Rescheduling of Railway Rolling Stock with Dynamic Passenger Flows

L. Kroon, G. Maroti and L. Nielsen

ERIM REPORT SERIES *RESEARCH IN MANAGEMENT*	
ERIM Report Series reference number	ERS-2010-045-LIS
Publication	December 2010
Number of pages	35
Persistent paper URL	http://hdl.handle.net/1765/22613
Email address corresponding author	gmaroti@rsm.nl
Address	Erasmus Research Institute of Management (ERIM) RSM Erasmus University / Erasmus School of Economics Erasmus Universiteit Rotterdam P.O.Box 1738 3000 DR Rotterdam, The Netherlands Phone: + 31 10 408 1182 Fax: + 31 10 408 9640 Email: info@erim.eur.nl Internet: www.erim.eur.nl

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT

REPORT SERIES
RESEARCH IN MANAGEMENT

ABSTRACT AND KEYWORDS	
Abstract	Traditional rolling stock rescheduling applications either treat passengers as static objects whose influence on the system is unchanged in a disrupted situation, or they treat passenger behavior as a given input. In case of disruptions however, we may expect the flow of passengers to change significantly. In this paper we present a model for passenger flows during disruptions and we describe an iterative heuristic for optimizing the rolling stock to the disrupted passenger flows. The model is tested on realistic problem instances of NS, the major operator of passenger trains in the Netherlands.
Free Keywords	passenger railways, disruption management, passenger flows, rolling stock
Availability	The ERIM Report Series is distributed through the following platforms: Academic Repository at Erasmus University (DEAR), DEAR ERIM Series Portal Social Science Research Network (SSRN), SSRN ERIM Series Webpage Research Papers in Economics (REPEC), REPEC ERIM Series Webpage
Classifications	The electronic versions of the papers in the ERIM report Series contain bibliographic metadata by the following classification systems: Library of Congress Classification, (LCC) LCC Webpage Journal of Economic Literature, (JEL), JEL Webpage ACM Computing Classification System CCS Webpage Inspec Classification scheme (ICS), ICS Webpage

Rescheduling of railway rolling stock with dynamic passenger flows

L. Kroon, G. Maróti and L. Nielsen
Rotterdam School of Management
Erasmus University of Rotterdam
P.O.Box 1738, 3000 DR, Rotterdam, The Netherlands
E-mail: {lkroon,gmaroti,lnielsen}@rsm.nl

December 10, 2010

Abstract

Traditional rolling stock rescheduling applications either treat passengers as static objects whose influence on the system is unchanged in a disrupted situation, or they treat passenger behavior as a given input. In case of disruptions however, we may expect the flow of passengers to change significantly.

In this paper we present a model for passenger flows during disruptions and we describe an iterative heuristic for optimizing the rolling stock to the disrupted passenger flows. The model is tested on realistic problem instances of NS, the major operator of passenger trains in the Netherlands.

Keywords: passenger railways, disruption management, passenger flows, rolling stock.

1 Introduction

Most rolling stock rescheduling applications in the literature either treat passengers as static objects whose influence on the system is unchanged in a disrupted situation, or they treat passenger behavior as a given input.

There are indeed situations where these assumptions on passengers are appropriate. When the changes to the system caused by a disruption are only light or moderate, we may assume that passenger demand is unchanged. Additionally, if suitable historical data is available we may assume that passenger behavior is given as input. In other situations, however, we may expect the flow of passengers to change significantly.

The changed demand for capacity may be alleviated by rescheduling the rolling stock, thus transferring capacity from low demand trains to high demand trains. Any attempt to reschedule the rolling stock implies a balance between the rescheduling effort and the corresponding service level. In this paper we present a model for passenger flows during disruptions and we describe an iterative heuristic for optimizing the rolling stock to the disrupted passenger flows.

2 Models for passenger flows in various logistic settings

The dynamics of passenger flows have been studied in several logistic settings such as railways, airlines and transit networks, and with different focus such as line planning, revenue management and disruption management. We here list a number of references to studies in the literature, discuss their methods and assumptions, and compare them to the problem studied in this paper.

