Multi-objective disruption management for robust railway operations

Léa Ricard [*] Michel Bierlaire [*] Salma Derouiche [†] Matthias Hellwig [†]

January 28, 2026

Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

[*]École Polytechnique Fédérale de Lausanne (EPFL), School of Architecture, Civil and Environmental Engineering (ENAC), Transport and Mobility Laboratory, Switzerland, {lea.ricard,michel.bierlaire}@epfl.ch
[†]SMA and partners Ltd, Switzerland {S.Derouiche, M.Hellwig}@sma-partner.com

Contents

1 Introduction **2**

2 Literature review **2**

3 Problem definition **4**
3.1 Recovery decisions . . . 5
3.2 Passenger assignment . . . 5

4 Event-activity graph **6**
4.1 Events (nodes) . . . 6
4.2 Activities (arcs) . . . 6

5 Mathematical formulation **7**
5.1 Operational Constraints . . . 9
5.2 Passenger assignment constraints . . . 10
5.3 Objective functions . . . 12
5.4 Linearization and valid inequalities . . . 13

6 Adaptive large neighborhood search heuristic **14**
6.1 General framework . . . 14
6.2 Passenger assignment algorithm . . . 16
6.3 Neighborhood operators . . . 16
6.3.1 Destroy operators . . . 16
6.3.2 Insertion ordering . . . 17
6.3.3 Repair operators . . . 18
6.3.4 Combined operators . . . 19
6.4 Initial solutions . . . 19

7 Computational experiments **21**
7.1 Case study . . . 21
7.2 Heuristic performance . . . 22
7.3 Results . . . 24
7.3.1 Partial blockage . . . 24
7.3.2 Complete blockage . . . 25

8 Conclusions **27**

A Notation for the train rescheduling problem **28**

B Code and data **29**

1 Introduction

Railway systems are highly sensitive to disruptions, which can lead to cascading delays, train cancellations, and passenger dissatisfaction. Effective timetable rescheduling is therefore essential to restore operations and minimize passenger inconvenience when disruptions occur. The management of rail disruptions is particularly challenging due to the numerous operational constraints and the combinatorial nature of the rescheduling decisions.

In this context, the present work aims to build a decision-support tool for managing railway disruptions that integrates operational and passenger perspectives within a multi-objective optimization framework. The proposed approach considers the sometimes divergent interests of the system's actors, including passenger needs, deviations from the reference timetable, and operational costs, while respecting infrastructure capacity constraints. Unlike previous research focusing on fully automatic timetable optimization, this project emphasizes practical applicability: it seeks to assist dispatchers and planners in "repairing" disrupted timetables through an interactive, data-driven method embedded within existing operational tools.

To achieve this objective, the project integrates and extends previous research conducted at the Transport and Mobility Laboratory, particularly the PhD thesis *"Integration of passenger satisfaction in railway timetable rescheduling for major disruptions"* by Stefan Binder, which introduced one of the first multi-objective, passenger-centric methodologies for disruption management. Furthermore, the proposed tool is designed to interface with *Viriato*, the professional software used by SBB for timetable planning, enabling the evaluation of algorithms on real production data. The research direction and objectives have been discussed with Pierre Dominguez, Data Science Project Manager at SBB, who provided input from SBB's perspective on the practical needs and requirements for disruption management tools.

Building on these foundations, this work formulates an event–activity model that jointly reschedules trains and reassigns passengers under disruptions at a near-microscopic level. While macroscopic models allow for fast optimization on large-scale networks at the cost of infrastructure detail, near-microscopic formulations capture track-level operations and local routing decisions with higher accuracy but greater computational effort. By operating at this level of detail, the proposed model can represent multiple parallel tracks and realistic track changes within stations, enabling a more accurate depiction of operational constraints during disruptions. Passenger groups dynamically adapt to the new disposition timetable through a utility-based assignment mechanism. An exact MILP formulation is proposed alongside an adaptive large neighborhood search (ALNS) heuristic to efficiently explore the solution space. Together, they provide a computationally tractable and behaviorally realistic framework for disruption management, unifying macroscopic scalability with near-microscopic fidelity.

The remainder of this report is organized as follows. Section 2 reviews the relevant literature on timetable rescheduling and passenger assignment under disruptions. Section 3 defines the problem, detailing the recovery decisions and the passenger assignment component. Section 4 introduces the event–activity graph representation. Section 5 presents the mathematical formulation, including operational constraints (Section 5.1), passenger assignment constraints (Section 5.2), objective functions (Section 5.3), and the linearization and valid inequalities (Section 5.4). Section 6 describes the ALNS heuristic, outlining the general framework and the tailored neighborhood operators developed for our formulation. Section 7 reports the computational experiments, describing the case study, analyzing the heuristic performance, and illustrating representative solutions for two disruption scenarios. Section 8 concludes the study and outlines directions for future research.

2 Literature review

In this section, we focus our review on recent contributions where the railway network is modeled at the macroscopic or near-microscopic level and which deal with major disruptions. We refer readers interested in the timetable rescheduling problem in other settings to the surveys of Cacchiani et al. (2014) and Sharma et al. (2023).

Veelenturf et al. (2016) presented an event-activity model for the timetable rescheduling problem, which minimizes the number of cancellations and delays. The underlying event–activity graph adopts a time-continuous

representation, where nodes correspond to events (i.e., arrivals or departures) and arcs represent activities (such as running, dwelling, or transferring). Their method allows for train rerouting and short-turning and optimizes for all stages of the disruption — before, during, and after a disruption. Computational tests are carried out over the Dutch railway network, composed of one, two, or more tracks between each pair of stations. Their method can address relatively large-scale instances in a reasonable time, but does not consider the assignment of passengers to routes.

This gap is filled in Binder et al. (2017*b*), where a dynamic passenger assignment (i.e., passenger flows dynamically adapted to the disposition timetable) is implemented. Unlike the approach in Veelenturf et al. (2016), Binder et al. (2017*b*) formulates the problem using a space-time graph. This graph captures both infrastructure and passenger movements within a time-expanded network, where each node corresponds to a specific location at a given time step. Their model has three objectives: passenger satisfaction, operational costs, and deviation from the undisrupted timetable. The first objective is directly optimized, whereas the other two objectives are added as ϵ-constraints. The model is tested on a part of the Dutch railway network. Since this model is more complex than the one of Veelenturf et al. (2016), only smaller instances can be solved to optimality.

Zhu and Goverde published two papers on the timetable rescheduling problem. In their first one, Zhu & Goverde (2019), the passenger flows are static. Zhu & Goverde (2020) extends their model to dynamic passenger flows. Both models use event-activity networks and aim at minimizing passenger delays. The capacity at the stations is taken into account. Furthermore, flexible short-turning (i.e., trains can turn at any station along their itinerary) is implemented. In Zhu & Goverde (2019), an instance composed of a part of the Dutch network is solved optimally for hundreds of disruption scenarios. A tailored fix-and-optimize algorithm is developed in Zhu & Goverde (2020) to solve an instance of the same order of magnitude heuristically.

Altazin et al. (2020) presented an event-activity model for the timetable rescheduling of dense railway networks (e.g., commuter rail networks). A simulation-based method is proposed to compute the objective values associated with a disposition timetable, namely the recovery time, the quality of the service, and the number of decisions. Two greedy heuristics are benchmarked against a full-enumeration method on the French Transilien railway network.

A bi-objective MILP formulation for the timetable rescheduling problem with a seat reservation mechanism and dynamic passenger reassignment is introduced in Hong et al. (2021). The two objectives considered are to maximize the number of passengers reaching their destination in the disrupted network and minimize the train delays. Experiments are carried out on the Beijing-Shanghai high-speed train line.

Zhan et al. (2021) developed an efficient alternating direction method of multipliers (ADMM) algorithm for the timetable rescheduling problem. This method decomposes the problem into two subproblems: a train rescheduling problem and a passenger routing problem. The advantage of this method is that each subproblem can be reformulated as a shortest-path problem, for which fast dynamic programming algorithms are available.

Extension of the previous works to large-scale networks is presented in Zhang et al. (2023). However, to be able to tackle real-life networks with up to 47 stations and 350 trains, the passenger assignment is considered as static. An efficient heuristic based on the Lagrangian relaxation is proposed. This heuristic decomposes the problems into independent train-based subproblems and dynamically generates constraints. Solutions to real-life networks with an average optimality gap of 2.27% are obtained within a time limit of 300 seconds. In comparison, in Binder et al. (2017*b*), instances of up to 11 stations and 24 trains were solved to optimality within a time limit of 3600 seconds. Also, in Zhu & Goverde (2020), instances of up to 17 stations and 72 trains were solved heuristically within a time limit of 300 seconds.

Finally, Xiu et al. (2024) proposed a near-microscopic level model for the timetable rescheduling and train rerouting problems. This model accounts for track reallocation within each station by incorporating station-related arcs to account for station capacity. They also presented an improved ADMM heuristic featuring a heuristic sorting rule.

Table 1 summarizes the main attributes of the literature on the timetable rescheduling problem reviewed. The columns display the representation level of the railway network, either macroscopic (Macro) or near-microscopic (Near-micro), the graph used, the solution method employed (Algo.), whether multiple objectives are considered or not (Mult. obj.), the recovery decisions implemented, the type of passenger assignment (Pass. assign.), the type of tracks, whether vehicle capacity is constrained or not (Veh. capa.), and the size of the largest

instance solved.

Author(s)	Level	Graph	Algo.	Mult. obj.	Recovery decisions	Pass. assign.	Tracks	Veh. capa.	Largest instance size
Veelenturf et al. (2016)	Macro	Event-activity	MILP	•	C, D, O, R, ST	None	Multiple		S = 26, T = 161, D = 2h, H = 2.75h
Binder et al. (2017*b*)	Macro	Space-time	MILP	•	A, C, D, E, O, R	Dynamic	Double	•	S = 11, T = 24, D = 2h, H = 2h
Zhu & Goverde (2019)	Macro	Event-activity	MILP		A, C, D, O, ST	Static	Single, double		S = 17, T = 48, D = 3h, H = 4h
Altazin et al. (2020)	Macro	Event-activity	SIM, H	•	A, C, D, ST	Dynamic	Single, double	•	S = 29, T = 25, D = 20 min, H = 2.5
Zhu & Goverde (2020)	Macro	Event-activity	MH		A, C, D, O, ST	Dynamic	Double		S = 17, T = 72, D = 2h, H = 3h
Hong et al. (2021)	Near-micro	Space-time	MILP	•	A, D, O, R	Dynamic	Double	•	S = 8, T = 20, D = 55 min, H = 4h
Zhan et al. (2021)	Near-micro	Space-time	ADMM, DP	•	C , D, O	Dynamic	Double	•	S = 17 , T = 20, D= 2h, H = 6h
Zhang et al. (2023)	Macro	pace-time	LR-H	•	C, D, O, R	Static	Double		S = 47, T = 350, D = 6h, H = 18h
Xiu et al. (2024)	Near-micro	Space-time	ADMM		C, D, O, R	Static	Single, double	•	S = 15, T = 36, D = 2h, H = 6.5h
Our work	**Near-micro**	**Event-activity**	**MILP, ALNS**	•	**A, C, D, E, O, ST**	**Dynamic**	**Multiple**	•	**S = 25, T = 58, D = 1h, H = 2h**

Table 1: Characteristics of selected railway rescheduling problems

Algorithm: ADMM — alternating direction method of multipliers, ALNS — adaptive large neighborhood search, DP — dynamic programming, H — heuristic, LR-H — Lagrangian relaxation heuristic, MILP — mixed integer linear optimization method, MH — metaheuristic, SIM — simulation
Recovery decisions: A — adding or skipping stops, C — canceling, D — delaying, E — emergency vehicle, O — ordering, R — rerouting, ST - short-turning trains
Largest instance size: S — number of stations, T — number of trains, D — disruption duration, H — planning time horizon

Our work, summarized in the last row, introduces the first near-microscopic model integrating both dynamic passenger assignment and multiple track configurations, including flexible track changes at station levels. Existing models cannot represent these elements simultaneously, which motivates the development of our new event–activity–based formulation. This formulation integrates station-track and section-track occupation information together with an aggregated representation of passenger movements. By using two complementary levels of detail—near-microscopic for train movements and macroscopic for passenger flows—our approach achieves computational efficiency without losing any relevant information. Furthermore, we propose an ALNS heuristic, enabling the solution of relatively large networks within practical time limits.

3 Problem definition

The timetable rescheduling problem consists of rerouting trains and passengers in a disrupted railway network. In our case, the objectives are to minimize passenger inconvenience, operational costs, and deviation from the planned timetable. In this section, we describe the problem and introduce its notation. The recovery decisions used for building a disposition timetable are defined in Section 3.1. Furthermore, the passenger assignment subproblem is introduced in Section 6.2.

