Efficient exploration of the multiple objectives of the railway timetable rescheduling problem

Stefan Binder

Yousef Maknoon

Michel Bierlaire

Ecole Polytechnique Fédérale de Lausanne   **May 2017**

STRC | **17th Swiss Transport Research Conference**
Monte Verità / Ascona, May 17 – 19, 2017

Ecole Polytechnique Fédérale de Lausanne

Efficient exploration of the multiple objectives of the railway timetable rescheduling problem

Stefan Binder, 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,michel.bierlaire}@epfl.ch

Yousef Maknoon
Transport and Logistics Group
Delft University of Technology
Jaffalaan 5, 2628 BX Delft, Netherlands
phone:
fax:
M.Y.Maknoon@tudelft.nl

May 2017

Abstract

Major disruptions, such as the unavailability of railway tracks due to unexpected events (e.g., adverse weather conditions), occur regularly in railway networks. In this research, we address the timetable rescheduling problem arising in case of such a disruption. When designing the so-called disposition timetable, a compromise between the following aspects needs to me made: deviation from the undisrupted timetable (both from the operator's and the passengers' point of view), low operational cost, and acceptable passenger service. We propose a heuristic framework that quantifies these aspects, in which the timetable is optimized using an Adaptive Large Neighborhood Search meta-heuristic, and a passenger assignment model evaluates the timetable in an iterative manner. The operators of the rescheduling heuristic are inspired from real-life recovery strategies (such as train cancellations, delays, reroutings and bus or taxi additions), but also from optimization methods (e.g., feasibility restoration operators). Using a heuristic to solve the rescheduling problem allows for an efficient investigation of its multiple dimensions, as we show on a realistic case study based on the S-train network of Canton Vaud, Switzerland.

Keywords

1 Introduction

Major disruptions, such as the unavailability of railway tracks due to unexpected events (e.g., rolling stock breakdown, adverse weather conditions), occur regularly in railway networks. Due to its complexity, the recovery problem is usually broken up into three consecutive phases: timetable rescheduling, rolling stock rescheduling and crew rescheduling. In this research, we focus on the issue of timetable rescheduling, as the integrated problem cannot be solved in any reasonable time. We consider large-scale disruptions related to the unavailability of one (or more) tracks between two (or more) stations for a known time period. We therefore look at the rescheduling problem from a macroscopic point of view, disregarding details such as track assignments in stations or signalling.

When a disruption occurs in a railway network, the original timetable needs to be updated to a so-called disposition timetable. The latter needs to be conflict-free in terms of operational constraints (e.g., no two trains can be scheduled on the same resource at the same time) and as convenient as possible for the passengers. The objective of the railway operator is to minimize the operational costs, while the aim of the passengers is to receive the best possible level of service. The two goals are usually incompatible: the best possible service for the passengers may also be the most expensive option for the operator. This inadequacy is the key motivation for our work: constructing disposition timetables that take into account passenger satisfaction, while keeping operational costs low. Furthermore, the deviation from the undisrupted timetable is considered as a common objective to be minimized by both the passengers and the operator.

In Binder *et al.* (2017b), we formulated the multi-objective railway timetable rescheduling problem as an Integer Linear Program. It minimizes the passenger inconvenience and imposes upper bounds (using epsilon-constraints) on the operational cost and the deviation from the undisrupted timetable. The model was validated on a case study inspired from the Dutch Railway network. Small to medium-sized instances could be solved to optimality in reasonable computational time using CPLEX. However, the number of passengers the model can handle is limited, as every passenger adds another layer of decision variables to the problem. Also, the substantial number of rescheduling options for every train limits the size of solvable problems.

In order to overcome the shortcomings of Binder et al. (2017), we propose a new solution algoritm for this problem. An Adaptive Large Neighborhood Search (ALNS) meta-heuristic is implemented to construct the disposition timetable in an iterative manner. Neirhborhood operators remove/add trains from/to the timetable at every iteration of the algorithm. The operators of the rescheduling heuristic are inspired from real-life recovery strategies (such as

train cancellations, delays, reroutings and bus or taxi additions) and from optimization methods (e.g., feasibility restoration operators). The algorithm keeps track of non-dominated solutions using an archive of solutions.

Using this heuristic to solve the rescheduling problem allows for an efficient investigation of its multiple dimensions, as we show on a real case study based on the morning peak hour of the S-train network of Canton Vaud, Switzerland. The exploration of the three-dimensional Pareto frontier allows to analyze the trade-off between the objectives and to quantify the quality of the disposition timetable according to the objectives.

The remainder of this paper is structured as follows. Section 2 recalls the description of the multi-objective railway timetable rescheduling problem, which was initially introduced by Binder *et al.* (2017b). In Section 3, we present in detail the novel solution algorithm we propose. Section 4 reports first results of the computational experiments on the case study. Finally, Section 5 concludes the paper and provides directions for further research.

2 Problem description

In the passenger railway service, a timetable is defined as the set of arrival and departure times of every train at each station where it stops. In the event of a disruption in the network, a so-called disposition timetable needs to be implemented by the railway operator. The latter is thus faced with the challenge of obtaining a timetable that reflects a compromise between the following objectives: minimization of passenger inconvenience, of operational costs and of the deviation from the undisrupted timetable.

The problem addressed in this paper is to generate a set of "good" disposition timetables, and to quantify the trade-off between the objectives in order to assist railway operators in the design of such timetables. The multi-objective timetable rescheduling framework we present in this section builds upon the work of Binder *et al.* (2017b). For the sake of completeness, we recall here its main features and assumptions. We begin by describing the macroscopic infrastructural model representing the operational side of the problem. The formal definition of the disruption, and of the associated recovery decisions, is given thereafter. We then explain the assumptions on passenger behavior that lead to our passenger travel choice model and to the passenger's assignment on the railway network. We conclude this section by presenting the mathematical formulation of the problem as a space-time graph, which allows us to formally define the objective functions.