Bratu and Barnhart (2006) study disruption management at a major airline company. They present MIP models that incorporate decisions on aircraft, crew and passenger recovery with the options of postponing or canceling flight legs. The objective is to simultaneously minimize operating costs, estimated passenger delay and disruption costs. The operator is assumed to have full control of the passenger flow in the network and can thus decide how passengers are matched with available capacity.

The assumptions on operator control of assignment of passengers may be realistic in some railway settings. Consider a railway system where a seat reservation is required to board a train. In that case the operator does have control of the matching of passengers to trains. This contrasts the situation in the railway network of NS where the operator decides how to assign capacity to the timetable, but passengers decide how to utilize the available capacity.

Dumas and Soumis (2008) present a model for the passenger flow in airline networks given data on the demand between pairs of origin and destination and the temporal distribution of bookings. Furthermore, the authors assume knowledge on the spill of passengers between itineraries, i.e. if a booking for a certain set of flights is rejected they know the proportion of passengers who would attempt to book a certain alternative set of flights.

Dumas et al. (2009) use the above passenger flow model for the fleet assignment problem. This is the problem of assigning a fleet of aircraft to a set of flights while maximizing expected revenue. In earlier models for the fleet assignment problem fixed demands for itineraries are assumed given (see Hane et al. (1995) and Abara (1989)), and the problem can be expressed by a large MIP model. But since Dumas et al. (2009) use a complex nonlinear passenger flow model, they separate the fleet assignment decisions and the computation of the revenue from the resulting passenger flow. The computations are performed in an iterative approach where in each iteration a fleet assignment is computed and the passenger flow model is used to evaluate the quality of the current assignment and estimate the impact of changing a decision in the next iteration. The iterative approach presented by Dumas et al. (2009) is used as an inspiration for the solution approach presented in this paper. We do, however, utilize another model for the passenger flow and the context is significantly different; revenue management in an airline versus disruption management in a railway network.

Passenger flow modeling in urban transit networks is a well studied field. Oppenheim (1995) gives a comprehensive overview and discusses the behavioral approach in which travelers are assumed to make travel choices which are "best" for them. In this approach passengers can be aggregated by temporal and spatial origin and destination. The simplifications obtained through this aggregation comes at the cost of information on the individual's traveling choices.

The delay management problem considers the decision of deliberately delaying the departure of a vehicle to allow passengers on delayed arriving vehicles to maintain their connections. The problem is related to the problem studied in this paper as both problems share the goal of minimizing passenger delays. However, vehicle capacities are generally not considered in the models in literature. We refer to Ginkel and Schöbel (2007), Giovanni et al. (2007), and Schachtebeck (2010) for an overview of delay management.

Passenger flows are also considered in the strategic railway planning phase when the line planning is conducted. In this setting the passenger flow is considered aggregated by origin/destination pairs over the entire day. Line planning then amounts to determining service lines that balance operational cost and service according to the expected passenger flows. A number of references to literature on line planning can be found in Nielsen (2011).

3 Framework for rolling stock rescheduling with dynamic passenger flows

In this paper we investigate a certain class of disruptions where passenger behavior may influence the performance of the system considerably. The problem arises from two observations. The first observation is that in case of a blockage in the railway network passengers will attempt to get to their destinations by alternative traveling routes. The second observation is that passengers who change their routes lead to an altered demand for capacity on the trains serving the alternative routes. In fact, the demand for capacity on the alternative routes may be so great that the trains cannot accommodate all passengers leading to further delays and instability of the system.

