Passenger-oriented railway disposition timetables in case of severe disruptions

Stefan Binder

Yousef Maknoon

Michel Bierlaire

Ecole Polytechnique Fédérale de Lausanne

April 2015

STRC | **15th Swiss Transport Research Conference**

Monte Verità / Ascona, April 15 – 17, 2015

Ecole Polytechnique Fédérale de Lausanne

Passenger-oriented railway disposition timetables in case of severe disruptions

Stefan Binder, Yousef Maknoon, Michel Bierlaire
Transport and Mobility Laboratory
Ecole Polytechnique Fédérale de Lausanne
Station 18, 1015 – Lausanne
phone: +41 21 693 93 29
fax: +41-21-693 80 60
{s.binder,yousef.maknoon,michel.bierlaire}@epfl.ch

April 2015

Abstract

Delays are one of the major reasons for passenger dissatisfaction in the railway industry. Depending on the gravity of the delay, timetables, crew schedules or rolling stock may be affected. In this research, we address the issue of timetable recovery in case of severe disruptions. Once an initial delay has occurred, the original timetable needs to be updated to a so-called disposition timetable. This new timetable has to be conflict-free in terms of operational constraints (e.g., trains cannot use the same track section at the same time) and as convenient as possible for the passengers. The recent scientific literature on recovery models mainly focuses on the operational point of view, thus paying less attention to the impact of passenger dissatisfaction in case of disruptions. This observation is the motivation for introducing a hybrid methodology that takes the satisfaction of both parties (i.e., passengers and railway companies) into account. Passengers' travel choices are represented by means of a passenger assignment model that uses a path disutility function encompassing travel time, departure time shift and number of connections for every possible path between origin and destination. An adaptive large neighbourhood search meta-heuristic is implemented in order to generate operationally feasible disposition timetables. Its objective is to minimize the overall passenger disutility as well as operational costs. The flexibility of our framework allows to evaluate several recovery strategies (e.g., partial train cancellation, complete train cancellation, train addition, train replacement, capacity addition). This model will assist train operating companies when evaluating the trade-off between economic and infrastructural feasibility of recovery schemes on the one hand side and passenger satisfaction on the other.

Keywords

train rescheduling problem, demand-driven, integer linear program, meta-heuristic

1 Introduction

Delays are one of the major reasons for passenger dissatisfaction in the railway industry. Depending on the gravity of the delay, timetables, crew schedules or rolling stock may be affected. In this research, we address the issue of timetable recovery in case of a severe disruption (e.g., the unavailability of a track between two stations for several hours). Once an initial delay has occurred, the original timetable needs to be updated to a so-called disposition timetable. This new timetable has to be conflict-free in terms of operational constraints (i.e., no two trains can be scheduled on the same resource at the same time) and as convenient as possible for the passengers. The current practice in the field still heavily relies on predetermined scenarios and personal experience of train traffic controllers. However, due to the high utilization rate of modern railway networks, a decision made at one location in the network can have domino effects in the whole network. Furthermore, as railway companies in heavily disrupted situations are mainly concerned with operational questions in order to provide a feasible disposition timetable, passenger considerations are pushed into the background.

The problem of finding a conflict-free disposition timetable has received a large attention in the recent scientific literature, especially in case of minor disturbances in railway networks. On the other hand, the literature on timetable adjustment during large disruptions is very limited (see Section 2 for an overview of the current state-of-the-art in train timetable rescheduling). The literature focusing on minimizing the negative effects of delays for passengers has also been developing but remains sparse. These observations are the motivation for introducing a hybrid methodology that takes into account both parties' satisfaction when generating a disposition timetable for highly disrupted situations. In our formulation, the problem is addressed by a realistic simulation-optimization model that integrates both operational rescheduling and passenger routing.

The main contribution of our approach is to introduce a demand-driven framework that takes the passenger perspective into account when generating a disposition timetable. Hence, several recovery strategies that are relevant in case of major disruptions in railway networks (such as train cancellations, partial train cancellations, train re-routings or additional train or bus services) can be assessed not only from the operational point of view, but also from the customer side. A key asset of the model is the flexibility of the choice-based passenger routing. We introduce a linear disutility function for every passenger path from origin to destination (encompassing travel time, waiting time and penalties for transfers and early/late departures) and assume that passengers minimize this generalized travel time. Passengers can therefore easily be re-routed through another part of the network or update their departure time in case of a disruption. Finally, our model deals with a railway network, instead of a single railway line, as the existing timetable evaluation methods do. It therefore allows to assess the effects of recovery strategies on a network level.

The remainder of this paper is structured as follows. Section 2 reviews the current state of research in the train timetable rescheduling area. Section 3 then presents the passenger-oriented train timetable rescheduling problem in detail. The problem is formally defined as an integer linear program in Section 3.1. The number of passengers this exact formulation can handle is very limited, thus a heuristic solution approach is presented in Section 4. Based on the western part of the Swiss railway network, a case study is presented as an illustrative example in Section 5. Finally, Section 6 concludes the paper and provides directions for future research.

