.75	180	60	0	-20.5
I7_d60_offp	180	193.75	180	186.5	0	-3.7
I8_d75_offp	180	144	180	23	0	-84.0

not computationally challenging. The main advantage of PRO becomes visible in the hard instances that reach the time limit under CPLEX (I4, I6–I8). In these cases, CPLEX likely returns suboptimal incumbents within the fixed time budget, whereas PRO delivers markedly better objective values under the same cap, suggesting that early variable fixation provides stronger search guidance and helps the solver focus on high-impact routing decisions. The improvements are particularly pronounced in instances such as I4 and I8, suggesting that PRO provides stronger guidance for the search and helps CPLEX focus on high-impact decisions, leading to better incumbents when the direct MILP solve shows slow incumbent improvement and limited gap reduction within the fixed time limit.

Overall, these results highlight the efficiency and practicality of the PRO framework under varied disruption conditions; however, to fully assess its applicability, further testing is required for more complex disruption settings, such as long disruption durations during peak hours or multiple simultaneous infrastructure failures.

4.3 Conclusion and Future directions

This study introduced a PRO framework that efficiently addresses rTPP under disruptions, incorporating prediction, feasibility repair, and final optimization. The proposed framework exhibits computational efficiency while maintaining solution quality in most scenarios. In the future, we will focus on improving solution quality through the development of lightweight primal heuristics that enhance the objective function post-feasibility restoration, without incurring significant computational overhead. Moreover, additional validation regarding large-scale disruptions and multi-failure scenarios is crucial for evaluating the framework's scalability.

References

1. Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., Wagenaar, J.: An overview of recovery models and algorithms for real-time railway rescheduling. Transportation Research Part B: Methodological **63**, 15–37 (2014)

2. Cadarso, L., Maróti, G., Marin, A.: Smooth and controlled recovery planning of disruptions in rapid transit networks. IEEE Transactions on Intelligent Transportation Systems **16**(4), 2192–2202 (2015)
3. Chakroborty, P., Vikram, D.: Optimum assignment of trains to platforms under partial schedule compliance. Transportation Research Part B: Methodological **42**(2), 169–184 (2008)
4. Fischetti, M., Monaci, M.: Using a general-purpose mixed-integer linear programming solver for the practical solution of real-time train rescheduling. European Journal of Operational Research **263**(1), 258–264 (2017)
5. García-Ródenas, R., López-García, M.L., Cadarso, L., Codina, E.: An efficient greedy heuristic for the real-time train platforming problem. Computers & Operations Research **164**, 106525 (2024)
6. Haque, M.T., Eisold, J., Bešinović, N.: A quantitative framework for assessing rail station operational resilience under infrastructural modifications. Available at SSRN 5291587 (2025)
7. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2 edn. (2009)
8. Hizir, A.E., Barnhart, C., Vaze, V.: Large-scale airline crew recovery using mixed-integer optimization and supervised machine learning. Available at SSRN 5180044 (2025)
9. Lamorgese, L., Mannino, C.: An exact decomposition approach for the real-time train dispatching problem. Operations Research **63**(1), 48–64 (2015)
10. Lusby, R.M., Larsen, J., Ehrgott, M., Ryan, D.M.: A set packing inspired method for real-time junction train routing. Computers & Operations Research **40**(3), 713–724 (2013)
11. Morabit, M., Desaulniers, G., Lodi, A.: Learning to repeatedly solve routing problems. Networks **83**(3), 503–526 (2024)
12. Pascariu, B., Sama, M., Pellegrini, P., D'ariano, A., Pacciarelli, D., Rodriguez, J.: Train routing selection problem: Ant colony optimization versus integer linear programming. IFAC-PapersOnLine **54**(2), 167–172 (2021)
13. Pascariu, B., Sama, M., Pellegrini, P., d'Ariano, A., Rodriguez, J., Pacciarelli, D.: Formulation of train routing selection problem for different real-time traffic management objectives. Journal of Rail Transport Planning & Management **31**, 100460 (2024)
14. Pellegrini, P., Pesenti, R., Rodriguez, J.: Efficient train re-routing and rescheduling: Valid inequalities and reformulation of recife-milp. Transportation Research Part B: Methodological **120**, 33–48 (2019)
15. Rashedi, N., Sankey, N., Vaze, V., Wei, K.: A machine learning approach for solution space reduction in aircraft disruption recovery. European Journal of Operational Research **323**(1), 297–308 (2025)
16. Samà, M., Corman, F., Pacciarelli, D., et al.: A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations. Computers & Operations Research **78**, 480–499 (2017)
17. Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., Pacciarelli, D.: Ant colony optimization for the real-time train routing selection problem. Transportation Research Part B: Methodological **85**, 89–108 (2016)
18. Samà, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., Pacciarelli, D.: On the tactical and operational train routing selection problem. Transportation Research Part C: Emerging Technologies **76**, 1–15 (2017)

19. Sehmisch, S., Szymula, C., Haque, M.T., Bešinović, N.: A data-driven problem reduction framework for local train rescheduling in station areas. In: 2025 IEEE International Conference on Intelligent Rail Transportation (ICIRT). pp. 194–199. IEEE (2025)
20. Yongxiang, Z., Qingwei, Z., Yong, Y., Xu, Y., Qiyuan, P.: A fast approach for reoptimization of railway train platforming in case of train delays [j]. Journal of Advanced Transportation **2020**, 1–20 (2020)
21. Zhang, Q., Lusby, R.M., Shang, P., Zhu, X.: Simultaneously re-optimizing timetables and platform schedules under planned track maintenance for a high-speed railway network. Transportation Research Part C: Emerging Technologies **121**, 102823 (2020)