The problem of rescheduling the rolling stock while accounting for the dynamics of passenger flows can be stated as the following generic optimiza-

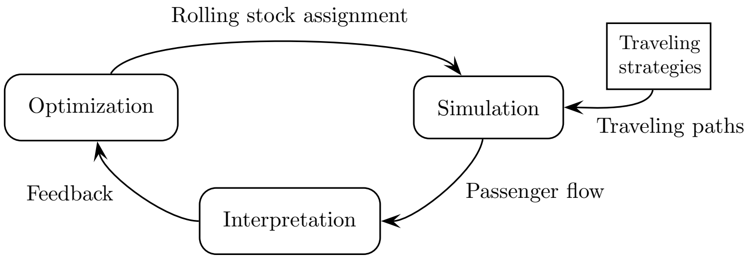


Figure 1: Iterative procedure for solving the rolling stock rescheduling problem with dynamic passenger flows.

tion problem.

$$\min c(x) + d(y)$$

subject to

$$x \in \mathcal{X}$$
$$y = f(x) \in \mathcal{Y}$$

where $\mathcal{X}$ is the set of rolling stock assignments to the timetable and $\mathcal{Y}$ is the set of feasible passenger flows. The function $f : \mathcal{X} \to \mathcal{Y}$ returns the expected passenger flow for a given assignment of rolling stock $x \in \mathcal{X}$. The cost function consists of two terms; the function $c : \mathcal{X} \to \mathbb{R}$ that gives the system related cost of a rolling stock assignment, and the function $d : \mathcal{Y} \to \mathbb{R}$ that gives the service related cost of a passenger flow.

We suggest an approach for solving the above model by iteratively rescheduling the rolling stock, simulating the passenger flows, and interpreting the flows to give an optimization direction for the next iteration. The approach is sketched in Figure 1. The framework is fully modular, which means it is possible to exchange any component in the procedure if we want to test a different set of underlying assumptions on the system.

Given a feasible rolling stock assignment, the expected passenger flow is computed by means of simulation. We introduce a model for passengers based on a multi commodity flow in an intuitive graph in Section 4. The model of passenger behavior follows a set of assumptions on passenger behavior. The assumptions relate to the traveling strategy applied by the passengers and to the interaction of the passengers when competing for capacity in the trains. We describe a simulation algorithm that implements this set of assumptions on passenger behavior in Section 5.

Based on the expected passenger flows computed in the simulation step, a feedback mechanism creates an optimization function for the next iteration. This function penalizes the assignment of rolling stock with too little capacity compared to the demand specified by the expected passenger flows. In Section 6 we discuss the design of such a feedback mechanism.

The optimization step involves assigning the rolling stock to the trains while taking the system related costs into account as well as the service related costs given by the feedback mechanism. Chapter 3 of Nielsen (2011) describes in detail how to adapt the Composition Model to this purpose.

Starting the iterative approach

We start the iterative procedure by performing the optimization step. Since no feedback from earlier iterations is available we only use the goals related to the system as objective in this first step. Based on the resulting assignment of capacity we perform the simulation step and compute the feedback for the next round of optimization.

Stopping criterion

We do not claim any guarantee on the performance of the heuristic iterative approach and there is therefore no natural stopping criterion. However, since the approach may be used in a time critical environment we could terminate the process when a time limit is reached or perform a fixed number of iterations. Alternatively, we could continue until no improvement is found during a number of iterations.

In our tests we limit ourselves to the simple stopping criterion of using a fixed number of iterations. In this way an acceptable number of solutions are visited within reasonable computation time.

4 Modeling the passenger flow

The model for passenger behavior relies on the following assumptions on individual passengers. A passenger enters the railway system at a specific time and wants to travel from an origin station to a destination station. If the passenger does not reach his destination within a certain time interval, we assume the passenger leaves the system and either gives up the travel or pursues another mode of transportation. We call the last time instant at which a passenger will accept to arrive the *deadline* of the passenger. Furthermore, each passenger has a *traveling strategy* for how to travel in the network. This strategy decides which trains he attempts to board given the available information on the state of the system.

The deadline represents the time at which a passenger leaves the system to either give up the intended travel or find a different mode of transportation. The deadline thereby implies that passengers are not willing to wait endlessly to get to their destination. When a passenger leaves the system due to the deadline it represents both a direct monetary cost for the operator as the passenger may be eligible for compensation, and an indirect monetary cost through the loss of goodwill.