2 Literature review

The literature review presented in this section focuses on recent contributions to the train timetable rescheduling (TTR) problem. It is based on the thorough review paper on railway recovery models by Cacchiani *et al.* (2014). Publications are classified according to three criteria that facilitate the identification of gaps where contributions can be made to the TTR literature, hence justifying the relevance of our work.

Disturbance / Disruption In a railway network, a disturbance is a primary delay (i.e., a process that takes longer than initially scheduled) — or a set of primary delays — that causes secondary delays that can be handled by rescheduling the timetable only, without rescheduling the resource duties (such as crews and rolling stock). On the other hand, a disruption is a (relatively) large external incident strongly influencing the timetable and requiring resource duties to be rescheduled as well.

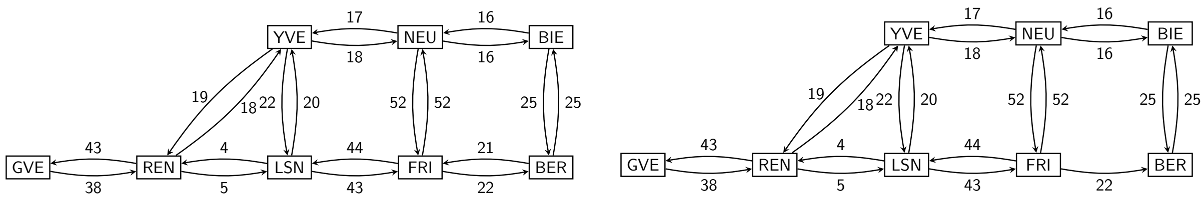
Microscopic / Macroscopic representation In a microscopic approach, the railway infrastructure is modelled very precisely, sometimes at the switch or track section level, in order to compute detailed running times and headways between trains. In a macroscopic approach, the infrastructure is considered at a higher level where stations and tracks are represented by nodes and arcs in a graph, respectively. Details such as signals or track sections are ignored.

Operations-centric / Passenger-centric model Operations-centric models focus on minimizing parameters related to the train company operations, such as delays or the number of cancelled trains, whereas passenger-centric models focus on minimizing the negative effects of disruptions and disturbances for passengers.

The thorough review of railway recovery models presented in Cacchiani *et al.* (2014) shows that the major part of the recent scientific literature deals with disturbances rather than disruptions. Further, in most papers, the railway network is represented at the microscopic rather than at the macroscopic level. Most papers also have an operations-centric approach to railway timetable rescheduling, instead of a passenger-centric view. However, in reality, passenger reactions play a significant role in recovery policies. This is why the present work focuses on the generation of passenger-oriented timetables in case of severe disruptions and proposes to assess the impacts of recovery strategies on a network level. The literature reviewed in this section thus focuses on the works that are most relevant to our case, i.e., that deal with disruptions at a macroscopic level (for a complete overview of the TTR literature, refer to Cacchiani *et al.* (2014), and references therein). First, operations-centric models are presented. Then, passenger-centric works are reviewed and the differences with the present work are pointed out.

Narayanaswami and Rangaraj (2013) develop a MILP model that detects and resolves conflicts because of a disruption that blocks part of a single track railway line. Disrupted train movements are rescheduled in both directions of the line for a small instance, with the objective of minimizing the total delay of all trains. The model does not allow trains to be cancelled. Albrecht *et al.* (2013) consider the problem of disruptions due to track maintenance, arising when maintenance operations take longer than scheduled and thus force to cancel additional trains. A disposition timetable including track maintenance is constructed using a problem space search meta-heuristic. This heuristic is also used to generate quickly disposition timetables in case of a disrupted system. Louwerse and Huisman (2014) consider the case of partial and complete blockades in case of a major disruption. They develop a mixed-integer programming model to generate the disposition timetable. Two disruption measures are applied: train cancelling and train delaying. Schedule regularity constraints (e.g., operating approximately the same number of trains in each direction during a partial blockade) are included in the formulation in order to take the rolling stock problem into account implicitly. In case of a complete blockade, both sides of the disruption are considered independently (i.e., trains will reverse before the disrupted area but no coordination with the other side is considered).

Kanai *et al.* (2011) develop a model that simulates train traffic and passenger flows simultaneously and define several functions to measure passengers' disutility. A tabu search algorithm is used to decide if passenger connections are to be maintained or not. The model only considers a single railway line (i.e., not a network) and evaluates the effects of small disturbances. Kunimatsu *et al.* (2012) also define disutility functions in order to evaluate different timetables from the passenger perspective. The disutility function encompasses travel times, waiting times and penalties for transfers and congestion. Again, only a single railway line is considered by the model. Cadarso *et al.* (2013) consider an integrated timetabling and rolling stock problem that accounts for passenger demand by splitting it up into two steps. In the first step, anticipated



(a) Sample network.

(b) Disrupted sample network.