At the near-microscopic level, the infrastructure is represented by a set S of stations, of which a subset is part of the set of bifurcations $B \subseteq S$ (i.e., stations without a platform for passengers to board and egress trains). All non-bifurcation stations and some bifurcations are assigned a set P_s of station tracks. Let Q denote the set of section tracks, where each track $q \in Q$ connects two neighboring stations. Trains can change section track at any station (including bifurcations). Each section track can in principle be used in both directions, although in double- or multi-track systems, tracks are typically assigned a predefined operating direction. The minimum travel time required to traverse a section track varies depending on the train type, and this information is provided as input to the problem.

Let $K = K^t \cup K^b$ be the set of vehicles, where K^t is the set of trains and K^b is the set of emergency buses. For each vehicle $k \in K$, let q_k be its passenger capacity. Rolling stock is stored overnight in shunting yards that are located at some stations. Let $S_b \subseteq S$ denote the subset of stations with a shunting yard. In the scheduled

timetable, each train is defined by an itinerary composed of a sequence of stations as well as the arrival and departure times at each station. Furthermore, the station track assigned at each station and the section tracks used are given.

Let G be a set of passenger groups, where each group $g \in G$ is associated with a triple (o_g, d_g, t_g), where $o_g, d_g \in S$ are the origin and destination stations, respectively, and t_g is the desired departure time. We are also given the size n_g of each group $g \in G$. Each passenger group is assigned, if possible, to a specific path in the planned and disposition timetables, as we detail in Section 6.2. Otherwise, the group exits the system, resulting in a significant penalty.

Finally, information about the disruption is provided, including the affected section or station tracks and the corresponding start and end times. The affected tracks are assumed to be completely unavailable in both directions, and if multiple tracks are impacted, they share the same period of unavailability.

3.1 Recovery decisions

In railway timetable rescheduling, various recovery decisions can be implemented to manage disruptions and ensure service continuity. We implement the following decisions:

- Canceling—Removing a train's run from the schedule completely or partially.
- Delaying—Postponing the departure or arrival of a train beyond its scheduled time.
- Ordering—Adjusting the sequence in which trains are dispatched to optimize traffic flow and track usage.
- Using an emergency bus—Deploying an additional bus to supplement disrupted services.
- Adding / skipping stops—Modifying train paths to make unscheduled stops or to bypass certain stops.
- Short-turning trains—Terminating a train's journey early and initiating a return trip from that point, retracing the same stations as the partial journey but in reverse order.

3.2 Passenger assignment

Each passenger group is typically assumed to be indivisible. Note that a group can consist of one or several individuals. Additionally, for simplicity, demand is assumed to be inelastic. This means that the desired origin and destination locations, as well as the desired departure time of each individual, remain the same in the event of a disruption.

Let R_{o_g, d_g} be the set of all paths linking origin location o_g to destination location d_g. Each path may include access, in-vehicle, dwelling, transfer, and egress activities, or represent a penalty path when the group cannot be routed within the disposition timetable and must exit the system. A utility function is associated with each path, and it is assumed that all passengers select the path with the highest utility. The utility function for a path r is taken from Robenek et al. (2016), which is based on the following attributes:

- In-vehicle time (VT_r)—The duration, in minutes, that the passenger group spends on one or more trains in movement along the route.
- Waiting time (WT_r)—The duration, in minutes, that the passenger group spends waiting in a train or between consecutive trains (excluding the waiting time for the first train).
- Number of transfers (NT_r)—The number of times the passenger group needs to switch trains along the path.
- Early departure ($ED_r = \max(0, d_g - t)$)—The time difference, in minutes, between the desired departure time (d_g) and the actual departure time (t) from the origin, if the departure is earlier than desired.
- Late departure ($LD_r = \max(0, t - d_g)$)—The time difference, in minutes, between the actual departure time (t) and the desired departure time (d_g) from the origin, if the departure is later than desired.

The utility of a path is normalized based on in-vehicle time to generate generalized travel time values. Additionally, the ticket price of a path is not considered, as we assume that in the event of a disruption, passengers will not need to purchase a new ticket if they change their itinerary. Given these considerations, the utility of a path r is defined as:

$$V_r = -(VT_r + \beta_1 WT_r + \beta_2 NT_r + \beta_3 ED_r + \beta_4 LD_r), \tag{1}$$

4 Event-activity graph

In the event–activity graph, events represent points in time, while activities correspond to processes with a non-negative duration linking two events. Both events and activities are divided into two categories: disaggregated (train-related) and aggregated (passenger-related). The former represent vehicle movements, whereas the latter represent passenger movements. Let $G(N, A)$ be the event-activity graph, where $N = N_{dis} \cup N_{agg} \cup N_b \cup N_o \cup N_d$ denotes the set of events, including disaggregated events N_{dis}, aggregated events N_{agg}, shunting events N_b, and passengers' origin and destination events N_o and N_d. The set A is the set of activity arcs. Next, we define these events and activities in detail.

4.1 Events (nodes)

Each arrival and departure event i is represented by a node $(e_i, s_i, p_i, k_i, tp_i)$, where:

- e_i indicates whether the event is an arrival ($e_i = 1$) or a departure ($e_i = 0$).
- $s_i \in S$ represents the station.
- $p_i \in P_s$ denotes the station track (if applicable).
- k_i is the vehicle identifier.
- tp_i is the event type: 'regular' ($tp_i = 0$), 'rerouting' ($tp_i = 1$), or 'emergency' ($tp_i = 2$).

Aggregated events are defined only for regular stations, as bifurcations are not considered at the aggregated level—passengers cannot board or alight at these locations, making such information irrelevant from a passenger-utility perspective. Moreover, events at bifurcations are not necessarily associated with a specific station track when such tracks do not exist in the infrastructure.

Regular events are events that are in the planned timetable. For each regular event i, the corresponding planned time t_i^k is known. Rerouting events are events associated with the potential short-turning of trains. A set of rerouting events is created for each train for which we add a short-turning possibility, as explained in Section 4.2. Emergency bus events are created at the origin (departure event) and destination (arrival event) of potential emergency bus itineraries. Thus, for each emergency bus, there are two associated events. Not all emergency events are associated with a station track since these events happen outside the railway network.

4.2 Activities (arcs)

Activities link pairs of events and, in the case of train running activities, are associated with a section track. The event–activity network G is therefore a directed multigraph, since multiple activities may connect the same pair of events.

The activities are described using the following additional notation. Let c denote the operational costs of an emergency bus per kilometer traveled, and let $d_{i,j}$ be the length, in kilometers, of the section track associated with arc (i, j). Additionally, let y_i denote the arrival or departure time of event i, such that $y_j - y_i$ is the duration of arc (i, j).

Train activities include running, waiting, pass-through, short-turning, starting a journey, and ending a journey. Emergency buses are modeled as an additional activity type. The characteristics of these activities—namely, the incoming event i, the outgoing event j, the associated section track $q \in Q$ (if applicable), and the

cost c_a—are detailed in Table 2. Each train begins and ends its circulation with a starting and an ending activity, which connect its first and last event to a shunting-yard node $r \in N_b$; in between these two activities, the train performs a sequence of running, waiting, pass-through, or short-turning activities.

Running activities connect a departure event at station s to an arrival event at station $s' \neq s$ for the same train. Each running activity is associated with a section track $q \in Q$. Let S_q denote the set of stations connected by section track q, with $|S_q| = 2$ for all $q \in Q$. Let P_q be the set of station tracks accessible from section track q, and conversely, let Q_p be the set of section tracks accessible from station track p. For each pair of connected stations s, s', we create a running activity for every section track q with $S_q = \{s, s'\}$ whenever $q \in Q_p$ for the station track p of the incoming event and the station track of the outgoing event belongs to P_q. All accessibility information is obtained from *Viriato*.

Waiting and pass-through activities link an arrival event to a departure event at the same station and, when applicable, on the same station track. Only pass-through activities can occur at bifurcations, since waiting at a station allows passengers to board or alight from trains, which is not possible at bifurcations. However, if a bifurcation is equipped with station tracks, the corresponding pass-through activity may have a positive duration.

We create a short-turning activity for each train scheduled to pass through the disrupted area during the disruption. A short-turning activity is preceded and followed by a waiting or pass-through activity and occurs between two events at the same station and on the same station track. This station corresponds to the last station equipped with a shunting yard that the train passes before entering the disrupted area. For trains assigned a short-turning activity, we generate a copy of the events and of the running, waiting, pass-through, and ending activities that occur before the short-turning, replicating them in reverse order. The copied events are assigned $tp = 1$ and are given a scheduled time equal to the scheduled time of the event at the start of the short-turning.

Finally, for each short-turning activity, we create a corresponding emergency bus activity linking the last stations equipped with a shunting yard before the disrupted area on each of its sides. The scheduled times of these activities are set to ensure that passengers have sufficient time to transfer between the emergency bus and the corresponding short-turned train. An emergency bus activity (i, j) incurs an operational cost of $c \cdot d_{i,j}$. Emergency buses do not have starting or ending activities.

Passenger activities, summarized in Table 3, include in-vehicle, dwelling, transferring, access, egress, and penalty activities. The structure of passenger running, dwelling, access, and egress activities parallels the corresponding train activities: running, waiting, starting, and ending, but they are aggregated by not considering station and section tracks and skipping bifurcations, where passengers cannot board or egress vehicles. Only access and egress activities are associated with a passenger group.

Transferring activities occur between two events at the same station but associated with different vehicles (either between two trains or between a train and a bus, and vice versa). To limit the number of possible transfers, we allow transfers only at predefined *transfer points* or *hubs* specified by the user. In addition, when the transfer is between two trains, we prohibit transfers between trains running in opposite directions, thereby avoiding connections that would cause passengers to backtrack in their itinerary. Finally, transfers to trains that have no further stops after the transferring station are disallowed.

Penalty arcs are used when there is no available routing for a passenger group within the graph. These arcs are assigned a high cost equal to the time horizon H. The cost of the other activities is given by equation (1).

Figure 1 illustrates a portion of an event–activity graph showing the events and activities of a single train operating between four stations, one of which is a bifurcation, together with the corresponding passenger events and activities. In this example, each station is equipped with two station tracks, except for the bifurcation, which has none, and two section tracks connect each pair of consecutive stations. For clarity, short-turning, emergency buses, starting, ending, transferring, access, egress, and penalty activities are not shown.

5 Mathematical formulation

In this section, the mathematical formulation for the multi-objective timetable rescheduling problem is presented. The operational constraints are detailed in Section 5.1, followed by the passenger routing constraints, which are formulated in Section 5.2. The three objective functions—minimizing passenger inconvenience, op-

Name	Incoming event i : r / (e, s, p, k, tp)	Outgoing event j : r / (e', s', p', k', tp')	Section track q	c_a
Running (A_r)	$e = 0, s \in S$, $p \in P_s, k \in K^t$ $tp \in \{0, 1\}$	$e' = 1, s' \neq s \in S$, $p' \in P'_s, k' = k$ $tp' = tp$	$S_q = \{s, s'\}$, $q \in Q_p, p' \in P_q$	0
Waiting (A_w)	$e = 1, s \in S \setminus B$, $p \in P_s, k \in K^t$ $tp \in \{0, 1\}$	$e' = 0, s' = s$, $p' = p, k' = k$ $tp' = tp$	-	0
Pass-through (A_p)	$e = 1, s \in S$, $p \in P_s, k \in K^t$ $tp \in \{0, 1\}$	$e' = 0, s' = s$, $p' = p, k' = k$ $tp' = tp$	-	0
Short-turning (A_{st})	$e = 0, s \in S \setminus B$, $p \in P_s, k \in K^t$ $tp = 0$	$e' = 1, s' = s$, $p' = p, k' = k$ $tp' = 1$	-	0
Emergency bus (A_e)	$e = 0, s \in S \setminus B$, $p = \varnothing, k \in K^b$ $tp = 2$	$e' = 1, s' \in S \setminus B$, $p' = \varnothing, k' = k$ $tp' = 2$	-	$c \cdot d_{i,j}$
Starting (A_{sta})	$r \in N_b$	$e' = 1, s' = s(r)$, $p' \in P_{s'}, k' \in K^t$ $tp' = 0$	-	0
Ending (A_{end})	$e = 0, s = s(r)$ $p \in P_s, k \in K^t$ $tp \in \{0, 1\}$	$r \in N_b$	-	0