Rather than modeling every single passenger in the system, we aggregate passengers with the same characteristics into *passenger groups*. Let $\mathcal{P}$ be the set of passenger groups. A passenger group $p \in \mathcal{P}$ has size n_p, and enters the system at time τ_p at origin station o_p. The group has destination station d_p and deadline $\tilde{t}_p$. The group uses traveling strategy S_p which is described in further detail in Section 4.2.

4.1 The passenger graph

The passengers travel in the time expanded graph defined by the timetable. We define the *passenger graph*, $G = (V, A)$, in the following way. Let $\mathcal{S}$ be the set of stations and let $\mathcal{T}$ be the set of trips. We add a node to the graph for each departure or arrival of a train at a station. Then a node denotes a station $s \in \mathcal{S}$ at a specific time τ. The set of nodes is thus defined as the set V:

$$V = \{(s, \tau) \mid \text{a train departs from or arrives at station } s \in \mathcal{S} \text{ at time } \tau\}$$

The set of arcs consists of two kinds of arcs, the *trip arcs* and the *time arcs*. A trip arc denotes a train traveling from one station to another whereas a time arc exists between every pair of consecutive nodes at the stations to denote waiting time at the station:

$$\begin{aligned} A = &\{(u, v) \in V \times V \mid \text{a train departs at time } \tau \text{ from station } s \text{ where} \\ &\quad u = (s, \tau) \text{ and arrives at station } s' \text{ at time } \tau' \text{ where } v = (s', \tau')\big\} \\ &\cup \big\{(u, v) \in V \times V \mid u = (s, \tau), v = (s, \tau') \text{ where there does not} \\ &\quad \text{exist a node } w = (s, \tilde{\tau}) \text{ with } \tau < \tilde{\tau} < \tau'\big\} \end{aligned}$$

The trains have a limited capacity depending on the rolling stock assigned to them. We define the capacity of a trip arc as the maximum number of passengers the train can accommodate according to the safety regulations. The capacity of a time arc is set to be infinite.

4.2 Traveling strategy

The traveling of a passenger constitutes a path in the passenger graph G. The traveling strategy S_p of a passenger group $p \in \mathcal{P}$ determines how the passengers in the group want to travel. It is a function $S_p : V \to \mathtt{Paths}(G)$ from the nodes to the set $\mathtt{Paths}(G)$ of directed paths in G. The path given by $S_p(v)$ denotes the preferred traveling path in the network when the group p is situated at node v. The definition of the traveling strategy S_p allows the passenger group to dynamically change its path in the event of a disruption i.e. a change in the structure of the graph. Note that S_p could be generalized to return a distribution of the passengers on a set of paths. But we limit this study to the special case where the traveling strategy implies that all passengers in the group prefer the same path.

The trip arcs in the graph have finite capacity so it may not be possible for all passengers in a group to travel with the same arc. This causes the passenger groups to interact when boarding the train as they compete for the scarce capacity. Once a passenger is in a train he occupies the capacity until he leaves the train and does therefore not have to compete with other passengers who attempt to board the train later. The assumptions on the interaction of the passengers are discussed in Section 5.1. Due to the interaction, the traveling of the passengers in a group p implies a network flow in the passenger graph rather than just a path. The flow originates from a source node $(o_p, \tau_p) \in V$ and flows to a set of nodes later in time. The combination of the flows of all the passenger groups constitutes a multi commodity flow in the passenger graph.

4.3 The quality of a passenger flow

The quality of the passenger flow is measured by several criteria;

- Whether trains are overcrowded.
- Whether passengers are delayed compared to their expected arrival time.
- Whether passengers arrive at all at their destination station within their set deadline.

In this study we limit ourselves to the two last criteria i.e. delays and arrival within the deadline. More specifically, we define the *inconvenience* of a passenger as the number of minutes of delay experienced by the passenger plus a penalty for reaching the deadline. This penalty may then depend on the time of day, and whether the passenger can reach the destination by other means of transportation. In our experiments we penalize a passenger leaving the system by the number of minutes between the expected arrival time with the intended traveling path and the deadline. We define the overall service objective as the sum of the inconvenience experienced by each passenger.