Figure 1: A sample railway network, inspired from the Western part of the SBB network. Nodes stand for stations and directed edges for tracks between stations. The numbers on the edges are the running times between the stations.

disrupted demand is computed using a multinomial logit model. As demand figures are estimated before the timetable is adjusted, they are based on line frequencies in an anticipated disposition timetable, rather than on actual arrival and departure times. In the second step, the timetabling and rolling stock rescheduling problem is formulated and solved as a MILP model, subject to the anticipated demand calculated in the first step. This formulation is the one coming closest to the present work, where the effects of disruptions on the passenger demand is explicitly dealt with. However, the main difference between the two approaches is that, in Cadarso *et al.* (2013), the passenger demand is evaluated on an expected timetable before solving the timetable and rolling stock rescheduling problem, whereas in our case, the passenger demand is assigned on an actual timetable (with determined departure and arrival times for every train) in order to evaluate passenger satisfaction. Furthermore, the interaction between demand and supplied capacity is ignored in Cadarso *et al.* (2013).

3 Problem description

Consider a railway network with stations and uni-directional tracks between the stations. In our formulation, we model this network by a graph where nodes represent the stations and directed edges stand for the tracks between the stations. During regular operations, the normal timetable is run on the full graph. We then assume that an edge (or several edges) in the graph becomes unavailable (e.g., because of a fallen tree obstructing the track). The problem we are addressing is to generate a disposition timetable for the disrupted graph that is as convenient as possible for the passengers, while maintaining operational feasibility of the timetable. Figures 1(a) and 1(b) illustrate an undisrupted and a disrupted sample graph.

According to Louwerse and Huisman (2014), the railway disruption management process consists of three consecutive phases with different utilization rates of the network, defined as the number of trains running in the network at a given time. The first phase is the transition phase from the regular timetable to the disposition timetable. The utilization of the network decreases during this phase as trains scheduled on the unavailable edge need to be cancelled or rerouted. The disposition timetable is then operated during the second phase and the utilization stabilizes at a lower level than during regular operations. Finally, the third phase is the transition phase from the disposition timetable back to the regular timetable. In this paper, we focus on the second phase of the rescheduling problem. As input to our model, we consider the locations of all trains according to the regular timetable. More precisely, we assume that the trains are located in the station closest to their position when the disruption happens during regular operations. The output of the model is a disposition timetable, denoting which trains are operated and the associated departure and arrival times at stations.

A train is defined as an ordered sequence of (non-repeated) stations that are visited with a given arrival and departure time. Note that all sequences of stations can be different from one train to the other and that the concept of train lines is therefore not used in our formulation. In order to determine the first station of the trains, the notion of train depots at stations is introduced. The number of trains available in a depot is an upper bound on the number of trains that can depart from this station during the planning horizon. Two types of trains are in the depots: trains that are located at the station at the beginning of the disruption and additional trains that can be scheduled if needed in the disposition timetable. Furthermore, we assume that all trains have equal characteristics, including a given passenger capacity.

On the demand side, passengers are considered in a disaggregate manner. It is assumed that each passenger has an origin and a destination station, as well as a desired departure time from origin.[1] We assume that passengers have perfect knowledge of the system and choose their path between origin and destination by minimizing their generalized travel time (made of travel time, waiting time, transfer time and penalties for transfers and early/late departure). An iterative framework is introduced in order to take train capacities into account (refer to Binder *et al.* (2014) for details on the capacitated passenger assignment tool). It is assumed that passengers use their capacitated shortest path in terms of generalized travel time and that they do not reconsider their path during their journey. Further, we assume that passenger demand will not change due to the disruption. However, we include a “taxi” option for disrupted passengers. That is, passengers whose generalized travel time is more than one hour longer than the shortest possible travel time between origin and destination leave the system.

[1] Note that the latest desired departure time from origin of a passenger with a desired arrival time at destination can be computed by traversing the time-expanded network in reverse topological order. Thus, only passengers with desired departure time from origin are considered in our case.

3.1 Mathematical formulation

In this section we define the problem mathematically as an integer linear program. To that end, a time-expanded network (see Ahuja *et al.* (1993) and, e.g., Nguyen *et al.* (2001), Hamdouch and Lawphongpanich (2008) for applications) is used.

A time-expanded network is a directed graph $G(V, E)$. In our formulation, the set of nodes V consists of time-expanded nodes, train depot nodes as well as passenger origin and destination nodes. A time-expanded node $a_s^t \in N$ (respectively $d_s^t \in N$) represents an arrival (respectively departure) event, at station $s \in S$ at time $t \in T$, where S is the set of stations and T the set of time steps. Train depot nodes $r \in R$ as well as origin and destination nodes ($o_p \in O$ and $d_p \in D$) are time-invariant nodes. The former describe the number of trains available at the beginning of the rescheduling problem in every depot, while the latter represent the origin and destination of passenger $p \in P$. In the following, train nodes will be indexed by $i \in V = N \cup R$ and passenger p's nodes by $i \in V_p = N \cup \{o_p, d_p\}$.

The arcs in arc set E represent activities between the nodes and can be classified into two categories: train arcs $(i, j) \in A \subset V \times V$ and passenger p's arcs $(i, j) \in A_p \subset V_p \times V_p$. Table 1 lists the different types of arcs, the nodes they connect as well as their interpretation.

Table 1: List of arcs in time-expanded network.