2.1 Infrastructural model

Time is discretized into $n + 1$ time intervals of length τ (typically, one minute) and we introduce the set of time steps $H = \{0, \tau, 2\tau, \ldots, n\tau\}$, where $n\tau$ is the considered planning horizon. We model the railway network at a macroscopic level. The infrastructure is represented by a set of stations $s \in S$ and a set of tracks $Q \subseteq S \times S$ connecting the stations. A track $(s, s') \in Q$ is an uninterrupted railway track linking s to s' directly, without passing in any other station. Each station s is characterized by the presence or absence of a shunting yard. We denote by $S_R \subseteq S$ the subset of stations with a shunting yard, and by R the set of shunting yards. Every shunting yard $r_s \in R$ is associated with exactly one station $s \in S_R$.

We define two stations $s, s' \in S$ to be neighbouring if $(s, s') \in Q$ and $(s', s) \in Q$. Between two neighbouring stations, the running time $t(s, s')$, in minutes, and the distance $d(s, s')$, in kilometers, are known and equal for all trains. Trains cannot switch tracks between stations and overtakings occur only within stations (i.e., a platform in a station can be reached from any incoming/outgoing track). Each track can be used in one direction at the time, or it can be assigned to opposite directions alternatively. A certain headway is respected if two consecutive trains are running in the same direction on the same track. In case the track is used in opposite directions, a set of conflicting movements is defined to ensure proper separation of the trains.

Two different types of trains are considered: *original trains* and *emergency trains*. The set of original trains K_1 contains the trains that are operated in the undisrupted timetable. Their schedule is an input to the rescheduling model. The set of emergency trains K_2 represents trains that are located in shunting yards, ready to be scheduled if needed. All emergency trains begin and end their journey at a shunting yard, and n_r denotes the number of emergency trains available in shunting yard r. The set of all trains is denoted by $K = K_1 \cup K_2$, and all trains in K are characterized by their capacity q_k, defined as the maximal number of on-board passengers.

2.2 Disruption and associated recovery decisions

We define a disruption in the network as the unavailability of at least one whole track between neighboring stations. Multiple tracks can become unavailable at the same time, and at different locations in the network. We assume that the length of the disruption is known in advance, and that it is less or equal to the considered time horizon. Hence, we also consider what happens after the end of the disruption (i.e., once all tracks can be used again).

In order to recover from the disruption, we consider the five following decisions (the three first ones concern original trains):

Cancellation A train may be fully or partially canceled. A partially canceled train is only operated on a subset of the stations of its original route and canceled afterwards. Observe that a full cancellation is a special case of a partial cancellation.

Delay The arrival or departure of a train at a station may be delayed up to a maximal amount of time. A train may also be delayed only for a part of its route. We do not allow trains to run earlier than in the undisrupted timetable, as this is usually avoided in practice because passengers might miss their planned train.

Rerouting A train may be rerouted through another path than the originally planned one.

Emergency train At every station with a shunting yard, a limited number of emergency trains is available. These may be scheduled as needed.

Emergency bus If the track between two neighboring stations is disrupted, an emergency bus may be scheduled to connect the two stations directly.

2.3 Passenger travel choice

Passenger demand is assumed to be known, in the form of an origin-destination (OD) matrix. The latter describes the number of passengers entering the system at a given origin station, at a certain time, and who wish to travel to a given destination station. The availability of such data becomes more and more frequent with the gradual introduction of smart cards in public transportation networks. Based on the OD matrix, a passenger p is denoted by a triplet (o_p, d_p, t_p), where $o_p \in S$ is the origin station, $d_p \in S$ the destination station, and $t_p \in H$ the desired departure time from the origin. Note that, as we assume deterministic train travel times in our approach, a passenger can equivalently be characterized by the desired arrival time at the destination. We adopt the former representation in the following. The set of all passengers is denoted by P.

For every passenger, we consider the set $\Omega(o_p, d_p)$ of all paths linking the origin station o_p to the destination station d_p. A passenger path is a sequence of access, in-vehicle, waiting, transfer and egress movements (refer to Section 2.4 for a definition in terms of arcs in a space-time graph). In order to distinguish different paths, we associate a generalized cost with every alternative (i.e., path) and assume that each passenger group chooses the one with the lowest generalized cost. Similarly to Binder *et al.* (2017a), we assume that the price of the trip is equal among all paths for a given passenger, and we focus on homogeneous demand (thus disregarding socio-economic characteristics differentiating the passengers). The generalized cost of path $\omega \in \Omega(o_p, d_p)$ for

passenger p therefore depends on the following attributes:

- *In-Vehicle Time* (VT_ω^p): time, in minutes, spent by the passenger in one (or more) train(s) along the path,
- *Waiting Time* (WT_ω^p): time, in minutes, spent by the passenger waiting between two consecutive trains at a station along the path (does not consider the waiting time for the first train),
- *Number of Transfers* (NT_ω^p): number of times the passenger needs to change trains along the path,
- *Early Departure* ($ED_\omega^p = \max(0, t_p - t)$): time difference (in minutes) between the desired (t_p) and the actual (t) departure time from origin, if early,
- *Late Departure* ($LD_\omega^p = \max(0, t - t_p)$): time difference (in minutes) between the actual (t) and the desired (t_p) departure time from origin, if late.

Based on the aforementioned description for a given passenger p, the generalized cost of alternative ω is defined as follows:

$$C_\omega^p = VT_\omega^p + \beta_1 \cdot WT_\omega^p + \beta_2 \cdot NT_\omega^p + \beta_3 \cdot ED_\omega^p + \beta_4 \cdot LD_\omega^p, \tag{1}$$

where $\beta_1, \ldots, \beta_4$ are the relative weights of the attributes described above. C_ω^p is in minutes and expresses the generalized travel time of passenger p along path $\omega \in \Omega(o_p, d_p)$. As commonly done in the literature, the weights of the various elements of the generalized travel time are defined relative to the in-vehicle time of the path. We use the values reported in Table 3, obtained from the literature.