5 Simulating the passenger flow

We designed a deterministic simulation algorithm to calculate the expected passenger flow. In this section we account for our assumptions on passenger behavior and then we present the simulation algorithm.

5.1 Assumptions

For the simulation of passenger behavior, we make assumptions on three key elements: (*i*) What information is available to the passengers? (*ii*) Which traveling strategy do passengers apply and how do they use the available information? (*iii*) How do passengers interact? We here discuss the assumptions we applied to these aspects and their implications on the practical applicability of our approach. We emphasize that the approach is modular which allows us to replace the strategy and interaction rules by any other set of assumptions.

Information

We assume that passengers know the timetable i.e. they know the departure and arrival times of all trains. In addition they know, at the occurrence of a disruption, which trains are canceled. Note that passengers do not know anything about cancellations before the disruption occurs. Furthermore, they do not know anything about the utilization of the capacity of the trains in the network either, and therefore they do not have any knowledge of whether they are able to board trains on their traveling path.

The assumption that all passengers know all departure and arrival times of all trains is a reasonable assumption since they are published in the timetable. However, when a disruption occurs there may be some uncertainty about the exact departure times of trains affected by the disruption. In this case it may be overly optimistic to assume knowledge of the exact process times. The assumption that passengers do not have any knowledge about the availability of capacity in the trains before attempting to board is realistic in a railway system where passengers cannot reserve capacity beforehand, and capacity is used on a first come first serve basis.

Strategy

We assume that the passengers in group p want to get to their destination as fast as possible. This means that they prefer traveling via a path in the passenger graph from the node (o_p, τ_p) to a node (d_p, τ) with smallest possible τ. If several such paths exist, the passengers prefer the path that involves the smallest number of transfers from one train to another. And again, if several such paths exist the passengers prefer the path with the earliest departure time.

We illustrate the assumptions by an example in Figure 2. The passenger graph is based on five trips $t_1, \dots, t_5$ from station O to station D of which trips t_1, t_2 and t_4 have half an hour of traveling time whereas trips t_3 and t_5 take only 20 minutes. The nodes are named (s, τ) representing station s at time τ. Trip arcs are represented by solid arcs and time arcs are represented by dashed arcs.

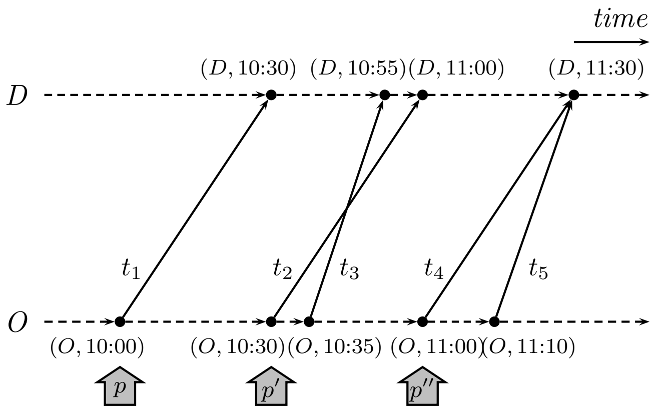


Figure 2: Part of a passenger graph with four trips from station O to station D. The gray arrows below the graph show when the three example passenger groups enter the system.

A passenger group p traveling from origin $o_p = O$ to destination $d_p = D$ starting at time $\tau_p = 10{:}00$ will thus prefer traveling with trip arc t_1. A passenger group p' also traveling from origin $o_{p'} = O$ to destination $d_{p'} = D$ starting at time $\tau_{p'} = 10{:}30$ will prefer waiting on time arc $(O, 10{:}30)(O, 10{:}35)$ and then travel with trip arc t_3 rather than travel with t_2. On the other hand, a passenger group p'' also traveling from traveling $o_{p''} = O$ to destination $d_{p''} = D$ starting at time $\tau_{p''} = 11{:}00$ will travel with trip arc t_4 rather than wait for t_5 as they have the same arrival time.