Table 2: Train activities properties

Name	Incoming event i : o / (e, s, k, tp)	Outgoing event j : d / (e', s', k', tp')	c_a
In-vehicle (A_{iv})	$e = 0, s \in S \cup B$, $k \in K, tp \in \{0, 1, 2\}$	$e' = 1, s' \neq s \in S \cup B$, $k' = k, tp' = tp$	$y_j - y_i$
Dwelling (A_d)	$e = 1, s \in S$, $k \in K^t, tp \in \{0, 1, 2\}$	$e' = 0, s' = s$, $k' = k, tp' \in \{0, 1, 2\}$	$\beta_1(y_j - y_i)$
Transfer. (A_t)	$e = 1, s \in S$, $k \in K, tp \in \{0, 1, 2\}$	$e' = 0, s' = s$, $k' \neq k \in K, tp' \in \{0, 1, 2\}$	$\beta_2 + \beta_1(y_j - y_i)$
Access (A^g_{acc})	$o \in N_o$	$e' = 1, s' = o_g$, $k \in K, tp' \in \{0, 1, 2\}$	$\beta_3 \max(0, d_g - y_i)$ $+\beta_4 \max(0, y_i - d_g)$
Egress (A^g_{egr})	$e = 0, s = d_g$ $k \in K, tp \in \{0, 1, 2\}$	$d \in N_d$	0
Penalty (A_{pen})	$o \in N_o$	$d \in N_d$	H

Table 3: Passenger activities properties

erational costs, and deviations from the undisrupted timetable—are introduced in Section 5.3. Lastly, Section 5.4 proposes a linearized version of the formulation, leading to a mixed-integer linear optimization model, and

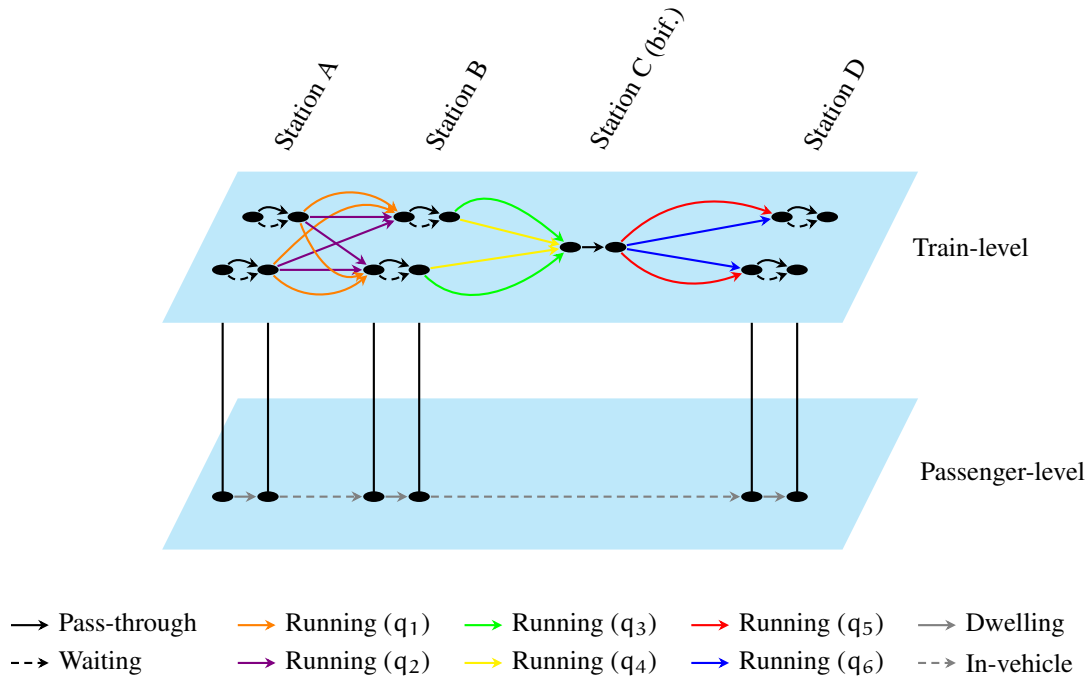

Figure 1: Simplified event-activity graph

some valid inequalities.

5.1 Operational Constraints

The operational constraints are defined based on the following primary decision variables:

- x_a: A binary variable equal to 1 if activity a is used, and 0 otherwise.
- z_i: A binary variable equal to 1 if event i is canceled, and 0 otherwise.
- y_i: A continuous variable representing the arrival or departure time at event i.

Constraints (2)–(20) ensure operational feasibility by adhering to infrastructure capacities and timing requirements:

$$\sum_{(i,j)\in A_{sta}\cup A^k} x_{(i,j)} = 1 \qquad \forall k \in K^t, \tag{2}$$

$$\sum_{a\in A_i^-} x_a = \sum_{a\in A_i^+} x_a + z_i \qquad \forall i \in N \setminus N_e, \tag{3}$$

$$z_i = 0 \qquad \forall i : s_i \notin S_b, \tag{4}$$

$$y_j \geq y_i + tt_{(i,j)} - H(1 - x_{(i,j)}) \qquad \forall (i,j) \in A_r \cup A_e, \tag{5}$$

$$y_j \geq y_i + \psi - H(1 - x_{(i,j)}) \qquad \forall (i,j) \in A_w, \tag{6}$$

$$y_j \geq y_i + \sigma - H(1 - x_{(i,j)}) \qquad \forall (i,j) \in A_{st}, \tag{7}$$

$$y_j \geq y_i - H(1 - x_{(i,j)}) \qquad \forall (i,j) \in A_p, \tag{8}$$

$$y_j \le y_i + H(1 - x_{(i,j)}) \qquad \forall (i,j) \in A_p : p_i = \varnothing, \tag{9}$$

$$0 \ge y_i - t_i \le \lambda_{k_i} \qquad \forall i \in N, \tag{10}$$

$$y_i \ge y_{j'} + \varphi - H(2 + u_{aa'} - x_a - x_{a'}) \qquad \forall a = (i,j), a' = (i',j') \in A_w \cup A_p : p_i = p_{i'} \tag{11}$$

$$y_{i'} \ge y_j + \varphi - H(3 - u_{aa'} - x_a - x_{a'}) \qquad \forall a = (i,j), a' = (i',j') \in A_w \cup A_p : p_i = p_{i'} \tag{12}$$

$$y_i \ge y_{i'} + \tau_{k_{i'}} - H(2 + u_{aa'} - x_a - x_{a'}) \qquad \forall a = (i,j), a' = (i',j') \in A_r : q_a = q_{a'}, \; s(i) = s(i'), \tag{13}$$

$$y_j \ge y_{j'} + \tau_{k_{j'}} - H(2 + u_{aa'} - x_a - x_{a'}) \qquad \forall a = (i,j), a' = (i',j') \in A_r : q_a = q_{a'}, \; s(i) = s(i'), \tag{14}$$

$$y_{i'} \ge y_i + \tau_{k_i} - H(3 - u_{aa'} - x_a - x_{a'}) \qquad \forall a = (i,j), a' = (i',j') \in A_r : q_a = q_{a'}, \; s(i) = s(i'), \tag{15}$$

$$y_{j'} \ge y_j + \tau_{k_j} - H(3 - u_{aa'} - x_a - x_{a'}) \qquad \forall a = (i,j), a' = (i',j') \in A_r : q_a = q_{a'}, \; s(i) = s(i'), \tag{16}$$

$$y_{i'} \ge y_j + \tau_{k_j} - H(2 + u_{aa'} - x_a - x_{a'}) \qquad \forall a = (i,j), a' = (i',j') \in A_r : q_a = q_{a'}, \; s(i) \ne s(i'), \tag{17}$$

$$y_i \ge y_{j'} + \tau_{k_{j'}} - H(3 - u_{aa'} - x_a - x_{a'}) \qquad \forall a = (i,j), a' = (i',j') \in A_r : q_a = q_{a'}, \; s(i) \ne s(i'), \tag{18}$$

$$x_a, z_i, u_{aa'}, u_{aa'} \in \{0,1\} \qquad \forall a, a' \in A, i \in N, \tag{19}$$

$$0 \le y_i \le H \qquad \forall i \in N, \tag{20}$$

where A^k is the set of activities of vehicle k; $tt_{(i,j)}$ is the minimum travel time on arc (i,j); $\psi, \sigma, \varphi, \tau$ are the minimum waiting, short-turning, separation, and headway times, respectively; A_i^+ and A_i^- are the sets of train activities leaving and entering node i, respectively; and $u_{aa'}$ for $a, a' \in A$ are auxiliary binary variables used to enforced 'or '-type of conditions.

Constraints (2) and (3) are flow conservation constraints, ensuring that each train departs from the shunting yard at the beginning of its trip, and that each train either continues its journey after each node or is canceled, respectively. Flow conservation constraints are not applied to emergency bus activities, as these arcs are never part of a train path. Constraints (4) prevent train cancellations at stations where no shunting yard is available.

The arrival and departure times of trains are governed by constraints (5) – (10). Specifically, constraints (5) establish time precedence for running and emergency activities. Minimum waiting and short-turning times are enforced by constraints (6) and (7), respectively. Constraints (8) ensure that the pass-through time is greater than or equal to 0 minutes. For activities at stations without station tracks (e.g., some bifurcations), constraints (9) enforce a pass-through time of 0 minutes. Constraints (10) ensure that an event i cannot be rescheduled before its planned time, nor be delayed by more than λ_{k_i} minutes. The maximum delay time varies between passenger trains (i.e., trains with a capacity $q_k > 0$) and commercial trains.

The minimum headway time between any pair of departure and arrival events occurring at the same station and the same station track is enforced by constraints (11)–(12). For each pair of arcs a and a' occurring at the same station and the same station track, but associated with different trains, either constraint (11) or (12) holds.

Section track occupation is governed by constraints (13)–(18). For each pair of arcs a and a' occurring on the same section track but associated with different trains running in the same direction, either constraints (13)–(14) or (15)–(16) must hold (see Figure 2). These constraints ensure that headway times are planned between the departure of consecutive trains and between the arrival of consecutive trains. Enforcing headway constraints at both the departure and arrival points prevents trains from overtaking each other. Similarly, for trains running in opposite directions, either constraint (17) or (18) must hold (see Figure 3).

5.2 Passenger assignment constraints

The passenger routing constraints are defined based on the following additional decision variables:

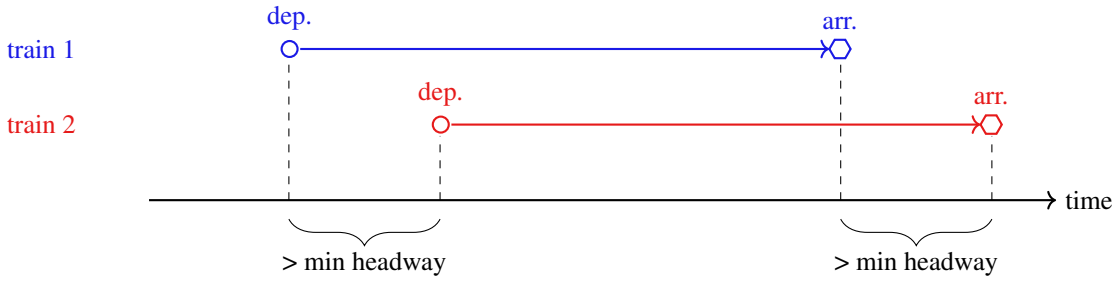

Figure 2: Section track occupation—constraints (13) – (16) for trains running in the same direction

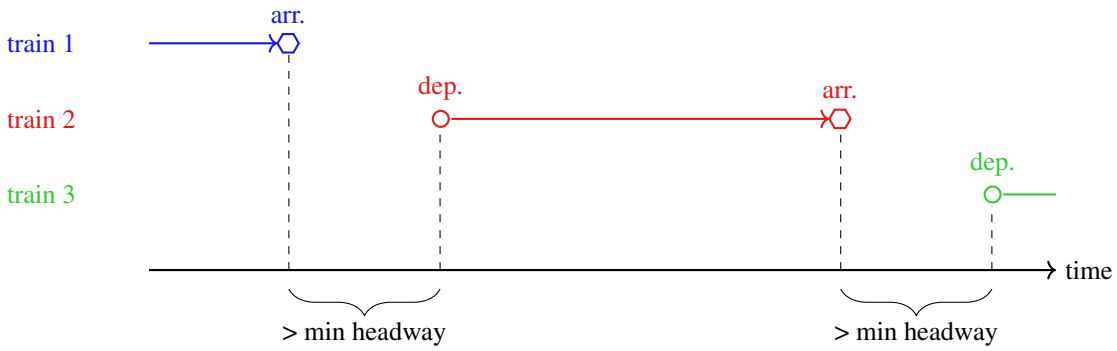

Figure 3: Section track occupation—constraints (17) and (18) for trains running in opposite directions

- w_a^g: A binary variable equal to 1 if group g uses arc a, and 0 otherwise.