2.3.1 Passenger assignment model

Obtaining accurate passengers flows in a public transportation network where passengers compete for the limited capacity of the trains is a challenging task. When the number of passengers attempting to board a train exceeds its available capacity, it has to be decided eventually which passengers can board the train and which cannot. There are two main paradigms to take this decision:

System optimum Passengers are assumed to collaborate in order to minimize the overall inconvenience (i.e., generalized travel time) of all passengers.

User equilibrium Passengers are assumed to be selfish actors that attempt to minimize their personal inconvenience.

Although a system optimal passenger assignment yields a better experience for everyone *on average*, public transportation users are usually not willing to accept a longer personal travel time for a theoretical "greater good". In this paper, we therefore assume passengers to be selfish and independent and to maximize their personal utility, given by Eq. (1). We use the passenger assignment model introduced by Binder *et al.* (2017a), where an exogenous ordering of the passengers is defined in order to assign the passengers incrementally on the network.

2.4 Mathematical formulation as a space-time graph

The timetable on which the passengers are assigned is mathematically represented by a directed space-time graph $G(V, A)$. The set of nodes $V = N \cup N_O \cup N_D$ consists of three different types of nodes. Starting from an empty graph, we add a space-time node $(s, t, k) \in N$ for each arrival/departure event of train $k \in K$ at/from station $s \in S$ at time $t \in H$. For instance, if train $k' \in K$ leaves station $s_1 \in S$ at time $t_1 \in H$, stops at station $s_2 \in S$ from time $t_2 \in H$ to $t_3 \in H$ and finishes its trip at station $s_3 \in S$ at time $t_4 \in H$, four space-time nodes are added: (s_1, t_1, k'), (s_2, t_2, k'), (s_2, t_3, k') and (s_3, t_4, k'). In addition, N_O and N_D are the sets of time-invariant origin and destination nodes of the passengers. We denote by $s(o)$ and $s(d)$ the station associated with node $o \in N_O$ and $d \in N_D$, respectively.

There are six types of arcs in the graph:

- *Driving arcs* model the movements of trains between stations. From the timetable, we define, for every train $k \in K$, the set of driving arcs A_{Dri}^k. A driving arc connects a departure event at one station $(s, t, k) \in N$ to an arrival event at the following station $(s', t', k) \in N$, with $t' = t + t(s, s')$. By repeating this procedure for every train in the timetable, we construct the set of driving arcs $A_{Dri} = \cup_{k \in K} A_{Dri}^k$.
- *Waiting arcs* model trains waiting at a station for passengers to board or alight. We define from the timetable, for every train $k \in K$, the set of waiting arcs A_{Wai}^k. A waiting arc connects an arrival event at a station $(s, t, k) \in N$ to a departure event from the same station $(s, t', k) \in N$, with $t' = t + w(s)$, where $w(s)$ is the waiting time at station s. By repeating this procedure, we construct the set of waiting arcs $A_{Wai} = \cup_{k \in K} A_{Wai}^k$.
- *Access arcs* model passenger p arriving at the origin. They are given by the set $A_{Acc}^p = \{(o, (s, t, k)) \in N_O \times N | s = s(o) = o_p\}$. Note that, by definition, passenger p can therefore take any train that departs from his origin station.
- *Egress arcs* model passenger p leaving the system at destination. They are given by the set $A_{Egr}^p = \{((s, t, k), d) \in N \times N_D | s = s(d) = d_p\}$.
- *Transfer arcs* model passengers transferring from one train to another in a station, with

Table 1: Arc weights and capacities.

Name	Start node	End node	c_a	c_a^p	q_a
Driving	(s, t, k)	(s', t', k)	$c \cdot d(s, s')$	$t' - t$	q_k
Waiting	(s, t, k)	(s, t', k)	0	$\beta_1 \cdot (t' - t)$	q_k
Access	o_p	(s, t, k)	-	$\beta_3 \cdot \max(0, (t_p - t)) + \beta_4 \cdot \max(0, (t - t_p))$	1
Egress	(s, t, k)	d_p	-	0	1
Transfer	(s, t, k)	(s, t', k')	-	$\beta_2 + (t' - t)$	∞
Penalty	o_p	d_p	-	$n\tau$	∞

a minimal transfer time m and a maximal transfer time M. The set of transfer arcs is constructed in the following way: $A_{Tra} = \{((s, t, k), (s, t', k')) \in N \times N | \forall s \in S, \forall k \in K, \forall k' \in K \setminus \{k\}, \forall t, t' \in H : m \leq t' - t \leq M\}$.

- *Penalty arcs* model passengers not succeeding to find a path from origin to destination. They are given by the set $A_{Pen} = \{(o, d) \in N_O \times N_D\}$.

The set of arcs is given by $A = A_{Dri} \cup A_{Wai} \cup (\cup_{p \in P} A_{Acc}^p) \cup (\cup_{p \in P} A_{Egr}^p) \cup A_{Tra} \cup A_{Pen}$. With each arc a, a capacity q_a and two weights (c_a^p and c_a) are associated. The capacity of driving and waiting arcs is given by the capacity of the associated train (i.e., the maximum number of passengers that can be in the train at the same time), q_k. Access and egress arcs have a capacity of one, as they are passenger-specific. Transfer and penalty arcs have infinite capacity. The cost of using an arc a for a train (c_a), or for a passenger p (c_a^p), are listed in Table 1. We assume that the operational cost is proportional to the distance travelled by the trains. Therefore, only driving arcs have an operational cost c_a different from zero (c is the cost of running a train, per kilometer). Passenger arcs are weighted according to the generalized travel time defined by Eq. (1). The cost of a path in the graph for a passenger is obtained by summing the weights c_a^p of the arcs in the path. Note that driving and waiting arcs are weighted differently if used by a train or a passenger.