Arc type	Start node type	End node type	Interpretation
Train departure arc	r	d_s^t	Departure of a train from station s at time t
Train driving arc	d_s^t	$a_{s'}^{t'}$	Train driving between stations s and s' from time t to t'
Train dwelling arc	a_s^t	$d_s^{t'}$	Train dwelling at station s from time t to t'
Access arc	o_p	d_s^t	Passenger p entering first train at station s at time t
Egress arc	a_s^t	d_p	Passenger p exiting last train at station s at time t
Riding arc	d_s^t	$a_{s'}^{t'}$	Passenger riding a train from station s to s' from time t to t'
Waiting arc	a_s^t	$d_s^{t'}$	Passenger waiting in a train at station s from time t to t'
Transfer arc	a_s^t	$d_s^{t'}$	Passenger transferring from one train to another at station s, from time t to t'

The passenger-oriented timetable rescheduling problem is modeled as an integer linear program by adapting the minimum cost flow problem. To that end, the following binary decision variables are defined:

$$x_{(i,j)} = \begin{cases} 1 & \text{if a train runs on arc } (i,j) \in A \\ 0 & \text{otherwise} \end{cases} \qquad w^p_{(i,j)} = \begin{cases} 1 & \text{if passenger } p \text{ uses arc } (i,j) \in A_p \\ 0 & \text{otherwise} \end{cases}$$

The objective function is a linear combination of operational costs and passenger disutility:

$$\min \sum_{(i,j)\in A | i \in R} c_t \cdot x_{(i,j)} + \sum_{p \in P} \sum_{(i,j)\in A_p} c^p_{(i,j)} \cdot w^p_{(i,j)},$$

where c_t is the cost of starting a train and $c^p_{(i,j)}$ is the cost of arc $(i, j) \in A_p$ for passenger p. The passenger cost of the arc corresponds to its value in the generalized travel time function. The constraints of the problem are given by Eqs. (1)–(10). Constraints (1) ensure that, for every depot r, not more than $n_r \in \mathbb{N}$ trains start from depot r. Flow conservation constraints are imposed on every node $i \in V$ by constraints (2). Constraints (3) forbid the usage of arcs that are disrupted, i.e. $(i, j) \in A_D$. Constraints (4), (5) and (6) ensure that every passenger p leaves origin, reaches destination and uses the shortest path in terms of the generalized travel cost. Finally, constraints (7) make sure that a passenger only uses an arc where a train runs and the capacity constraints of every train are expressed by Eqs. (8).

$$\sum_{j \in N} x_{(r,j)} \le n_r \qquad \forall r \in R \tag{1}$$

$$\sum_{i \in V} x_{(i,k)} = \sum_{j \in V} x_{(k,j)} \qquad \forall k \in V \tag{2}$$

$$x_{(i,j)} = 0 \qquad \forall (i,j) \in A_D \tag{3}$$

$$\sum_{(i,j)\in A_p | i = o_p} w^p_{(i,j)} = 1 \qquad \forall p \in P \tag{4}$$

$$\sum_{(i,j)\in A_p | j = d_p} w^p_{(i,j)} = 1 \qquad \forall p \in P \tag{5}$$

$$\sum_{i \in V_p} w^p_{(i,k)} = \sum_{j \in V_p} w^p_{(k,j)} \qquad \forall k \in V_p, \forall p \in P \tag{6}$$

$$w^p_{(i,j)} \le x_{(i,j)} \qquad \forall p \in P, \forall (i,j) \in A \cap A_p \tag{7}$$

$$\sum_{p \in P} w^p_{(i,j)} \le cap_{(i,j)} \cdot x_{(i,j)} \qquad \forall (i,j) \in A \cap A_p \tag{8}$$

$$x_{(i,j)} \in \{0,1\} \qquad \forall (i,j) \in A \tag{9}$$

$$w^p_{(i,j)} \in \{0,1\} \qquad \forall (i,j) \in A_p, \forall p \in P \tag{10}$$

4 Solution approach

The integer linear program presented in Section 3.1 has been validated on a sample network. The number of passengers it can handle is however very limited and we therefore introduce a heuristic solution approach to solve instances with a realistic number of passengers.

An algorithm that removes trains and inserts new ones in the timetable is presented. The adaptive large neighbourhood search (ALNS) meta-heuristic, based on destroy and repair operators that allow large changes to the solution, is used. It appears to be an ideal framework to solve our problem, as destroy operators would remove trains from the timetable, while repair operators would add new ones. The ALNS framework has first been introduced in Ropke and Pisinger (2006), where it is used to solve instances of the Pickup and Delivery Problem with Time Windows. It has also been used to solve the single-line timetabling problem in Barrena *et al.* (2014). One of the major advantages of this framework is that the search is allowed very large moves that can potentially re-arrange up to 50% of all trains in a given iteration. The price of doing these large moves is that the computational time to perform and evaluate the move increases considerably. Hence, only a fraction of the moves that could be evaluated by a standard heuristic per time unit can be performed in our case. However, computational experiments in Ropke and Pisinger (2006) showed that ALNS generally performs better than standard heuristics.