Constraints (21)–(19) ensure the feasibility of passenger routing and its consistency with the chosen train disposition timetable:

$$\sum_{a \in A^g_{acc} \cup A_{pen}} w^g_a = 1 \qquad \forall g \in G, \tag{21}$$

$$\sum_{a \in A^g_{egr} \cup A_{pen}} w^g_a = 1 \qquad \forall g \in G, \tag{22}$$

$$\sum_{a \in A_i^-} w^g_a = \sum_{a \in A_i^+} w^g_a \qquad \forall g \in G, i \in N, \tag{23}$$

$$w^g_{(o_g,i)} \leq \sum_{i' \in \Theta_i} \sum_{a \in A^+_{i'} \cap A_w} x_a \qquad \forall g \in G, (o_g, i) \in A^g_{acc} \tag{24}$$

$$w^g_{(i,d_g)} \leq \sum_{i' \in \Theta_i} \sum_{a \in A^-_{i'} \cap A_w} x_a \qquad \forall g \in G, (i, d_g) \in A^g_{egr} \tag{25}$$

$$2w^g_{(i,j)} \leq \sum_{i' \in \Theta_i} \sum_{a \in A^+_{i'} \cap A_w} x_a + \sum_{j' \in \Theta_j} \sum_{a' \in A^-_{j'} \cap A_w} x_{a'} \qquad \forall g \in G, (i,j) \in A_t : tp_i, tp_j \neq 2 \tag{26}$$

$$w^g_{(i,j)} \leq \sum_{i' \in \Theta_i} \sum_{a \in A^+_{i'} \cap A_w} x_a \qquad \forall g \in G, (i,j) \in A_t : tp_j = 2 \tag{27}$$

$$w^g_{(i,j)} \leq \sum_{j' \in \Theta_j} \sum_{a' \in A^-_{j'} \cap A_w} x_{a'} \qquad \forall g \in G, (i,j) \in A_t : tp_i = 2 \qquad (28)$$

$$y_j - y_i \geq m - H(1 - w^g_{(i,j)}) \qquad \forall g \in G, (i,j) \in A_t \qquad (29)$$

$$y_j - y_i \leq M + H(1 - w^g_{(i,j)}) \qquad \forall g \in G, (i,j) \in A_t \qquad (30)$$

$$w^g_{(i,j)} \leq \sum_{(i',j') \in \Theta_{(i,j)}} x_{(i',j')} \qquad \forall g \in G, (i,j) \in A_{iv}, \qquad (31)$$

$$\sum_{g \in G} n_g w^g_{(i,j)} \leq q_{k_i} \sum_{(i',j') \in \Theta_{(i,j)}} x_{(i',j')} \qquad \forall (i,j) \in A_{iv}, \qquad (32)$$

$$y_i = y_{i'} \qquad \forall i \in A^{pass}, i' \in \Theta(i) \qquad (33)$$

$$w^g_a \in \{0,1\} \qquad \forall a \in A, g \in G, \qquad (34)$$

where Θ_i and $\Theta_{(i,j)}$ are the sets of the disaggregated events and activities associated with an aggregated event and activity, respectively; m and M are the minimum and maximum transfer times, respectively; and q_k is the passenger capacity of vehicle k.

Constraints (21) and (22) ensure that each passenger group departs from its origin and reaches its destination or is assigned to an artificial penalty arc, respectively. Constraints (23) enforce flow conservation. Constraints (24) and (25) ensure that passenger access and egress arcs are associated with trains that are waiting at a station (and not passing through). Passenger transfers between two trains are governed by constraints (26), which ensure that a transfer occurs only between trains that are waiting (and not passing through). Transfers between a train and an emergency bus, and vice versa, are governed by constraints (27) and (28), respectively, ensuring that these transfers only happen between a train and bus at a station where the train is stopping. The minimum (m) and maximum (M) transfer times are enforced by constraints (29) and (30), respectively. Constraints (31) ensure that passenger groups are routed only along paths where trains are operating. Constraints (32) ensure that the number of passengers onboard vehicle k does not exceed q_k at any time. Finally, constraint (33) ensures consistency between the two layers of the event-activity graph by aligning the arrival and departure times of passenger events with the corresponding times of the trains.

5.3 Objective functions

We formulate the timetable rescheduling problem as a multi-objective optimization problem, where the goal is to minimize the following function:

$$\min f_i, \quad i \in \{p, o, d\}, \qquad (35)$$

where $f_i, i \in \{p, o, d\}$, represent the three objective functions: the passenger inconvenience (f_p), the operational costs (f_o), and the deviation costs (f_d). These objective functions are defined as follows:

$$f_p = \sum_{g \in G} \sum_{a \in A^g} c^g_a n_g w^g_a \qquad (36)$$

$$f_o = \sum_{a \in A_e} c_a x_a \qquad (37)$$

$$f_d = \delta_1 \sum_{k \in K} \sum_{i \in N_{dis} \cap N^k} (t^k - t_i) z_i + \delta_2 \sum_{A_{rr}} d_a x_a + \delta_3 \sum_{i \in N_{dis}} (y_i - t_i) + \delta_4 \sum_{a \in A^{change}_p} x_a$$
$$+ \delta_5 \sum_{a \in A^{change}_r} x_a + \delta_6 \sum_{a \in A^{change}_w \cup A^{change}_p} x_a, \qquad (38)$$

where $\delta_1, \ldots, \delta_6$ are the weighting factors of the deviation cost f_d. These weights correspond to the costs for cancellations, reroutings, delays, stop removals, changes in section tracks, and changes in station tracks, respectively. Furthermore, d_a is the distance associated with activity a, t^k is planned arrival time of train

k at its last station, A_{rr} is the set of rerouting activities (i.e., $\{a = (i,j) \in A_r | tp(i) = tp(j) = 1\}$), and $A_p^{change}, A_r^{change}, A_w^{change}$ are the sets of pass-through, running, and waiting activities for which the station track or the section track does not correspond to the one used in the planned timetable, respectively.

To address the multi-objective nature of the problem, we employ the ϵ-constraint method. In this approach, one objective is minimized while the others are treated as constraints. The procedure is repeated for each objective until all have been considered. The order of the objectives and the ϵ values serve as hyperparameters of the model, which can be adjusted by the decision-maker to generate various Pareto efficient solutions.

5.4 Linearization and valid inequalities

The objective function f_p is non-linear because the cost terms c_a^g depend on the event times y_i and are multiplied by the binary variables w_a^g. To linearize this expression, we introduce the continuous variables v_{1a}^g, v_{2a}^g, v_{3a}^g, v_{4a}^g, v_{5a}^g, and v_{6a}^g, and impose the following constraints

$$v_{1a}^g \geq y_j - y_i - H(1 - w_a^g) \qquad \forall g \in G, a \in A_{iv} \tag{39}$$
$$v_{2a}^g \geq \beta_1(y_j - y_i) + \beta_2 - (\beta_1 H + \beta_2) \times (1 - w_a^g) \qquad \forall g \in G, a \in A_t \tag{40}$$
$$v_{3a}^g \geq \beta_1(y_j - y_i) - H\beta_1(1 - w_a^g) \qquad \forall g \in G, a \in A_d \tag{41}$$
$$v_{4a}^g \geq \beta_3 v_{5a}^g + \beta_4 v_{6a}^g - (H\beta_3 + H\beta_4)(1 - w_a^g) \qquad \forall g \in G, a \in A_{acc}^g \tag{42}$$
$$v_{5a}^g \geq d_g - y_j \qquad \forall g \in G, a \in A_{acc}^g \tag{43}$$
$$v_{6a}^g \geq y_j - d_g \qquad \forall g \in G, a \in A_{acc}^g \tag{44}$$
$$v_{1a}^g, v_{2a}^g, v_{3a}^g, v_{4a}^g, v_{5a}, v_{6a} \geq 0 \qquad \forall g \in G, a \in A_{agg}. \tag{45}$$

The linearized objective function can then be written as

$$f_p = \sum_{g \in G} n_g \left(\sum_{a \in A_{iv}} v_{1a}^g + \sum_{a \in A_t} v_{2a}^g + \sum_{a \in A_d} v_{3a}^g + \sum_{a \in A_{acc}} v_{4a}^g + \sum_{a \in A_{pen}} H w_a^g \right), \tag{46}$$

The resulting formulation, consisting of the objective function (35) and constraints (2)–(34) and (37)–(46), is a mixed-integer linear optimization program.

We can additionally impose the following valid equalities without eliminating optimal solutions

$$y_i = y_j \qquad \forall i, j \in N, e_i = e_j, s_i = s_j, k_i = k_j, tp_i = tp_j, t_i = t_j \tag{47}$$

These equalities enforce identical arrival or departure times for similar events, that is, events that are both arrivals or both departures, and that share the same station, train, event type, and scheduled time (while possibly occurring on different tracks). In cases where a train visits the same station multiple times in the planned timetable, the condition $t_i = t_j$ ensures that only events with the same scheduled time are treated as similar. For events not included in the disposition timetable, their times are non-binding because the big-M terms deactivate time constraints whenever no incoming arc is selected. Their event times can therefore be set to any value greater than or equal to their planned time.

To strengthen the linear relaxation, we propose in what follows a reformulation of constraints (11)–(18). For each activity $a = (i,j)$, define the set of similar disaggregated activities

$$\Omega_a = \left\{ (i',j') \in A_{dis} \,\middle|\, s_i = s_{i'},\ s_j = s_{j'},\ k_i = k_{i'},\ k_j = k_{j'},\ tp_i = tp_{i'},\ tp_j = tp_{j'},\ t_i = t_{i'},\ t_j = t_{j'} \right\},$$

that is, activities sharing the same origin and destination stations, the same train, the same event types, and identical scheduled times (but potentially different tracks).

In all separation and headway constraints, the terms $-x_a - x_{a'}$ can be replaced by

$$-\sum_{\bar{a} \in \Omega_a} x_{\bar{a}} - \sum_{\bar{a}' \in \Omega_{a'}} x_{\bar{a}'},$$

ensuring that separation/headway requirements apply whenever *any* activity within the corresponding similarity sets is selected.

The reformulated constraints are therefore

$$y_i \geq y_{j'} + \varphi - H(2 + u_{aa'} - \sum_{\bar{a}\in\Omega_a} x_{\bar{a}} - \sum_{\bar{a}'\in\Omega_{a'}} x_{\bar{a}'}) \qquad \forall a=(i,j), a'=(i',j') \in A_w \cup A_p : p_i = p_{i'}, \tag{48}$$

$$y_{i'} \geq y_j + \varphi - H(3 - u_{aa'} - \sum_{\bar{a}\in\Omega_a} x_{\bar{a}} - \sum_{\bar{a}'\in\Omega_{a'}} x_{\bar{a}'}) \qquad \forall a=(i,j), a'=(i',j') \in A_w \cup A_p : p_i = p_{i'}, \tag{49}$$

$$y_i \geq y_{i'} + \tau_{k_{i'}} - H(2 + u_{aa'} - \sum_{\bar{a}\in\Omega_a} x_{\bar{a}} - \sum_{\bar{a}'\in\Omega_{a'}} x_{\bar{a}'}) \qquad \forall a=(i,j), a'=(i',j') \in A_r : q_a = q_{a'}, s(i) = s(i'), \tag{50}$$

$$y_j \geq y_{j'} + \tau_{k_{j'}} - H(2 + u_{aa'} - \sum_{\bar{a}\in\Omega_a} x_{\bar{a}} - \sum_{\bar{a}'\in\Omega_{a'}} x_{\bar{a}'}) \qquad \forall a=(i,j), a'=(i',j') \in A_r : q_a = q_{a'}, s(i) = s(i'), \tag{51}$$

$$y_{i'} \geq y_i + \tau_{k_i} - H(3 - u_{aa'} - \sum_{\bar{a}\in\Omega_a} x_{\bar{a}} - \sum_{\bar{a}'\in\Omega_{a'}} x_{\bar{a}'}) \qquad \forall a=(i,j), a'=(i',j') \in A_r : q_a = q_{a'}, s(i) = s(i'), \tag{52}$$

$$y_{j'} \geq y_j + \tau_{k_j} - H(3 - u_{aa'} - \sum_{\bar{a}\in\Omega_a} x_{\bar{a}} - \sum_{\bar{a}'\in\Omega_{a'}} x_{\bar{a}'}) \qquad \forall a=(i,j), a'=(i',j') \in A_r : q_a = q_{a'}, s(i) = s(i'), \tag{53}$$

$$y_{i'} \geq y_j + \tau_{k_j} - H(2 + u_{aa'} - \sum_{\bar{a}\in\Omega_a} x_{\bar{a}} - \sum_{\bar{a}'\in\Omega_{a'}} x_{\bar{a}'}) \qquad \forall a=(i,j), a'=(i',j') \in A_r : q_a = q_{a'}, s(i) \neq s(i'), \tag{54}$$

$$y_i \geq y_{j'} + \tau_{k_{j'}} - H(3 - u_{aa'} - \sum_{\bar{a}\in\Omega_a} x_{\bar{a}} - \sum_{\bar{a}'\in\Omega_{a'}} x_{\bar{a}'}) \qquad \forall a=(i,j), a'=(i',j') \in A_r : q_a = q_{a'}, s(i) \neq s(i'). \tag{55}$$

6 Adaptive large neighborhood search heuristic

To efficiently solve formulation (2)–(10), (19)–(35), and (37)–(55), we develop a tailored ALNS algorithm. Section 6.1 presents the overall structure of the algorithm. Section 6.2 details the procedure used to compute an optimal passenger assignment for a given disposition timetable, which is a key component of our ALNS approach. Section 6.3 introduces the problem-specific operators, designed to take into account the characteristics of our model—particularly the presence of multiple track configurations and the flexibility of track changes at stations. Finally, the method to generate feasible initial solutions is detailed in Section 6.4.