2.5 Objective functions

The operator and the passengers have different goals in railway operations. Passengers would prefer a direct train from their origin to their destination, arriving exactly at their desired arrival time. Considering the operational cost of this solution, it is obviously impossible to provide such a service to every passenger. Hence, a trade-off between these two objectives needs to be found. In case of disruptions in railway operations, we also need to take into account the deviation from the undisrupted timetable. Considering this "cost" is necessary in order to avoid solutions where the schedule of the entire network is overhauled because of a local disruption.

This is beneficial both for the passengers and the operator. Also, once the disruption is resolved, it is easier and quicker to come back to the undisrupted timetable if the disposition timetable is not too different. Also, the deviation cost captures the fact that our model for operational cost or passenger dissatisfaction might not be comprehensive, i.e., if we had true representations for the latter, there would be no need for a deviation cost. Operational cost, passenger inconvenience and deviation from the undisrupted timetable are considered as three objectives to minimize.

The three objective functions are defined as follows. As detailed in Section 2.3, *passenger inconvenience* (z_p) is given by the generalized travel time of the passengers. The *operational cost* of the timetable (z_o) is the running cost of original trains as well as emergency trains. The *deviation cost* (z_d) represents the deviation from the undisrupted timetable and is a weighted sum of the different rescheduling possibilities: cancellations, reroutings, delays and the cost of adding an emergency train (the respective weighting factors are $\delta_c, \delta_r, \delta_d, \delta_e$).

3 Solution algorithm

The multi-objective railway timetable rescheduling problem (MO-RTRP) defined in the previous section has been formulated as an Integer Linear Program in Binder *et al.* (2017b). The authors illustrate the problem on a realistic case study and solve it to optimality on several instances. The size of solvable instances, however, remains limited, mainly because every additional passenger adds another layer of decision variables to the problem.

In this section, we propose a new solution algorithm for the MO-RTRP. It is based on Adaptive Large Neighborhood Search (ALNS), a heuristic proposed by Ropke and Pisinger (2006) to solve the Pickup and Delivery Problem with Time Windows. It has also been used recently in railway timetable scheduling (see, e.g., Barrena *et al.* (2014)*,* Robenek *et al.* (2017)). ALNS is a type of large neighborhood search in which a number of fairly simple operators compete in modifying the current solution. At every iteration of the algorithm, the operators add and remove trains to the timetable. We use simulated annealing as the search guiding meta-heuristic, as it seems to be the preferred approach in the ALNS literature.

The pseudo-code of the ALNS framework is shown in Algorithm 1. The sections referenced in the algorithm indicate where the different concepts are explained in further detail. The algorithm assumes the existence of an initial solution. A solution *s* is defined as a timetable, i.e., the departure and arrival times of all trains at all stations. The initial solution does not need to be feasible, hence the undisrupted timetable is used as an input. In the context of multi-objective

optimization, it is common to define dominance relations among solutions in order to consider the trade-offs between the differents objectives (see Section 3.1 for formal definitions). The algorithm produces an archive of non-dominated solutions $\mathcal{A}$.

Input : Initial solution s

            Initial (final) temperatures T_i^{start} (T_i^{end}), $i \in \{p, o, d\}$

            Set of neighborhood operators Π

            $\rho_1 > \rho_2 > \rho_3 \geq 0$

Output : Archive of non-dominated solutions $\mathcal{A}$

---

1   $T_i \leftarrow T_i^{start}, i \in \{p, o, d\}, \mathcal{A} \leftarrow \{s\}, n \leftarrow 0$

2   **while** $T_i > T_i^{end}, i \in \{p, o, d\}$ **do**

3      $n \leftarrow n + 1$

4   Select neighborhood operator $\pi \in \Pi$ (Section 3.2)

5   Apply π to s, and obtain s'

6   Assign passengers on s' (Section 2.3.1)

7   Evaluate $z_i(s'), i \in \{p, o, d\}$ (Section 2.5)

8   **if** *s' can be archived (Section 3.1.1)* **then**

9   Add s' to $\mathcal{A}$ (Section 3.1.1)

10          $s \leftarrow s'$

11   Update the score of π by ρ_1 (Section 3.2)

12   **else**

13   **if** *s' is accepted by the SA criterion (Section 3.1.1)* **then**

14              $s \leftarrow s'$

15   Update the score of π by ρ_2 (Section 3.2)

16   **else**

17   Update the score of π by ρ_3 (Section 3.2)

18   Update the temperatures T_i according to the annealing schedule (Section 3.1.2)

19   Periodically, update weights of all operators and reset scores (Section 3.2)

20   Periodically, select s randomly in $\mathcal{A}$ (Section 3.1.3)

21   **return** A

Algorithm 1: ALNS framework.

3.1 Multi-objective simulated annealing

Kirkpatrick *et al.* (1983) initially proposed the meta-heuristic known as simulated annealing (SA). Its characteristic feature is a probabilisitic criterion for the acceptance of worsening solutions

in the search for a global optimum. In contrast with simple descent heuristics that only accept solutions which improve the value of the objective function, this allows to avoid that the procedure gets trapped in a local optimum. As SA is a well-known heuristic, we will focus here on its extension to a multi-objective problem, based on the work of Suppapitnarm *et al.* (2000).

3.1.1 Archiving and acceptance criterion

In order to address the multi-objective aspect of the problem, we define the concepts of dominance and Pareto optimality. Let S be the set of all solutions of the MO-RTRP, and s_1, s_2 two particular solutions. We say that s_1 dominates s_2 (equivalently, s_2 is dominated by s_1) if $z_i(s_1) \leq z_i(s_2), i \in \{p, o, d\}$ and $\exists i \in \{p, o, d\} | z_i(s_1) < z_i(s_2)$. In other words, s_1 is not worse than s_2 in any objective and s_1 is strictly better than s_2 in at least one objective. We denote this property by $s_1 \prec s_2$. A solution $s^* \in S$ is said to be Pareto optimal if it is not dominated by any other solution: $\nexists s \in S | s \prec s^*$. The set of all Pareto optimal solutions is $\mathcal{P}^* = \{s^* \in S | \nexists s \in S : s \prec s^*\}$.