4.1 General framework

The general pseudo-code for a minimizing ALNS framework is shown in Algorithm 1. The algorithm assumes the existence of an initial solution. In our case, the regular timetable, where trains scheduled to run on the disrupted track are cancelled, is used as an initial solution. At the beginning of each iteration of the algorithm, one Removal operator and one Insertion operator are chosen (Line 4), based on past performance of the operators (see Section 4.2 for details). Then, the operators are applied on the current solution s' (Line 5) and the solution is evaluated by the capacitated passenger assignment procedure described in Binder *et al.* (2014). The operational cost of the solution is given by the sum of running times of all trains, while the passenger cost is given by the sum of generalized travel times of all passengers. The total cost of the solution s, $z(s)$, is the sum of operational and passenger cost. The current solution s' is accepted if the total cost decreases (Line 8) or with probability $e^{-(z(s')-z(s))/T}$, where $T > 0$ is the current temperature, if the total cost increases (Line 16). This acceptance criterion that allows worsening solutions with a given probability comes from simulated annealing. The motivation for this probabilistic acceptance criterion is the fact that simple descent heuristics that only accept solutions which are better than the current solution have a tendency to get trapped in local minima. Based on the quality of the accepted solution, the scores of the chosen operators are updated (Lines 11, 13 or 17, see Section 4.3 for details). The algorithm is split into segments of length L_s. Whenever the iteration count hits a multiple of L_s, the weights of all operators are updated and their scores reset (Line 19). Finally, the temperature T is decreased according to Bierlaire (2015), p. 738. The algorithm terminates when the stopping criterion is met and returns the best solution found s^*.

input : Initial solution s, Initial (final) temperature T_0 (T_f)

1 $T \leftarrow T_0$, $s^* \leftarrow s$
2 **while** $T > T_f$ **do**
3     $s' \leftarrow s$
4   Choose Removal and Insertion operator
5   Apply the operators to s'
6   Assign passengers on s'
7   **if** $z(s') < z(s)$ **then**
8         $s \leftarrow s'$
9   **if** $z(s) < z(s^*)$ **then**
10             $s^* \leftarrow s$
11   Update score of chosen operators with σ_1
12   **else**
13   Update score of chosen operators with σ_2
14   **else**
15   **if** *s' is accepted by simulated annealing criterion* **then**
16             $s \leftarrow s'$
17   Update score of chosen operators with σ_3
18   **if** *Iteration count is multiple of L_s* **then**
19   Update weights of all operators and reset scores
20   Update T
21 **return** s^*

Algorithm 1: Adaptive large neighbourhood search meta-heuristic.

4.2 Choice of the operators

At every iteration of Algorithm 1, one Removal operator and one Insertion operator are selected. To that end, weights are assigned to the different operators and the roulette wheel selection principle is applied. That is, for N operators with weights $w_i, i \in \{1, \ldots, N\}$, operator i is chosen with probability $w_i / \sum_{j=1}^{N} w_j$. This procedure is applied once for the Removal operators and once for the Insertion operators, thus the selection of one type of operator is independent of the other type of operator.

4.3 Adaptive weight adjustment

The operator weights introduced in Section 4.2 are updated using information from earlier iterations. The idea is to keep track of a score π_i for every operator i. This score measures the recent performance of the given operator; a higher score corresponds to a better performance. As explained in Section 4.1, the ALNS algorithm is divided into segments of iterations. The score of all operators is set to zero at the beginning of every segment. Then, at every iteration, the score of the selected operators is updated.[2] We distinguish four cases: If the last remove-insert operation of the algorithm

- resulted in a new global best solution, the operators are rewarded with a score increase $\sigma_1 > 0$.
- resulted in a solution with a lower total cost than the current solution, the score of the selected operators is increased by $\sigma_2 < \sigma_1$.
- resulted in a solution with a higher total cost than the current solution, but the solution was accepted by the simulated annealing criterion, the scores are increased by $\sigma_3 < \sigma_2$.
- resulted in a solution that was not accepted, the scores of the selected operators are not increased.

At the beginning of the algorithm, all weights are set to one. Then, at the end of every segment, the weights are updated using the recorded scores in the following manner. After segment j, the weight of operator i in segment $j + 1$, $w_{i,j+1}$ is computed as follows:

$$w_{i,j+1} = (1 - \eta)w_{i,j} + \eta\frac{\pi_i}{n_i},$$

where η is the reaction factor that controls how quickly the weight adjustment responds to changes in the effectiveness of the operators and n_i is the number of times operator i was used in the previous segment (if $n_i = 0$, we assume that the weight of operator i remains unchanged in segment $j + 1$).

[2]Note that the scores of both the Removal and the Insertion operators are increased by the same amount, as it is not possible to tell which one led to the improvement.

4.4 Operators

Given a solution, Removal and Insertion operators delete and insert trains in the timetable. For every Insertion operator, constraints on headways, minimum dwell and run times are considered in order to provide an operationally feasible timetable. The operators that were used in the present work are described in Sections 4.4.1 and 4.4.2. At every iteration of Algorithm 1, the selected Removal operator is used ρ_r times, while the selected Insertion operator is used ρ_i times. ρ_r and ρ_i follow semi-triangular distributions and are integers randomly drawn from $[1, n]$, respectively $[1, n_{\max} - n]$, where n is the number of trains in the current solution and $n_{\max}$ the maximal number of trains in the system.