6.1 General framework

Our tailored ALNS algorithm iteratively explores the solution space by modifying non–Pareto-dominated solutions while learning the performance of each operator. Let A and P be the sets of accepted and non-dominated solutions, respectively. At each iteration, a solution s from $A \cup P$ is selected, and ν trains are modified, where ν is drawn from $U(1, V)$ with $V \leq |K|$. The modification is performed either by a 3-step destroy–order–repair operator or by a combined operator.

A new solution $s+1$ is added to both P and A if it is a Pareto-efficient solution, that is, if, for every $s \in P$,

(i) $s+1$ is no worse than s in all objectives (in our case f_p, f_o, f_d), and

(ii) $s+1$ is strictly better in at least one objective.

Otherwise, it is added to A with probability

$$p_{acc} = \exp\left(-\frac{D_A(s+1)}{T}\right), \tag{56}$$

where $D_A(s)$ is a function that returns the number of solutions in A that dominate the solution s and T is the current simulated-annealing temperature, updated each time a solution is accepted as

$$T = \max(T \times \alpha, T_{min}), \tag{57}$$

with α being the cooling rate and T_{min} the minimum simulated annealing temperature.

Furthermore, each time a solution is non-Pareto-dominated, it is added to P, and the score of the operator(s) used is increased by the maximum of 0.01 and the sum of the improvements (if any) in the three objectives relative to the values of solutions in P. When a solution is added to A but not to P, the operator score is increased by 0.01.

At each iteration, if the size of A exceeds $|A|_{max}$, it is pruned. Pareto-dominated solutions are randomly removed until the limit is reached; if necessary, non-Pareto-dominated solutions are also removed.

The algorithm stops when the time limit is reached and returns the set P; otherwise, it proceeds to the next iteration. Figure 4 summarizes our ALNS framework.

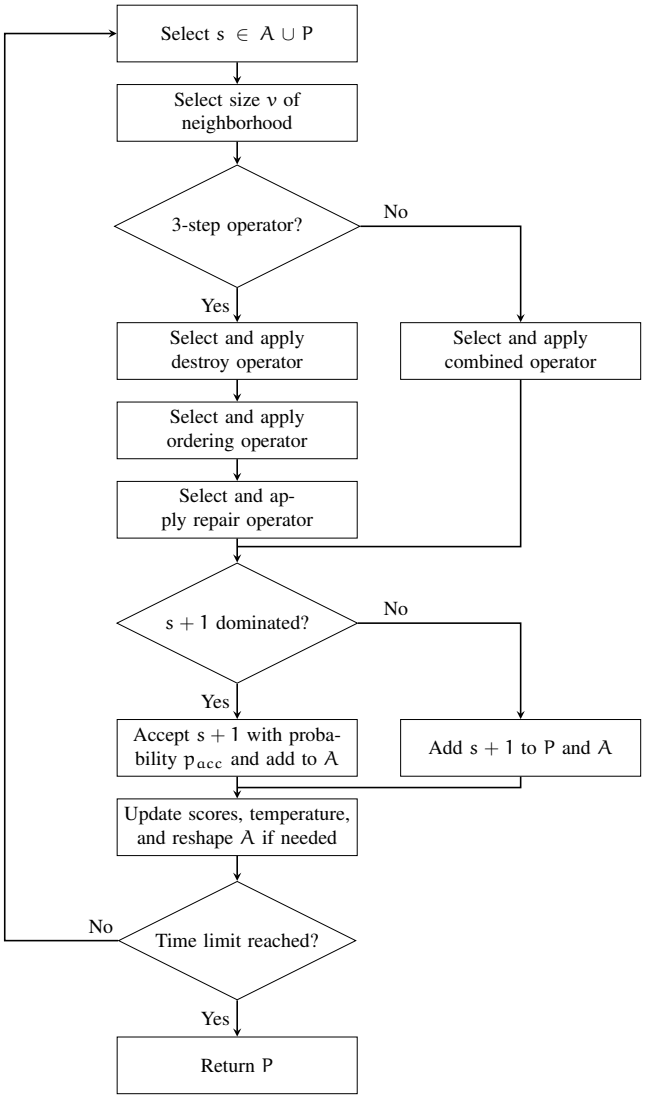

Figure 4: ALNS framework

6.2 Passenger assignment algorithm

To evaluate the passenger inconvenience cost f_p during the solution process, passenger assignment is performed using the procedure presented in Binder et al. (2017*a*). This assignment is done dynamically, such that, for each disposition timetable, the passenger groups are rerouted considering constraints (21)–(34).

The passenger assignment, detailed in Algorithm 1, takes as input an exogenous priority list Γ, which orders passenger groups according to their boarding priority. Passenger groups with higher priority are assigned a feasible path in the network first. Whenever an activity reaches its capacity, it is removed from consideration for all subsequent, lower-priority groups. As a result, high-priority groups are typically assigned their shortest path (with respect to the generalized travel time costs), while lower-priority groups may be routed along a more expensive path if their shortest path becomes saturated.

Algorithm 1: Passenger assignment

Input: A_{agg} : set of arcs; G: set of passenger groups;
Γ: priority list of passengers;
Output: c_g: path cost of group g;
x_a^g: binary variables, equal 1 if group g uses arc a

Order G according to Γ;
$c, x \leftarrow$ `ShortestPath`(G, A_{agg});
$u_a \leftarrow 0 \quad \forall a \in A_{agg}$;
while $G \neq \emptyset$ **do**
  $g \leftarrow$ `first(`G`)`;
  for $a = (i, j) \in A_{agg}$ **do**
    $\hat{u}_a \leftarrow u_a + n_g * x_a^g$;
  end
  if `any(`$\hat{u}_a > q_{k_i}$`)` **then**
    $A_{agg} \leftarrow A \setminus \{a \in A_{agg} : \hat{u}_a > q_{k_i}\}$;
    $G' \leftarrow \{g \in G : x_a^g = 1\}$;
    $c_g, x_a^g \leftarrow$ `ShortestPath`(G', A_{agg}) for $g \in G, a \in A_{agg}$;
  else
    $u \leftarrow \hat{u}$;
    $G \leftarrow G \setminus \{g\}$;
  end
end

6.3 Neighborhood operators

6.3.1 Destroy operators

In the proposed ALNS framework, destroy operators are used to partially dismantle the current timetable by removing a subset of trains, thereby creating room for the reconstruction phase to explore alternative scheduling decisions. The following operators are implemented:

1. *Random removal.* Selects ν trains randomly from the set K.
2. *Canceled-train removal.* Selects ν trains that are already partially or completely canceled.
3. *Most delayed removal.* Trains are ranked based on their accumulated delay, which is defined as the sum of the differences between the actual event times y_i and the scheduled times t_i for each event i in the timetable of train k. A subset of ν trains is then selected, with the selection probability proportional to their relative accumulated delays.

4. *Crowded removal.* Ranks trains by their passenger usage (i.e., total boardings) and selects ν trains with probability proportional to their relative usage. This favors the removal of highly loaded trains.

5. *Low-occupancy removal.* Similar to the previous operator, but trains are selected with probability inversely proportional to their relative usage, favoring the removal of sparsely loaded trains.

6. *Section-track removal.* Selects a random section track, identifies all trains using this track, and removes ν random trains from this subset.

7. *Station-based removal.* Randomly selects a station and removes ν random trains passing through this station.

8. *Consecutive station-based removal.* Similar to the previous operator, but removes ν consecutive trains. After selecting a train k that passes through a given station, the next $\nu - 1$ trains in the original timetable that also pass through this station are removed, or fewer if fewer than $\nu - 1$ such trains remain.

9. *Disruption-area removal.* Removes ν random trains that pass through the disrupted area—whether a station or a section track—at any time. The disrupted area includes all station tracks at a disrupted station and all section tracks between disrupted stations, even if these tracks remain available.

10. *Disruption-period removal.* Removes ν random trains that pass through the disrupted area during the disruption period.

11. *Extended-disruption removal.* Removes ν random trains that pass through the disrupted area during the disruption period extended by 10 minutes.

12. *Consecutive extended-disruption removal.* From the candidate set of trains passing through the disrupted area during the disruption period plus 10 minutes, one train k is randomly selected. The operator then removes the next $\nu - 1$ trains that are scheduled consecutively after k in the original timetable and that also belong to the candidate set, or fewer if fewer than $\nu - 1$ such trains remain.

6.3.2 Insertion ordering

The insertion ordering operators determine the sequence in which trains removed from the timetable by a destroy operator are reinserted. This ordering is then used by the repair operator to sequentially reschedule the trains, following the specified order.

1. *Random order.* The removed trains are randomly shuffled.

2. *Keep scheduled order.* The removed trains that pass through the disrupted area are first sorted by their scheduled entry time into the disrupted area, in ascending chronological order. The remaining trains are randomly shuffled and placed after those passing through the disrupted area in the resulting sequence.

3. *Apply regret-2 insertion.* This method first identifies all removed trains operating within the disrupted area. These trains are separated into two lists according to their direction of travel and sorted in ascending order of their scheduled entry times into the disrupted area. At each iteration, the algorithm considers the next candidate in each direction–the train with the earliest entry time in the disrupted area. For each candidate, it computes the *regret* of not rescheduling that train next, defined as the additional delay incurred if it is rescheduled after, rather than before, the train in the opposite direction. To compute the regret, we assume that trains are allowed to switch tracks and consider the occupation of the disrupted area tracks. Only the delay at the exit of the disrupted area is taken into account. The train with the highest regret is selected next and removed from its list. Information on track occupation is then updated. This procedure continues until one of the lists becomes empty. The final ordering consists of trains sorted by their regret, followed by the remaining trains in random order.

4. *Prioritize most crowded.* The trains are ranked in descending order of passenger usage (i.e., total boardings), so that the most crowded trains are rescheduled first.

5. *Group trains two-by-two.* This operator partitions the destroyed trains into three sets: trains running in one direction within the disrupted area, trains running in the opposite direction within the disrupted area, and trains that do not operate in the disrupted area. For each of the two directional sets, trains are paired in ascending order of their scheduled entry times into the disrupted area. If a direction contains an odd number of trains, one train remains unpaired, chosen at random between the first and the last. All resulting pairs and any unpaired trains are then ordered according to the earliest entry time into the disrupted area among their members. The trains from the third set are placed after these, in a random order.

6.3.3 Repair operators

Reinsert trains in the insertion order. After each repair operator, a correction step is applied: for each train k, if any event along its path incurs a delay greater than λ_k minutes, the train is canceled at the first such event; otherwise, it remains unchanged.

1. *Cancel partially.* This operator cancels each selected train at the first conflict it encounters—specifically, the first time the train is scheduled to occupy a station or section track already used by another train, where the track is unavailable due to headway and separation time constraints, or if the track is disrupted.

2. *Cancel completely.* This operators canceled completely trains at their starting activity.

3. *Delay.* This operator reschedules trains to the earliest available slots, following the order defined by the insertion ordering operator. Each train is rescheduled as close as possible to its initial timetable, while keeping the same station and section tracks. The operator calls the delay function (see Algorithm 2) with `track_change = 2`.

4. *Delay with track changes.* This operator behaves as the previous one but allows changes to station and section tracks when such modifications can help reduce the accumulated delays. It calls the delay function (see Algorithm 2) with `track_change = 0`.