The derivation of the entire set $\mathcal{P}^*$ is beyond the scope of this paper. Instead, we maintain an "archive" $\mathcal{A}$ of non-dominated solutions in Algorithm 1, as an approximation of $\mathcal{P}^*$. We begin by adding the initial solution to the archive (line 1). Then, at every iteration, the current solution s' is a candidate for archiving (line 8). The following cases need to be considered:

a) s' dominates at least one solution in $\mathcal{A}$,
b) s' neither dominates nor is dominated by any solution in $\mathcal{A}$,
c) s' is dominated by at least one solution in $\mathcal{A}$.

In the two first cases, s' is added to the archive (lines 9–11). Additionally, in case a), all solutions dominated by s' in $\mathcal{A}$ are removed from the archive. In the last case, s' is not added to the archive (lines 12–17) and $\mathcal{A}$ is not modified.

Given this archiving procedure, we define the following acceptance criterion for the current solution s'. If s' is archived, it is automatically accepted (line 10). If not, it is accepted with probability

$$p = \prod_{i \in \{p, o, d\}} \min \left\{ \exp \left(-\frac{z_i(s') - z_i(s)}{T_i} \right), 1 \right\}. \tag{2}$$

This acceptance probability is the product of individual acceptance probabilities for each

objective, with its associated temperature T_i. It is therefore not necessary to scale the objectives with respect to each other to form a composite objective function. Note that the individual acceptance probability, $\exp((z_i(s') - z_i(s))/T_i)$, may be greater than unity (if the difference is negative). We therefore take the minimum between the calculated value and one.

3.1.2 Temperature update

In SA, the "temperature" of the system is a control parameter that indicates how likely it is that a worsening solution is accepted in the search process. For high values of T_i in Eq. (2), virtually all new solutions are accepted, irrespective of the sign of the numerator. Conversely, only small increases in the objective function z_i are accepted for small values of T_i. Thus, the search is initiated with high values for the temperature, in order to allow for a broad exploration of the search space, and to escape local minima. The temperature is then gradually decreased as the search converges to a (hopefully global) minimum.

We use the cooling schedule proposed in Suppapitnarm *et al.* (2000) to update the temperatures. The temperatures T_i are initially set to a large value. Hence, virtually all solutions are accepted in this phase. After a pre-determined number of iterations N_1, each temperature T_i is set to the value of the standard deviation of the respective objective function, $\sigma(z_i)$, computed over accepted solutions. After this warm-up phase, the temperatures are lowered after a given number of iterations N_2, or a given number of acceptances, N_2', with $N_2' = 0.4N_2$, whichever comes first, according to the following formula:

$$T_i' = \alpha_i T_i,$$

where T_i' are the new temperatures and

$$\alpha_i = \max\left\{0.5, \exp\left(-\frac{0.7T_i}{\sigma_i}\right)\right\}.$$

σ_i is the standard deviation of the values of z_i, computed over solutions accepted at temperature T_i. The counter N_2 is then reset to zero.

3.1.3 Return-to-base strategy

In order to explore more intensively the search space close to the Pareto frontier, the random selection of a solution in the archive, from which to recommence the search, is implemented.

We decide the rate at which to come back to a solution from the archive in the following way. In the warm-up phase before the iteration count hits N_1, we do not perform any return-to-base moves, as the goal is to explore the search space as freely as possible at that time. Thereafter, the rate of return to solutions of the archive is increased in the following manner. Let N_i^B be the number of iterations to be executed prior to the ith return-to-base. We assume that this number decreases, according to $N_i^B = r_B N_{i-1}^B, i = 2, 3, 4, \ldots$, where r_B is a parameter between 0 and 1. We assume a lower bound for $N_i^B, \forall i$ of 10 iterations. Suitable values of the parameters found by Suppapitnarm *et al.* (2000) are $r_B = 0.9$ and $N_1^B = 2N_2$.

The choice of the solution in the archive is performed in a random manner. Note that this might not be a particularly efficient approach, as the search is more likely to return to well-explored parts of the search space, where there are already many archived solutions. A selection process as suggested in Suppapitnarm *et al.* (2000) may be implemented instead.

3.2 Neighborhood operators

Neighborhood operators are used in the ALNS framework in order to modify the current solution. We use the following operators (some of them are inspired from Barrena *et al.* (2014)):

Cancel trains completely This operator selects v trains in K and cancels them completely.

Cancel trains after a given station This operator selects v trains in K and cancels them after their arrival in station s on their path. The last station in the train journey might differ from one train to the next.

Delay trains completely This operators selects v trains in K and delays them on their complete journey by $\mu < \mu_{\max}$ minutes.

Delay trains after a given station This operators selects v trains in K and delays them by $\mu < \mu_{\max}$ minutes after their passage in station s.

Reroute trains between neighboring stations This operator selects v trains in K and reroutes them on a different path between the two neighboring stations s and s'.

Add an emergency train This operator schedules an emergency train, starting its journey from station s at time $t \in H$ and travelling on a pre-defined train line.

Add an emergency bus This operator schedules an emergency bus service between two neighboring stations s and s'.

Operator choice is governed by a roulette-wheel mechanism. We associate a weight w_π with every operator $\pi \in \Pi$. Given the set of operators Π, operator π is chosen with probability $w_\pi / \sum_{\pi' \in Pi} w_{\pi'}$ (line 4). The operator weights are updated using information from earlier iterations. The idea is

to keep track of a score s_π for every operator. This score measures the recent performance of the given operator; a higher score corresponds to a better performance. The ALNS algorithm is divided into segments of J 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. We distinguish three cases: If the last iteration of the algorithm

- resulted in a new solution added to the archive, the operator is rewarded with a score increase $\rho_1 > 0$;
- resulted in a solution that was not added to the archive, but was accepted by the SA criterion, the operator score is increased by $\rho_2 < \rho_1$;
- resulted in a solution that was neither added to the archive, nor accepted by the SA criterion, the score of the selected operator is not increased ($\rho_3 = 0$).