4.4.1 Removal operators

R1 — Remove trains randomly This operator selects ρ_r trains and removes them from the timetable. If ρ_r is small, it allows for minor adjustments to the timetable, while it yields a major transformation of the solution if ρ_r is large.

R2 — Remove trains with lowest demand This operator selects the train with the lowest demand (i.e., passenger-minutes) and removes it from the timetable. It is applied ρ_r times.

4.4.2 Insertion operators

I1 — Insert trains randomly This operator inserts a random train into the timetable. The initial station is selected by drawing from all possible stations. Then, the train can continue along one outgoing track from this initial station, considering headway and minimum running times on the track. At every subsequent station, the outgoing track is chosen randomly until a station is visited twice or there is no feasible departure time. This procedure is repeated ρ_i times.

I2 — Insert trains after highest demand train This operator inserts ρ_i trains in the timetable after the train with the highest demand (i.e., passenger-minutes). The new trains follow exactly the same route as the highest demand train and need to verify operational constraints.

5 An illustrative example

In this section, the methodology described in Section 4 is applied on a network with eight train stations. The illustrative example is described below, and results are reported thereafter.

5.1 Case study description

The infrastructure network of the illustrative example is inspired from the Western part of the Swiss federal rail network. It consists of eight stations: Geneva (GVE), Renens (REN), Lausanne (LSN), Fribourg (FRI), Bern (BER), Yverdon (YVE), Neuchâtel (NEU) and Biel (BIE). The available tracks as well as the minimal travel times between stations are shown in Fig.1.

We consider the morning rush hour on this network, between 5:00am and 9:00am, and a disruption between stations BER and FRI between 7:00am and 9:00am. Thus, the infrastructure network is given by Fig. 1(a) between 5:00am and 7:00am and by Fig. 1(b) between 7:00am and 9:00am. The regular timetable was downloaded from `http://www.sbb.ch`. During regular operations, 207 trains are running on this network.

Passenger origin-destination demand was generated based on a report by the Swiss National Railways (2013). The desired departure time of each passenger is generated using a Poisson process with average inter-arrival time $\lambda = 10,000/60$ passengers per hour. The origin-destination pair is drawn from a uniform distribution between all possible origin-destination pairs. A total of 40,466 passengers are generated.

5.2 Computational results

Five different scenarios were defined in order to compare the performance of the ALNS metaheuristic:

Regular All trains scheduled between 5:00am and 9:00 run on the infrastructure network given by Fig. 1(a).

Disrupted The trains scheduled to run on the disrupted track between 7:00am and 9:00am are cancelled. All other trains remain unchanged.

R1-I1 The trains scheduled to run on the disrupted track between 7:00am and 9:00am are cancelled. In addition, Algorithm 1 is run with two operators: R1 and I1.

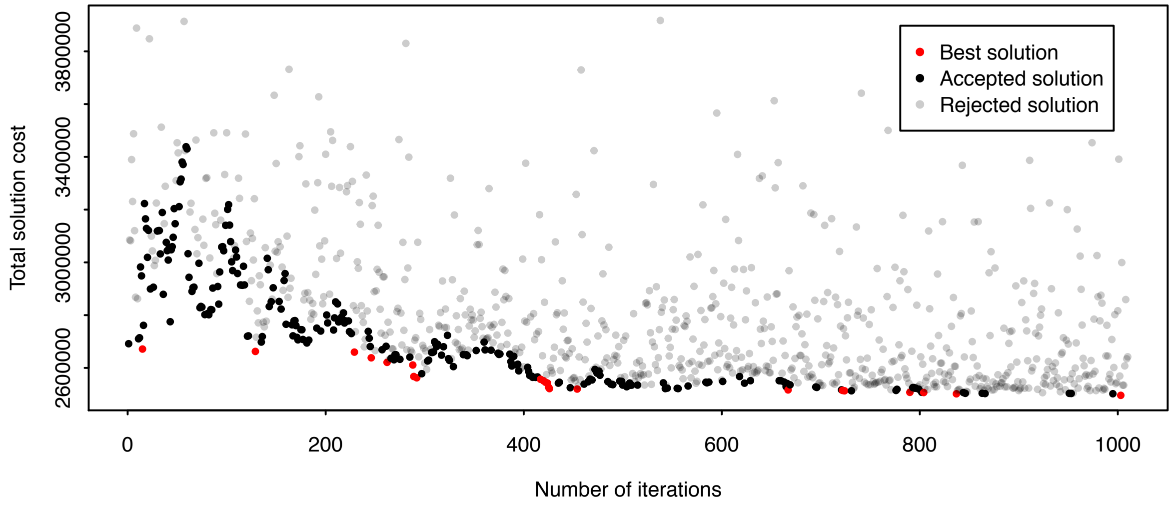


Figure 2: Total solution cost at every iteration (I1-I2-R1-R2).