5. *Delay with track changes in the disrupted area.* This operator behaves as the previous one but restricts track changes to the disrupted area only. It calls the delay function (see Algorithm 2) with `track_change = 1`.

6. *Add stops.* This operator converts pass-through activities from the initial timetable into waiting activities. For each reinserted train k, the number of added stops is randomly sampled from $U(1, \theta^k)$, where θ^k denotes the number of eligible scheduled pass-through activities that can be converted into waiting activities. After these modifications, the updated timetable is passed to the delay function (see Algorithm 2) with `track_change = 1` to restore feasibility.

7. *Remove stops.* This operator performs the inverse operation of the previous one, converting waiting activities into pass-through activities.

8. *Short-turn.* This operator short-turns the trains to be rescheduled. As each train has only one predefined short-turning option, no selection step is required. The delay function (see Algorithm 2) is then called for each short-turned train with `track_change = 0` to restore feasibility.

At the core of most repair operators is the delay procedure outlined in Algorithm 2, which reconstructs a feasible timetable by sequentially rescheduling the activities of a list of trains. The feasibility of the resulting timetable depends on the assumption that the timetable for trains not included in the list remains feasible, as their schedules are not modified. For each train, the procedure evaluates for each of its activities the feasible time slots on the set of admissible section or station tracks. This evaluation is handled by the `get_timeslot` function (Step 14), which determines the earliest available slot on a candidate track based on the current occupation table. The start time of an activity is set to the maximum among the scheduled start time, the completion time of the preceding activity, and the earliest unoccupied time on the considered track. The resulting end time must

respect several constraints: it must not precede the scheduled end time, must account for the minimum running time of the activity, must satisfy minimum headway separations when the previous occupation corresponds to a train running in the same direction, and, when backtracking is activated, must remain compatible with the start time of the next activity in the chain.

For each candidate track, the algorithm stores the corresponding earliest start and end times. In Step 16, it retains only the tracks associated with the minimal end times; these represent the most promising options. Step 17 further filters this set by keeping only those tracks that also achieve minimal start times, yielding a subset Q_{start}. The operator then applies a tie-breaking rule (Step 18): when the originally planned track is still available in Q_{start} and remains fully connected to the upcoming activities in the timetable, it is selected; otherwise, a track is chosen at random among the tracks in Q_{start}. A track is said to be *fully connected* when a train can reach any subsequent tracks from that track.

Feasibility with respect to temporal consistency is ensured by a controlled backtracking mechanism in Steps 20 to 25. If the earliest feasible start time of the current activity exceeds the latest permissible end time of the previous one, the algorithm backtracks by one or two activities, depending on whether the preceding activity corresponds to a bifurcation without station tracks or not. Conversely, when the activity is feasible, the operator updates the decision variables x_a^{s+1}, y_a^{s+1} and the track-occupation table (Step 28). This iterative repair logic enables the construction of a feasible timetable $s+1$ (assuming the trains $k \in K \setminus K^s$ have a feasible timetable) that respects operational rules while allowing limited structural deviations such as track changes within the policy defined by `track_change`.

6.3.4 Combined operators

Instead of calling a three-step operator—where a destroy, ordering, and repair operator are applied successively—the ALNS algorithm can invoke a combined operator, namely either the add emergency bus operator or the add emergency bus and short-turn operator. These operators are detailed next.

1. *Add an emergency bus.* This operator randomly selects one emergency bus arc and schedules it at its planned time.

2. *Add an emergency bus and short-turn.* This operator adds an emergency bus as in the previous operator and, in addition, short-turns one train at the destination station of that bus. The train is randomly selected among those that have a short-turning activity departing from the destination station of the emergency bus and whose scheduled departure time falls within the allowable transfer time window after the emergency bus arrives. After setting the train to the short-turning activities, the delay function (see Algorithm 2) with `track_change = 0` is called to restore feasibility.

6.4 Initial solutions

The ALNS algorithm begins by generating several feasible disposition timetables as initial solutions. The following steps are taken to create 5 initial solutions:

- Apply the *cancel partially* operator to all trains in a random order. This method only partially cancels trains passing through the disrupted area during the disruption period.

- Destroy all trains passing through the disrupted section track(s) using the *section-track removal* operator, order them according to the *keep scheduled order* insertion ordering, and apply the *delay* repair operator.

- Destroy all trains passing through the disrupted section track(s) using the *section-track removal* operator, order them according to the *keep scheduled order* insertion ordering, and apply the *delay with track changes* repair operator.

- Destroy all trains passing through the disrupted area using the *extended-disruption removal* operator, order them according to the *keep scheduled order* insertion ordering, and apply the *delay with track changes* repair operator.

Algorithm 2: Delay function

input : Ordered list K^s of trains to reschedule from destroyed solution s;
Event–activity graph $G(N, A)$;
For each train k, ordered list of activities A^k in the original timetable;
Decision variables x_a^s, $\forall a \in A_{dis}$, and y_i^s, $\forall i \in N$ for solution s;
Track occupancy table tracks_occu with reserved intervals;
Track-change policy track_change $\in \{0, 1, 2\}$: all, only in disrupted area, or none.

output: Decision variables x_a^{s+1}, y_i^{s+1} corresponding to the repaired timetable $s+1$.

$idx \leftarrow 0$, backtrack $\leftarrow$ **False**
$x_a^{s+1} \leftarrow x_a^s$, $\forall a \in A_{dis}$, $y_i^{s+1} \leftarrow y_i^s$, $\forall i \in N$
for $k \in K^s$ **do**
  while $idx \leq |A^k|$ **do**
    $a = (i, j) \leftarrow A^k[idx]$
    if (track_change $= 1$ and $a \notin$ disrupted area) or track_change $= 2$ **then**
      alt_tracks $\leftarrow \{q_a/p_a\}$ // No alternative section track allowed
    else
      alt_tracks $\leftarrow$ get_alternative_tracks(a)
    end
    min_end_time $\leftarrow \begin{cases} \text{next_activity_start_time}, & \text{if backtrack} = \textbf{True} \\ 0, & \text{otherwise} \end{cases}$
    Initialize min_start_times, min_end_times, max_end_times as empty dictionaries
    for $q/p \in$ alt_tracks **do**
      min_start_times[q], min_end_times[q], max_end_times[q] $\leftarrow$
       get_timeslot$(k,\ q/p,$ min_start_time, min_end_time, tracks_occu$)$
    end
    Q_end $\leftarrow \{q \mid$ min_end_times[q] minimal$\}$,
    Q_start $\leftarrow \{q \in$ Q_end $\mid$ min_start_times[q] minimal$\}$
    $q^\star \leftarrow \begin{cases} \text{planned_track}, & \text{if planned_track} \in \text{Q_start and planned_track fully connected} \\ \text{random(Q_start)}, & \text{otherwise} \end{cases}$
    if min_start_times[$q^\star$] $>$ last_max_end_time **then**
      backtrack $\leftarrow$ **True**
      if $A^k[i-1] \in A_p$ and $p_{A^k[i-1]} =$ None **then**
        $idx \leftarrow idx - 2$ // Previous activity at station without tracks
      else
        $idx \leftarrow idx - 1$
      end
    else
      backtrack $\leftarrow$ **False**, $idx \leftarrow idx + 1$
      update$(x_a^{s+1},\ y_i^{s+1},$ tracks_occu$)$
      min_start_time $\leftarrow$ min_end_times[$q^\star$], last_max_end_time $\leftarrow$ max_end_times[$q^\star$]
    end
  end
end

- Destroy all trains passing through the disrupted area using the *disruption-period removal* operator, order them according to the *apply regret-2 insertion* ordering, and apply the *delay with track changes* repair operator.

7 Computational experiments

Throughout the experiments, we use the following parameter settings. The minimum waiting time is set to $\psi = 30$ seconds, the minimum short-turning time to $\sigma = 6$ minutes, the minimum separation time to $\varphi = 2$ minutes, and the minimum headway time to $\tau_k = 2$ minutes for passenger trains and 3 minutes for freight trains. Transfer times range between $m = 0$ and $M = 30$ minutes. The maximum delay is limited to $\lambda_k = 30$ minutes for passenger trains and 60 minutes for freight trains. The weight factors are $\delta_1 = 50$, $\delta_2 = 1.5$, $\delta_3 = 1$, $\delta_4 = 10$, $\delta_5 = 50$, and $\delta_6 = 25$.

The simulated annealing procedure uses an initial temperature $T = 1.00$, a cooling rate $\alpha = 0.99$, and a minimum temperature $T_{min} = 10^{-4}$. Operating an emergency bus costs 600 CHF per kilometer, and each emergency bus has a capacity of 100 passengers. The weight parameters of Equation (1) are reported in Table 4.

Parameter	Value	Unit	Reference
β_1	2.5	min/min	Wardman (2004)
β_2	10	min/transfer	De Keizer et al. (2012)
β_3	0.5	min/min	Small (1982)
β_4	1	min/min	Small (1982)

Table 4: Values of the weighting factors of the utility of a path

All experiments are conducted on a Linux machine with 72 Intel(R) Xeon(R) Platinum 8360Y processors running at 2.40 GHz and 503 GB of RAM. Gurobi 11.0.0 is used as an exact benchmark to solve formulation (2)–(10), (19)–(35), and (37)–(55).

Next, we present the case study, a subset of the Swiss railway network between Nyon and Lausanne. Section 7.2 evaluates the performance of the ALNS heuristic on small instances. Section 7.3 presents the results for two disruption scenarios and discusses the professional relevance of the resulting disposition timetables.

7.1 Case study

We analyze the corridor between Nyon and Lausanne in Switzerland (see Figure 5), which features 25 stations, including bifurcations. The corridor includes segments with up to 6 parallel section tracks between pairs of stations and up to 10 station tracks per station. We focus on the morning peak period, from 07:00 AM to 09:00 AM, during which 80 trains (comprising both passenger and freight trains) pass. The passenger trains have capacities ranging from 675 to 1162 seats. Among these, several regional trains operate a round-trip service between Lausanne and Allaman.

Two disruption scenarios are considered: (i) a partial track disruption where one of the two section tracks between Gilly-Bursinel and Rolle is unavailable from 07:00 AM to 08:00 AM, and (ii) a complete track disruption where both section tracks between Gilly-Bursinel and Rolle are unavailable during the same period. In the second scenario, the bifurcation at Gilly-Bursinel and Rolle stations is fully disconnected, and the only way to move passengers between these stations during the disruption is by deploying emergency buses. These buses can operate between Gland and Rolle, which correspond to the nearest stations with a shunting yard before and after the disrupted area.

The passenger groups are randomly generated from the *Modèle National de Transport de Personnes*, developed by the *Office fédéral du développement territorial* in Switzerland. This model is designed to predict national-level traffic demand for various modes of transportation, accounting for the socio-economic characteristics of the population, demographic data, and congestion. The Swiss territory is divided into 8,000 traffic zones, and the output represents the average traffic for public transport, private car, bike, and walking between each pair of traffic zones on a regular open day, using projections for 2050 socio-economic data and the projected rail infrastructure for 2035.

To translate this model into station-to-station demand, we use the following procedure: First, we link the public transport demand for each traffic zone to its nearest train station. For demand assigned to stations outside

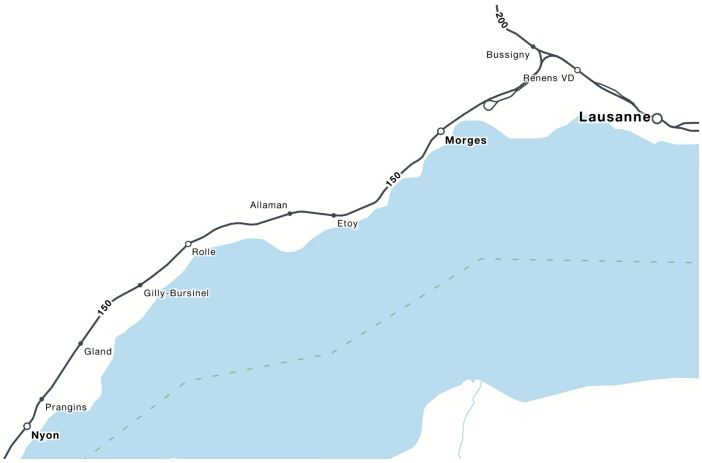

Figure 5: Nyon to Lausanne train line

of the studied zone, we transpose the corresponding demand to the station specified in Table 5. The overall approach is to assign traffic from the southwest of Nyon (i.e., Genève and Genève aéroport) to Nyon, from areas to the northwest of Lausanne (i.e., Biel/Bienne, Neuchâtel, Yverdon-les-Bains) to Busigny, and from other regions (e.g., Bern, Zürich, the alpine region, and Ticino) to Lausanne. Only major train stations are considered in the transposition, as the traffic from smaller stations is expected to be negligible.