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 π in segment $j + 1$, w_π^{j+1}, is computed as follows:

$$w_\pi^{j+1} = (1 - \eta)w_\pi^j + \eta\frac{s_\pi}{n_\pi},$$

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

3.3 Infeasibility

We allow the algorithm to reach infeasible solutions, in order to explore the search space more efficiently. Two types of infeasibilities are considered:

Headway-related infeasibility A solution is infeasible if a minimal headway h between two consecutive trains on a track (s, s') is not respected. In case both train run in the same direction, the minimal headway is h. In case they travel in opposite directions, the minimal headway is $t(s, s') + h$.

Disruption-related infeasibility A solution is infeasible if at least one train is scheduled on a track that is unavailable because of the disruption. Not that we therefore initialize the algorithm with an infeasible solution, as we use the undisrupted timetable as an initial solution.

In Algorithm 1, before selecting the neighborhood operator, we count the number of infeasibilities of the current solution. If the number is strictly positive, we restrict the roulette-wheel mechanism to select among the first five operators. The chosen operator is then applied on the train causing the first infeasibility. The parameters of the operator (μ, ν, s) are then chosen to attempt to remove the infeasibility. We apply this procedure until all infeasibilities are resolved.

4 Computational experiments

4.1 Case description

We illustrate our methodology on the network of regional S-trains in canton Vaud, Switzerland, during the morning peak hour. The timetable data used in this case study has been downloaded directly from the official website of the Swiss National Railways (SBB), `www.sbb.ch`, for the year 2016.
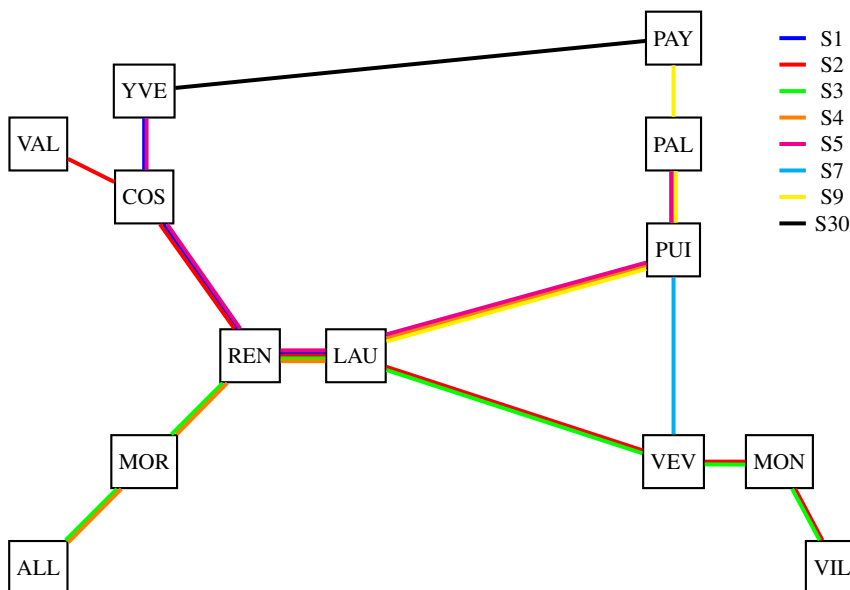

Figure 1: Network of S-trains in canton Vaud, Switzerland (2016).

The reduced network of S-trains is presented in Fig. 1. We consider the 13 main stations in this network, i.e. $S = \{LAU, REN, MOR, ALL, COS, VAL, YVE, VEV, MON, VIL, PUI, PAL, PAY\}$. The timetable of the morning peak hours, between 5:00am and 9:00am, is used for this case study. There are 8 bidirectional lines: S1, S2, S3, S4, S5, S7, S9 and S30. We include all trains with a departure time from the beginning of the line between 5:00am and 9:00am. Table 2 reports the first and last station of every train line, along with the departure time from the first station of

Table 2: List of S-train lines in canton Vaud, Switzerland (2016).

Line	From	To	Departure times						
S1	YVE	LAU	05:28	06:28	07:28	08:28			
	LAU	YVE	05:54	06:54	07:54	08:54			
S2	VAL	VIL	05:10	06:10	07:10	08:10			
	VIL	VAL	05:23	06:23	07:23	08:23			
S3	ALL	VIL		06:07	07:07	08.07			
	VIL	ALL	05:49	06:51	07:51	08:51			
S4	ALL	PAL	05:37	06:37	07:37	08:37			
	PAL	ALL		06:34	07:34	08:34			
S5	YVE	PAL	05:57	06:57	07:57	08:57			
	PAL	YVE		06:06	07:07	08:07			
S7	VEV	PUI		06:09	07:09	08:09			
	PUI	VEV		06:36	07:36	08:36			
S9	LAU	PAY	05:25	06:24	07:24	08:24			
	PAY	LAU	05:40	06:40	07:40	08:40			
S30	PAY	YVE	05:30	06:02	06:30	07:02	07:30	08:02	08:30
	YVE	PAY	05:04	06:04	06:33	07:04	07:33	08:04	08:33

the line. Overall, there are 65 S-trains considered in the morning peak hours. SBB is currently operating Stadler Flirt train units on lines S1, S2, S3, S4 and S5; and Domino train units on the other lines. Stadler Flirt train units have a capacity of 160 seats and 220 standing people, while Domino units can accommodate 188 sitting and 100 standing people. We consider two different disruption scenarios, corresponding to two different levels of disruption "severeness":

- Disruption VEV–MON, where both track between stations VEV and MON are unavailable from 5:00am to 7:00am.
- Disruption LAU–REN, where both track between stations LAU and REN are unavailable from 5:00am to 8:00am.