It is also part of the traveling strategy of a passenger group to react to the possibility that they may not reach their destination within the deadline. This is achieved by assuming that if a passenger group p is about to board a train that arrives after the deadline $\tilde{t}_p$, the passenger group will choose to leave the system rather than board the train. This behavior represents a passenger group choosing to either give up their intended travel or finding other means of transportation.

The assumed traveling strategy does not apply to all passengers in practice, some passengers may indeed choose for a later arrival time if it incurs fewer transfers, but in general passengers are expected to go for the earlier arrival time.

For the passengers leaving the system when they cannot reach their destination within their deadline, we recognize that passengers may have very different behaviors on when and where to leave the system. However, we would not expect passengers to stay in the system forever either. Modeling the behavior of leaving the system through a deadline thus enables us to track passengers who are severely affected by the state of the railway system and thereby represent a significant loss of goodwill. Note that the introduction of a deadline is not a limitation to the modeling power of the approach since, if the deadline is set sufficiently high, passengers will never leave the system at other stations than their destination.

Interaction

Passengers interact when attempting to board a train, in the sense that they compete for the limited capacity available in the train. When more passengers attempt to board a train than the available capacity allows for, only a portion of the passengers will enter the train. We assume that the number of passengers from each group who board a train is relative to the size of the group. Suppose passenger groups $p_1, \ldots, p_k$ attempt to board a trip arc a at node v with capacity c. Group p_i has size n_{p_i} and the combined size of the passenger groups is thus

$$n = \sum_{i=1}^{k} n_{p_i}$$

If $n > c$ then $f_{p_i} = c \cdot n_{p_i}/n$ passengers from group p_i board the train. The number of people from a group boarding a train may be fractional but that is not problematic since the train capacities are several hundreds and the contribution of fractional flows are therefore neglectable. Therefore we choose not to implement any tie breaking rule.

If not all passengers from group p_i are able to board a departing train then the boarded passengers as well as the rejected passengers will continue their journey according to their traveling strategy. The boarded passengers will constitute a flow of value f_{p_i} on arc a whereas the remaining $n_{p_i} - f_{p_i}$ passengers from the group will stay at node v and attempt to travel to d_{p_i} according to their strategy S_{p_i}. When some passengers from a passenger group p_i are rejected for boarding it is equivalent to splitting the group into two groups with the same characteristics except that one is situated at the arrival node of the arc with size f_{p_i} and the other at the departure node with size $n_{p_i} - f_{p_i}$.

Passengers who are already in the train and who wish to travel further do not participate in the above mentioned boarding procedure. This implies that when a train arrives at a station some of its capacity may already be in use.

Train capacities

In addition to the above assumptions on passenger behavior, we make one simplifying assumption on the train capacities. We assume that train capacities are only adapted at terminal stations and never underway. This implies that in any rolling stock assignment a train always has the same capacity on two consecutive trips. This further implies that there is enough capacity in a departing train to accommodate passengers who are already in the train and wish to travel further with the same train.

We note that this assumption is not a limitation of the approach since we can model trains with underway capacity adaptation by introducing parallel arcs in the passenger graph: Suppose a train uncouples a unit at an intermediate station, then parallel arcs can be used until that station. One set of arcs has capacity corresponding to the uncoupled unit and the other set of arcs has capacity corresponding to the remaining part of the train.

5.2 Simulation algorithm

We here describe a simulation algorithm that incorporates the above assumptions on traveling strategies and interactions between passengers. To the best of our knowledge there is no simulation algorithm in literature that allow us to incorporate exactly these assumptions.

Suppose the set of passenger groups is given by $\mathcal{P}$ and we are given a passenger graph $G = (V, A)$. For the purpose of simulating the passenger flow under the above assumptions we introduce a four-tuple (p, v, n, a) called a *container*. A container denotes n passengers from passenger group p that are positioned at node v and last traveled by trip arc a. The arc a thus indicates that the passengers are already in the train that performs a and do not participate in the boarding procedure if they wish to continue with the same train. If the passengers have not yet traveled with any arc, the arc a in the container is set to a dummy value ϕ.