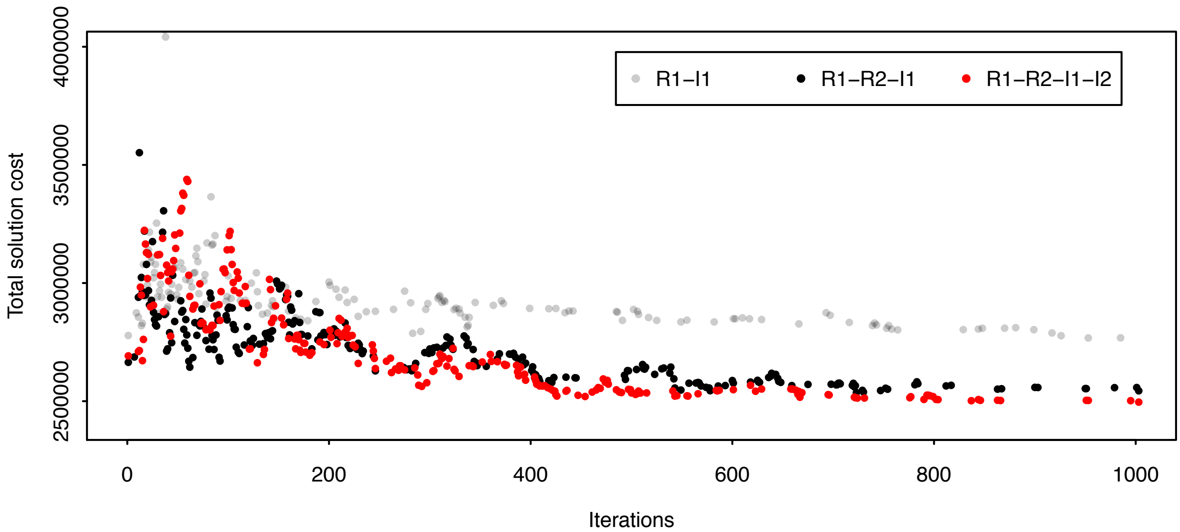


Figure 3: Total cost of accepted solution at every iteration.

R1-R2-I1 The trains scheduled to run on the disrupted track between 7:00am and 9:00am are cancelled. In addition, Algorithm 1 is run with three operators: R1, R2 and I1.

R1-R2-I1-I2 The trains scheduled to run on the disrupted track between 7:00am and 9:00am are cancelled. In addition, Algorithm 1 is run with four operators: R1, R2, I1 and I2.

Fig. 2 depicts the cost of the solution at every iteration in scenario I1-I2-R1-R2. It illustrates how the algorithm can escape from local minima by accepting worsening solutions, especially at the beginning of the running process. It can be observed that a relatively large number of worsening solutions is accepted at the beginning (high number of black dots at low iteration counts), while the algorithm becomes more conservative towards the end (higher proportion of grey dots).

The convergence of the algorithm can be compared for the three different rescheduling scenarios in Fig. 3, which depicts only accepted solutions. It can be observed that the inclusion of demand-driven operators R2 and I2 decreases the cost of the best solution found. Furthermore, running Algorithm 1 only with random removal and insertion operators R1 and I1 yields no improvement to the total cost, as can be seen explicitly from Table 2 that reports the usage of the different operators for the three scenarios where rescheduling was applied. The number of times an operator is used is reported in column # Used, while the number of times an operators finds a new best solution, a solution with a better cost than the current one or a worsening solution accepted by simulated annealing are reported in # Best, # Better and # Accepted, respectively. It can be seen that operator R1 and I1 in scenario R1-I1 fail to find a new best solution. In scenario R1-R2-I1, demand-driven operator R2 is used 624 times, while random operator R1 is used only 386 times. This can be explained by the better performance of R2 (higher # Best, # Better and # Accepted) that is rewarded by the adaptive weight adjustment. Surprisingly, in scenario R1-R2-I1-I2, demand-driven operator I2 is less used than random operator I1. However, this scenario globally over-performs scenario R1-R2-I1 and is therefore maintained.

Table 2: Operator usage statistics.

Scenario	Operator	# Used	# Best	# Better	# Accepted
R1-I1	R1	1010	0	79	80
	I1	1010	0	79	80
R1-R2-I1	R1	386	4	29	47
	R2	624	9	102	70
	I1	1010	13	131	117
R1-R2-I1-I2	R1	375	3	31	39
	R2	635	19	83	76
	I1	640	19	85	65
	I2	370	3	29	50

Table 3 presents the results for the five scenarios. The two first rows report the performance of the timetable without rescheduling (i.e., only the passenger assignment model is applied to evaluate the timetable, but no rescheduling is considered). The three subsequent rows show the results of the best solution found by the rescheduling process, with different operators used in Algorithm 1. Columns z, z_p and z_o report the total, passenger and operational cost, respectively. The improvement (in %) of the total cost, with respect to the disrupted scenario, is also reported. #DP is the number of disrupted passenger, i.e. passengers that cannot reach destination because of the saturation of the network, while #T is the number of trains in the solution. The last column presents the computational time needed to obtain the results. Again, it can be seen that scenario R1-I1 does not improve the total cost, compared with the disrupted scenario. However, scenarios

R1-R2-I1 and R1-R2-I1-I2 decrease the total cost by 5.1%, respectively 6.7%. The number of disrupted passengers is also decreased significantly, especially in scenario R1-R2-I1-I2. It is interesting to observe that the best solution found in the R1-R2-I1-I2 scenario uses less trains than the disrupted scenario, while the number of train minutes increases, showing that the trains in the solution perform longer journeys.