We consider a deterministic passenger demand, derived from SBB's annual report of 2015 (Swiss Federal Railways (2015)). Not all data required is available, so we rely on realistic assumptions and approximations to generate synthetic passenger data. We consider a total of 14,920 passengers in our case study. The exact procedures and assumptions used to obtain this number can be found in Appendix A of Binder *et al.* (2017a). The generalized travel time of the passengers is computed using the weights given in Table 3. The cost of the penalty arc is the time horizon (four hours). We impose a minimal transfer time m of four minutes and a maximal transfer time M of fifteen minutes.

All of the computational experiments were performed on a computer with a 2.4 GHz Intel Core i7 processor and 8 GB of RAM. The algorithms were implemented in Java. On average, one iteration of Algorithm 1 runs in about 0.2 seconds.

Table 3: Values of weighting factors in the passengers' generalized travel time.

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)

4.2 Results

Figs. 2 and 4 show the approximated Pareto frontiers for the two considered disruptions. The three-dimensional frontiers are projected on the respective axes, and the third objective appears in greyscale.

For the first disruption, the z_o-z_p projection shows that the same level of passenger inconvenience can be achieved at several different operational cost. This can be explained by the fact that the neighborhood operators may relie too much on randomness and therefore introduce unnecessary trains in the timetable, which are not interesting for passengers to board. Also, it can be observed that, for a given level of z_d, substantial improvements in passenger inconvenience can be achieved by only slightly increasing the operational cost. The z_o-z_d projection indicates a clear inverse-proportional relationship of the values of operational cost and deviation: high values of operational cost correspond to low values of deviation, and vice versa. The high weight put on cancellations in the definition of the deviation cost and the definition of operational cost as proportional to travelled distance explain this fact. The passenger inconvenience decreases as the operational cost and the deviation from the undisrupted timetable increase, as expected. Finally, the z_p-z_d projection shows the same findings, from a different angle.

The severeness of disruption LAU–REN, compared to disruption VEV–MON, can be observed in the following ways:

- The lowest possible level of z_d is twice as high for disruption LAU–REN.
- For the same level of z_o and z_d, the passenger inconvenience is higher.
- The operational cost of non-dominated solutions is lower for disruption LAU–REN, as no "interesting" trains can be scheduled on the highly disrupted network.

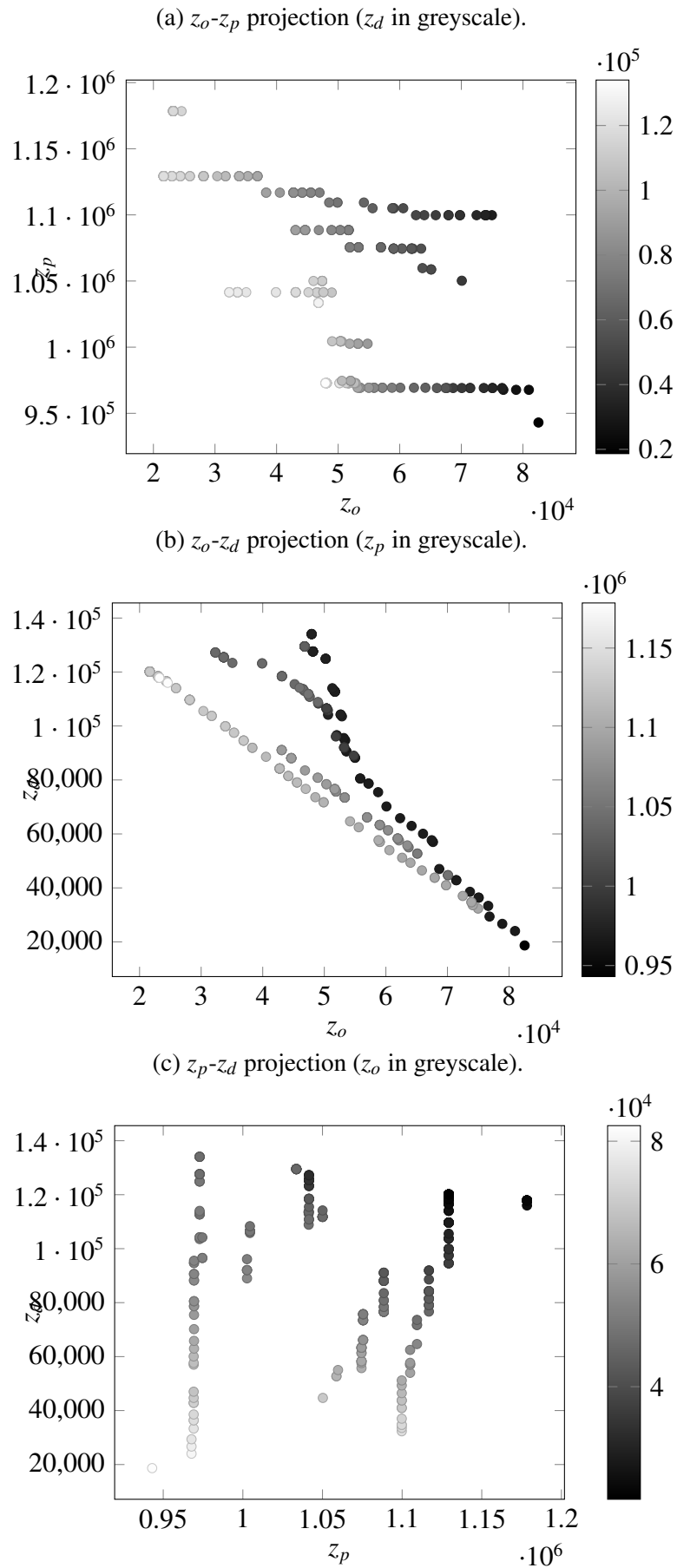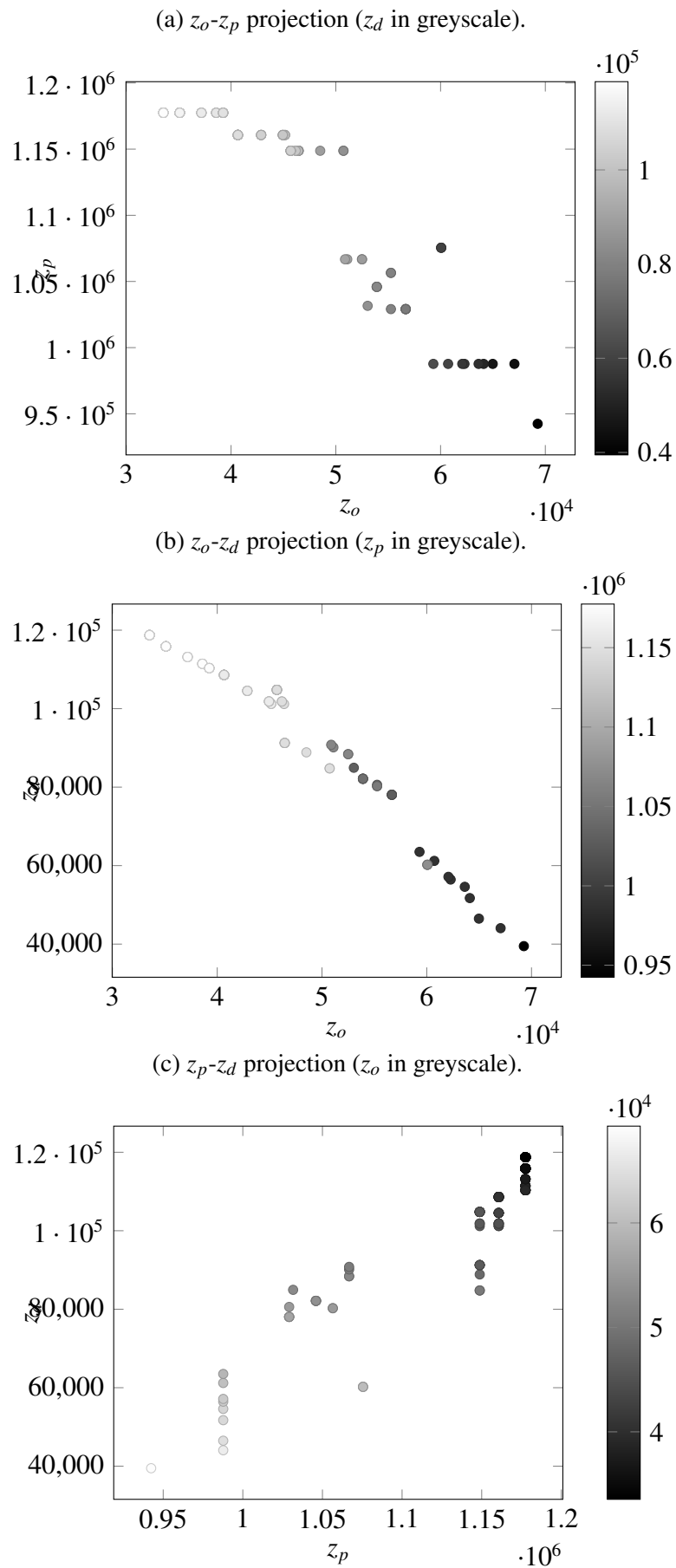Figure 2: Pareto frontiers for disruption VEV–MON.