Assigned to	Origin/destination station
Bussigny	Biel/Bienne, Neuchâtel, Yverdon-les-bains
Lausanne	Arth-Goldau, Baden, Basel, Bellinzona, Bern, Brig, Brugg, Bülach, Burgdorf, Chur, Dietikon, Effretikon, Frauenfeld, Fribourg/Freiburg, Landquart, Liestal, Lugano, Luzern, Montreux, Olten, Pfäffikon, Rapperswil, Rotkreuz, Schaffhausen, Schwerzenbach, Sion, Solothurn, St. Gallen, Stettbach, Thalwil, Thun, Uster, Vevey, Visp, Wädenswil, Wallisellen, Weinfelden, Wetzikon ZH, Winterthur, Zug, Zürich Altstetten, Zürich Enge, Zürich Flughafen, Zürich Hardbrücke, Zürich HB, Zürich Oerlikon, Zürich Stadelhofen
Nyon	Genève, Genève aéroport

Table 5: Re-assignment of demand for areas outside of the studied zone

Following this procedure, we can calculate a travel probability for each origin-destination pair of stations within the study area. This probability corresponds to the average daily traffic for each pair (including transposed demand) divided by the total traffic demand. The resulting probabilities are displayed in Figure 6. For instance, we observe that the most frequent trips are between Renens and Lausanne, the two largest cities in the region.

7.2 Heuristic performance

The performance of the proposed ALNS is evaluated on a reduced-size instance. The instance includes the stations between Nyon and Allaman (namely Nyon, Prangins, Gland, Gilly-Bursinel, Rolle, Perroy, and Allaman) during the morning peak period from 8:00 AM to 9:00 AM, under a 30-minute partial blockage from 8:00 AM to 8:30 AM. The blockage occurs between the same pair of stations as in the full-scale instance, namely between Gilly-Bursinel and Rolle.

The solutions obtained with the ALNS heuristic are compared to those of the MILP formulation solved with Gurobi 11.0.0 using different values of ϵ, where $\epsilon \in \{0, 0.01, 0.05, 0.1, 0.5\}$. For the exact approach, a

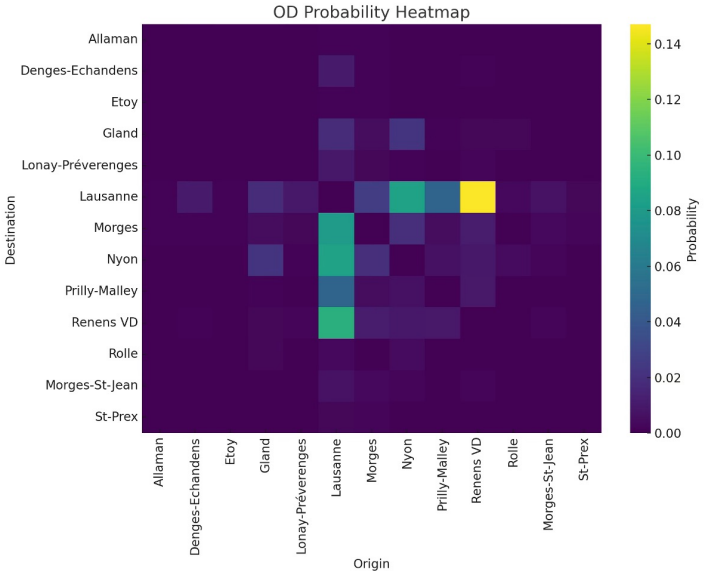

Figure 6: Origin-destination heatmap

time limit of 7200 seconds is imposed and up to 72 CPUs are made available. Despite this large computational budget, several instances cannot be solved to proven optimality, and the average optimality gap remains non-negligible. In particular, all instances converge for the f_d objective, whereas the average optimality gap for the f_p objective is 4.6%. This observation confirms that relying solely on exact methods is impractical in real-time disruption-management settings. The corresponding solutions, shown as red triangles in Figure 7, define the reference Pareto front used to evaluate the heuristic performance of our heuristic. Note that only the first two objectives (f_d and f_p) are reported, as none of the solutions for this instance involves emergency buses; consequently, $f_o = 0$ for all solutions.

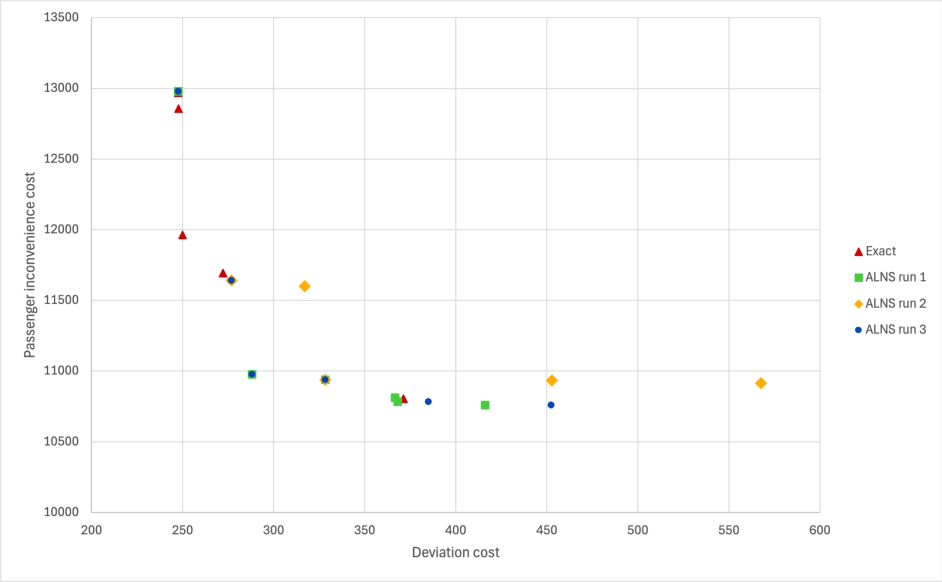

Figure 7: Heuristic performance

The ALNS heuristic is run with a time limit of 300 seconds on a single CPU. Three independent runs are performed to assess the robustness of the approach. The results are reported in Figure 7. Each run produces a set of non-dominated solutions that closely follow the trade-off curve identified by the exact method. While the three runs exhibit consistent behavior, they explore slightly different regions of the Pareto frontier.

Overall, these results indicate that our ALNS is well suited for disruption-management contexts. It identifies high-quality trade-offs between passenger inconvenience and timetable deviation while requiring substantially less computational effort than the exact approach.

7.3 Results

The ALNS algorithm is applied to the full Nyon–Lausanne instance over a planning horizon covering the morning peak period from 07:00 AM to 09:00 AM, with a time limit of 1,000 seconds for each disruption scenario. Section 7.3.1 addresses the disruption scenario involving a partial blockage, where one of the two section tracks between Gilly-Bursinel and Rolle is unavailable from 07:00 to 08:00. Section 7.3.2 considers the disruption scenario corresponding to a complete blockage, in which both section tracks between Gilly-Bursinel and Rolle are unavailable during the same time interval.

7.3.1 Partial blockage

The trade-off between passenger inconvenience cost and timetable deviation cost for three independent runs is reported in Figure 8. The obtained solutions exhibit passenger inconvenience costs ranging approximately from 18,000 to 31,000, and deviation costs between about 2,100 and 13,000. This set of solutions therefore provides a diverse pool of disposition timetables reflecting different compromises between passenger inconvenience and operational deviation.

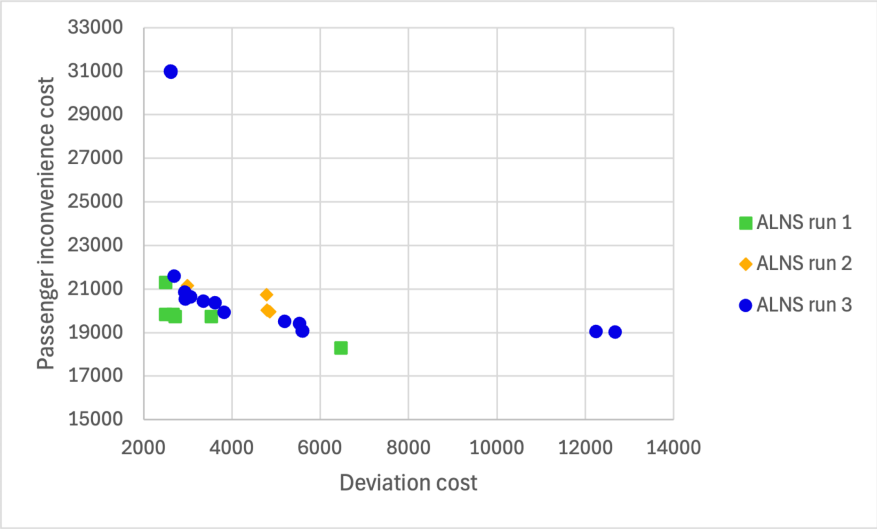

Figure 8: Partial blockage from 07:00 AM to 09:00 AM – Pareto front

An example solution from the resulting Pareto front is illustrated in Figure 9 using the *Graphical Timetables* tool in Viriato. The horizontal axis represents space, showing train movements between stations, while the vertical axis represents time. The red rectangle indicates the disrupted area. Passenger train movements are depicted using different colors: red for InterCity services, green for RegioExpress services, and black for regional trains (RER Vaud), while freight trains are displayed in blue. Dotted lines indicate that a train is using a different section track than originally planned between two stations.

In this solution, six section track changes are observed, all of which occur within the disrupted area. In addition, several trains are grouped in pairs or triplets, with trains traveling in the same direction entering the

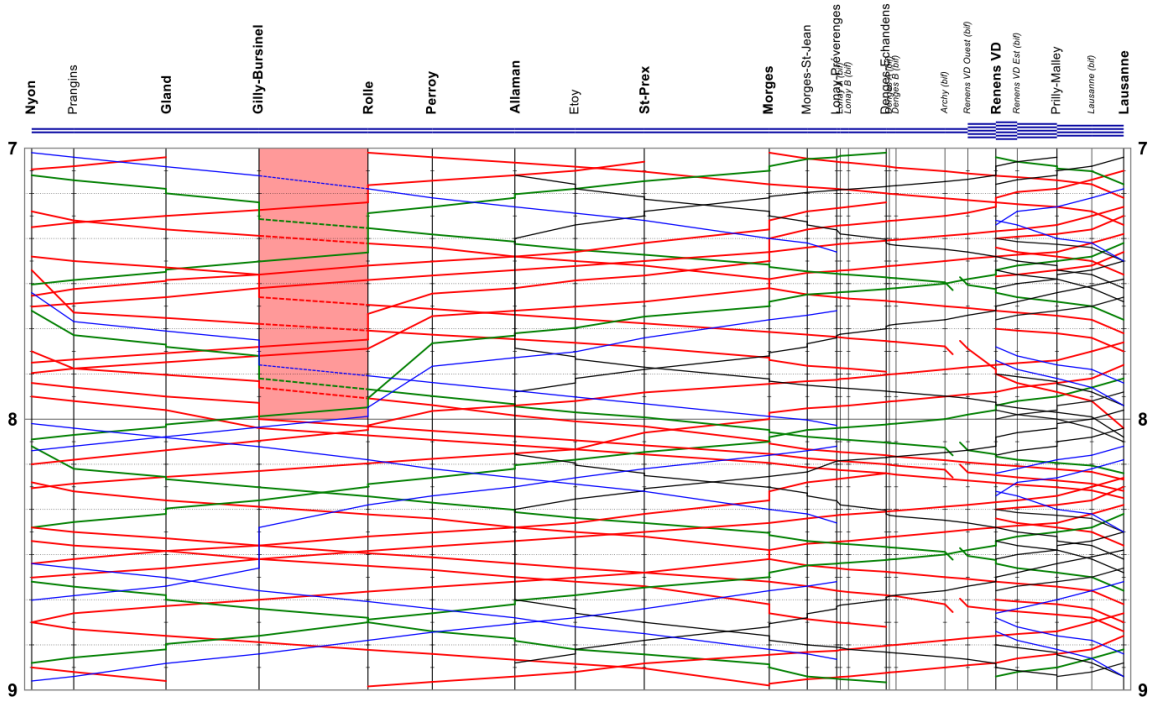

Figure 9: Partial blockage from 07:00 AM to 09:00 AM – Example of a solution

disrupted area consecutively. This behavior arises because the headway required between trains running in the same direction is smaller than that required between trains running in opposite directions. The algorithm therefore exploits this structure to sequence trains efficiently through the disrupted area, limiting the propagation of delays. This behavior is consistent with standard operational recovery practices. Finally, no trains traveling in the same direction cross each other in the timetable, that is, no train paths intersect between stations, which is a necessary condition for feasibility.