Table 3: Simulation results, with and without rescheduling.

Scenario	z [min] (Improv.)	z_p [min]	z_o [min]	# DP	# T	Time [s]
Regular	2,573,678.6	2,565,527.6	8,151.0	1,326	207	< 1
Disrupted	2,674,223.5	2,666,630.5	7,593.0	2,847	197	<1
R1-I1	2,674,223.5 (0%)	2,666,630.5	7,593.0	2,847	197	663
R1-R2-I1	2,536,551.1 (-5.1%)	2,525,843.1	10,708.0	2,152	186	1,024
R1-R2-I1-I2	2,496,095.8 (-6.7%)	2,483,594.8	12,501.0	1,645	194	1,140

6 Conclusion

Motivated by the need for a passenger-centric framework for the train rescheduling problem in case of severe disruptions, this paper presented a hybrid methodology that takes into account the viewpoint of the train operator and of the passengers when designing a disposition timetable. The problem is defined as an integer linear program and solved using a rescheduling meta-heuristic that generates operationally feasible timetables. A capacitated passenger assignment model evaluates the generated disposition timetable in terms of passenger satisfaction iteratively. The proposed methodology is applied on a sample network and gives satisfactory results in reasonable computational time.

Directions for further research include the comparison of the heuristic algorithm with the exact formulation on a small instance, in order to verify its validity. The inclusion of operators that are more specific to severe disruptions (e.g., partial train cancellation, train rerouting, addition of bus services) is the obvious extension of the current framework. Its high flexibility can easily incorporate new disruption-driven operators to remove or insert trains in the timetable. Eventually, a case study with real data is also expected.

7 References

Ahuja, R. K., T. Magnanti and J. Orlin (1993) *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, NJ.

Albrecht, A., D. Panton and D. Lee (2013) Rescheduling rail networks with maintenance disruptions using Problem Space Search, *Computers & Operations Research*, **40** (3) 703–712, March 2013, ISSN 03050548.

Barrena, E., D. Canca, L. C. Coelho and G. Laporte (2014) Single-line rail rapid transit timetabling under dynamic passenger demand, *Transportation Research Part B: Methodological*, **70**, 134–150, December 2014, ISSN 01912615.

Bierlaire, M. (2015) *Optimization: principles and algorithms*, Presses polytechniques et universitaires romandes.

Binder, S., J. Chen and M. Bierlaire (2014) Generation and evaluation of passenger-oriented railway disposition timetables, paper presented at the *Proceedings of the 14th Swiss Transportation Research Conference*, Ascona, Switzerland.

Cacchiani, V., D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf and J. Wagenaar (2014) An overview of recovery models and algorithms for real-time railway rescheduling, *Transportation Research Part B: Methodological*, **63**, 15–37, May 2014, ISSN 01912615.

Cadarso, L., A. Marín and G. Maróti (2013) Recovery of disruptions in rapid transit networks, *Transportation Research Part E: Logistics and Transportation Review*, **53**, 15–33, July 2013, ISSN 13665545.

Hamdouch, Y. and S. Lawphongpanich (2008) Schedule-based transit assignment model with travel strategies and capacity constraints, *Transportation Research Part B: Methodological*, **42** (7-8) 663–684, August 2008, ISSN 01912615.

Kanai, S., K. Shiina, S. Harada and N. Tomii (2011) An optimal delay management algorithm from passengers' viewpoints considering the whole railway network, *Journal of Rail Transport Planning & Management*, **1** (1) 25–37, November 2011, ISSN 22109706.

Kunimatsu, T., C. Hirai and N. Tomii (2012) Train timetable evaluation from the viewpoint of passengers by microsimulation of train operation and passenger flow, *Electrical Engineering in Japan*, **181** (4) 459–467, December 2012, ISSN 04247760.

Louwerse, I. and D. Huisman (2014) Adjusting a railway timetable in case of partial or complete blockades, *European Journal of Operational Research*, **235** (3) 583–593, June 2014, ISSN 03772217.

Narayanaswami, S. and N. Rangaraj (2013) Modelling disruptions and resolving conflicts optimally in a railway schedule, *Computers & Industrial Engineering*, **64** (1) 469–481, January 2013, ISSN 03608352.

Nguyen, S., S. Pallotino and F. Malucelli (2001) A Modeling Framework for Passenger Assignment on a Transport Network with Timetables, *Transportation Science*, **3**, 238–249.

Ropke, S. and D. Pisinger (2006) An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows, *Transportation Science*, **40** (4) 455–472, November 2006, ISSN 0041-1655.

Swiss National Railways (2013) SBB Facts and Figures, *Technical Report*.