(a) z_o-z_p projection (z_d in greyscale).

(b) z_o-z_d projection (z_p in greyscale).

(c) z_p-z_d projection (z_o in greyscale).

Figure 4: Pareto frontiers for disruption LAU–REN.

(a) z_o-z_p projection (z_d in greyscale).

(b) z_o-z_d projection (z_p in greyscale).

(c) z_p-z_d projection (z_o in greyscale).

5 Conclusion

Motivated by the need for a passenger-centric framework for the railway timetable 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 solved using a rescheduling meta-heuristic that generates operationally feasible timetables. The proposed methodology is applied on a sample network and gives satisfactory results in reasonable computational time. Every iteration of the heuristic takes less than one second, thus making it practical for the evaluation of several timetables. We present approximations of the Pareto frontier of the problem and analyze the trade-off between the different objectives.

The multi-objective railway timetable rescheduling problem is a hard problem and this work proposes a novel heuristic that drastically reduces the computational time needed to solve it. The use of operators inspired from practice allows train operators to easily implement the framework in order to evaluate the trade-off between the multiple objectives when designing a disposition timetable. Further research will focus on the definition of additional operators, and on the inclusion of our model in a broader framework to solve the complete recovery problem.

6 References

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.

Binder, S., Y. Maknoon and M. Bierlaire (2017a) Introducing exogenous priority rules for the capacitated passenger assignment problem, *Technical Report*, **TRANSP-OR 170209**, Transport and Mobility Laboratory, Ecole Polytechnique Fédérale de Lausanne.

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

de Keizer, B., K. Geurs and G. Haarsman (2012) Interchanges in timetable design of railways: A closer look at customer resistance to interchange between trains, paper presented at the *40th European Transport Conference*, Glasgow.

Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi (1983) Optimization by simulated annealing, *Science*, **220** (4598) 671–680.

Robenek, T., S. Sharif Azadeh, Y. Maknoon and M. Bierlaire (2017) Hybrid cyclicity: Combining the benefits of cyclic and non-cyclic timetables, *Transportation Research Part C: Emerging Technologies*, **75**, 228–253.

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.

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

Suppapitnarm, A., K. A. Seffen, G. T. Parks and P. J. Clarkson (2000) A simulated annealing algorithm for multi-objective optimization, *Engineering Optimization*, **33** (1) 59–85.

Swiss Federal Railways (2015) SBB: Facts and Figures 2015, *Technical Report*. Bibtex:swiss_federal_railways_sbb:_2015.

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