7.3.2 Complete blockage

When considering a complete blockage of one hour, deploying emergency buses becomes advantageous from the passengers' perspective. Emergency buses require more travel time than trains between consecutive stations, and passengers must typically make one or two transfers, first from train to bus and then from bus to train. As a result, itineraries involving emergency buses are associated with relatively high passenger inconvenience costs. Consequently, the use of emergency buses is only beneficial when the alternative consists of waiting for a substantial amount of time.

Figure 10 shows the solutions obtained from a single run of the ALNS. The passenger inconvenience cost is reported on the y-axis, the deviation cost on the x-axis, and the color of the points indicates the number of emergency buses used.

Solutions using one or two emergency buses exhibit higher deviation costs, as emergency buses are typically deployed in combination with short-turning a train at the arrival station of the bus. This allows passengers to continue their journey after reaching the terminus of the emergency bus. In contrast, solutions without emergency buses lead to higher passenger inconvenience costs, ranging from approximately 35,000 to 36,000. When two emergency buses are deployed, the passenger inconvenience cost can be reduced to slightly below 22,000.

An example solution from the resulting Pareto front is illustrated in Figure 11, again using the *Graphical Timetables* tool in Viriato. Since the blockage is complete, meaning that no train can pass between Gilly-

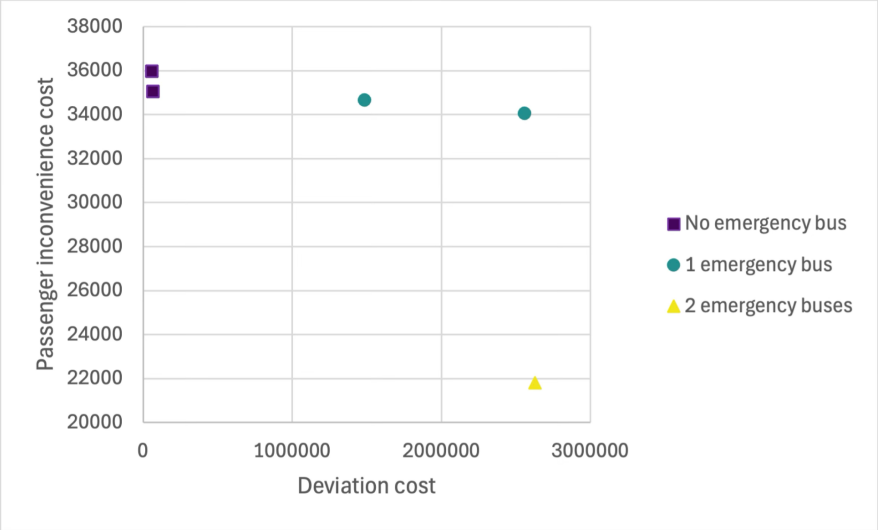

Figure 10: Complete blockage from 07:00 AM to 09:00 AM – Pareto front

Busignel and Rolle between 07:00 AM and 08:00 AM, all trains must either be canceled, short-turned, or delayed. In this example, no train is short-turned. Notably, the first trains to traverse the disrupted sections after the end of the blockage are passenger trains, while freight trains are postponed. This ordering is likely to contribute to a lower passenger inconvenience cost.

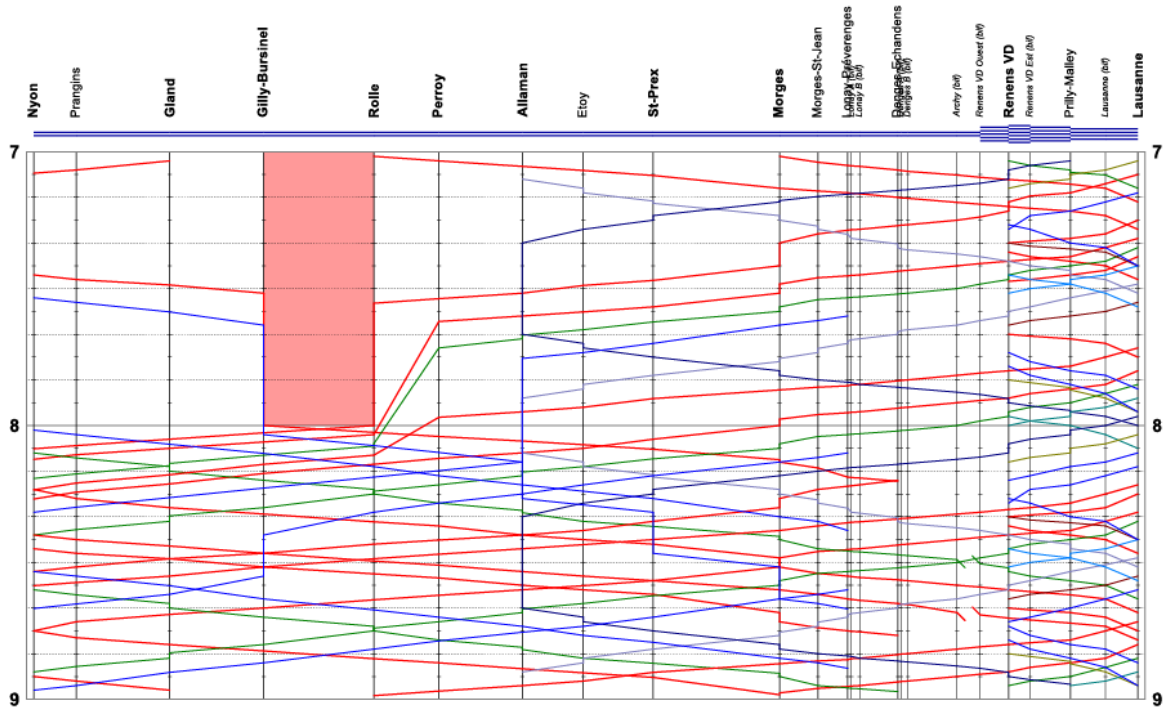

Figure 11: Complete blockage from 07:00 AM to 09:00 AM – Example of a solution

8 Conclusions

This report addressed the problem of railway timetable rescheduling under disruptions while explicitly accounting for passenger assignment. The problem was modeled using an event–activity graph, allowing operational decisions and passenger routing to be represented within a unified framework. A MILP formulation was proposed, capturing operational constraints, passenger flow conservation, and multiple objective components reflecting both operational performance and passenger inconvenience. To ensure tractability, suitable linearizations and valid inequalities were introduced.

Given the computational difficulty of the resulting formulation, especially for realistic railway networks and disruption scenarios, an ALNS heuristic was developed. The heuristic exploits the structure of the problem through tailored destroy, ordering, and repair operators that jointly modify timetable decisions and passenger assignments. Computational experiments on a real-world case study demonstrated that the proposed heuristic is able to produce high-quality solutions within short computation times. Compared to the exact approach, which requires substantial computational resources and time, the ALNS provides a practical alternative that remains suitable for real-time or near-real-time disruption management.

A Notation for the train rescheduling problem

Symbol	Description
Parameters	
n_g	size of passenger group g
c	cost per kilometer for operating an emergency bus
c_a	cost of arc a
q_k	passenger capacity of a train k
$\psi, \sigma, \varphi, \tau_k$	minimum waiting, short-turning, separation, and headway of train k times
H	time horizon
$\delta_1, \ldots, \delta_7$	weights of the cost function f_d
d_a	distance of activity a
t^k	planned arrival time of train k at its last station
t_i	planned arrival or departure time at event node $i \in N$
Sets	
K^t, K^b	set of trains and buses
K	set of vehicles ($K = K^t \cup K^b$)
Q	set of section tracks
B	set of bifurcations
G	set of passenger groups
S	set of stations
P_s	set of section tracks at station s
S_b	set of stations with a shunting yard
N	set of events
N_{dis}	set of disaggregated events ($N_{dis} \subset N$)
N_{agg}	set of aggregated events ($N_{agg} \subset N$)
N^k	set of events for train $k \in K$ ($N^k \subset N$)
N_o, N_d	set of passengers origin and destination events
N_e	set of emergency events, i.e., $\{i \in N \mid tp(i) = 2\}$
N_b	set of shunting events
A	set of activities
A^{pass}	set of passenger activities
A_{rr}	set of train rerouting activities, i.e., $\{a = (i,j) \in A_r \mid tp(i) = tp(j) = 1\}$
$A_\alpha^{\text{change}}, \alpha \in \{r, w, p\}$	sets of running, waiting, and pass-through activities on non-planned tracks
A_i^+, A_i^-	set of train activities leaving and entering node $i \in N \cup N_b$
A_i^{g+}, A_i^{g-}	set of passenger group g activities leaving and entering node $i \in N \cup N_o$
N_s	set of events associated with station $s \in S$
Variables	
x_a	binary variable indicating if arc a is used
w_a^g	binary variable indicating if user group g uses arc a
y_i	continuous variable of the arrival or departure time at event node $i \in N$
z_i	binary variable indicating if event i is canceled
$q_{aa'}$	binary variable used to enforce 'or'-type conditions
Functions	
Θ_i	outputs the disaggregated events linked with the aggregated event i
$\Theta_{(i,j)}$	outputs the set of disaggregated activities linked with the aggregated activity (i,j)

B Code and data

The full implementation of the proposed methodology is openly available in a public GitHub repository: `https://github.com/transp-or/Robust_Railway.git`. The repository contains the source code, documentation, and reproducibility material required to replicate the results presented in this paper. To properly run the code, access to the Viriato software and a Viriato database is required.

References

Altazin, E., Dauzère-Pérès, S., Ramond, F. & Tréfond, S. (2020), 'A multi-objective optimization-simulation approach for real time rescheduling in dense railway systems', *European Journal of Operational Research* **286**(2), 662–672.

Binder, S., Maknoon, Y. & Bierlaire, M. (2017*a*), 'Exogenous priority rules for the capacitated passenger assignment problem', *Transportation Research Part B: Methodological* **105**, 19–42.

Binder, S., Maknoon, Y. & Bierlaire, M. (2017*b*), 'The multi-objective railway timetable rescheduling problem', *Transportation Research Part C: Emerging Technologies* **78**, 78–94.

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

De Keizer, B., Geurs, K. T. & Haarsman, G. (2012), Interchanges in timetable design of railways: A closer look at customer resistance to interchange between trains, *in* '40th European Transport Conference, ETC 2012', Association for European Transport (AET).

Hong, X., Meng, L., D'Ariano, A., Veelenturf, L. P., Long, S. & Corman, F. (2021), 'Integrated optimization of capacitated train rescheduling and passenger reassignment under disruptions', *Transportation Research Part C: Emerging Technologies* **125**, 103025.

Robenek, T., Maknoon, Y., Azadeh, S. S., Chen, J. & Bierlaire, M. (2016), 'Passenger centric train timetabling problem', *Transportation Research Part B: Methodological* **89**, 107–126.

Sharma, B., Pellegrini, P., Rodriguez, J. & Chaudhary, N. (2023), 'A review of passenger-oriented railway rescheduling approaches', *European Transport Research Review* **15**(1), 14.

Small, K. A. (1982), 'The scheduling of consumer activities: work trips', *The American Economic Review* **72**(3), 467–479.

Veelenturf, L. P., Kidd, M. P., Cacchiani, V., Kroon, L. G. & Toth, P. (2016), 'A railway timetable rescheduling approach for handling large-scale disruptions', *Transportation Science* **50**(3), 841–862.

Wardman, M. (2004), 'Public transport values of time', *Transport Policy* **11**(4), 363–377.

Xiu, C., Pan, J., D'Ariano, A., Zhan, S. & Peng, Q. (2024), 'Passenger service-oriented timetable rescheduling for large-scale disruptions in a railway network: A heuristic-based alternating direction method of multipliers', *Omega* **125**, 103040.

Zhan, S., Wong, S., Shang, P., Peng, Q., Xie, J. & Lo, S. (2021), 'Integrated railway timetable rescheduling and dynamic passenger routing during a complete blockage', *Transportation Research Part B: Methodological* **143**, 86–123.

Zhang, C., Gao, Y., Cacchiani, V., Yang, L. & Gao, Z. (2023), 'Train rescheduling for large-scale disruptions in a large-scale railway network', *Transportation Research Part B: Methodological* **174**, 102786.

Zhu, Y. & Goverde, R. M. (2019), 'Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions', *Transportation Research Part B: Methodological* **123**, 149–181.

Zhu, Y. & Goverde, R. M. (2020), 'Integrated timetable rescheduling and passenger reassignment during railway disruptions', *Transportation Research Part B: Methodological* **140**, 282–314.