Algorithm 1 generates the passenger flow under the given assumptions. In short, the algorithm works by performing a time sweep over all trip arcs in the passenger graph and moving the containers through the graph according to the assumptions on traveling strategies and interaction. The algorithm returns a function $f : A \times \mathcal{P} \to \mathbb{R}$ that maps the arcs and passenger groups to the size of the flow of the particular group on a certain arc.

In the preprocessing in lines $1 - 3$ a set S of containers is initialized to hold each passenger group at its origin station at the time it enters the network. The set of arcs is then ordered by departure time. If several arcs depart at the same time their ordering is arbitrary. The function f is set to 0 for all pairs $(a, p) \in A \times \mathcal{P}$.

In line 4 we iterate over all arcs to apply the boarding procedure at each departure. In line 5 we denote the departure and arrival nodes of the arc a by (s, τ) and (r, σ), and in line 6 we denote the predecessor arc of a by $a_{\texttt{pred}}$. The predecessor arc is serviced by the train immediately before the trip arc a. This notion is necessary to determine which passengers are already in the train as they do not have to participate in the boarding procedure. Then the subset $S' \subseteq S$ is extracted holding the containers $g = (p, (s, \tau), n, b)$ that are situated at the departure node (s, τ) and want to travel with the trip in question. Note that this is determined by the traveling strategy S_p of the

```
   Input: Passenger graph $G = (V, A)$, set of passenger groups $\mathcal{P}$
   Output: Flow function $f : A \times \mathcal{P} \to \mathbb{R}$
1  $S := \{(p, (o_p, \tau_p), n_p, \phi) \mid p \in \mathcal{P}\}$
2  $L :=$ An ordering of $A$ by departure time
3  Let $f(a, p) := 0$ for all $a \in A$, $p \in P$
4  foreach $a \in L$ do
5      Let $(s, \tau)$ and $(r, \sigma)$ be the departure and arrival nodes of $a$
       respectively
6      Let $a_{\texttt{pred}}$ be the predecessor of $a$
7      $S' = \{g = (p, (s, \tau), n, b) \in S \mid g$ wants to travel with $a\}$
8      $S'' = \{g = (p, (s, \tau), n, b) \in S' \mid g$ is already in the train (i.e.
       $b = a_{\texttt{pred}})\}$
9      $c := \texttt{cap}(a) - \sum_{g \in S''} n(g)$
10     foreach $g \in S'$ do
11         if $b = a_{\texttt{pred}}$ then
12             $u := n(g)$
13         else
14             $u := \min\{n(g), c \cdot n(g) / \sum_{g' \in S' \setminus S''} n(g')\}$
15         $S := S \setminus \{g\}$
16         if $u < n$ then
17             Let $(s, \tau')$ be the next departure node at station $s$
18             $S := S \cup \{(p(g), (s, \tau'), n - u, \phi)\}$
19         if $r \neq d_p$ then
20             $S := S \cup \{(p(g), (r, \sigma), u, a)\}$
21         $f(a, p) := f(a, p) + u$
22 return $f$
```

Algorithm 1: Simulation algorithm for the passenger flow.

passengers in group p. Further, S'' is the subset of S' that is already in the train before it departs.

Next, in line 9, the free capacity of the arc a is calculated by subtracting from the total capacity $\texttt{cap}(a)$ the capacity taken up by passengers already in the train. Passengers already in the train are identified by having last traveled by arc $a_{\texttt{pred}}$. Note that the remaining capacity c is always non-negative as the capacity of the train is unchanged according to our assumptions, i.e. $\texttt{cap}(a_{\texttt{pred}}) = \texttt{cap}(a)$.

In the loop starting from line 10 each relevant container $g \in S'$ is handled. First the number of passengers u who obtain capacity on the train is calculated. If the passengers are not in the train (line 11) they participate in the boarding procedure (line 14) or else they all fit in the train (line 12).

The container is then removed from the set S, and replaced by new containers as follows. If not all passengers are assigned to the arc (i.e. $u < n$) then the $n - u$ rejected passengers from the group remain at the station at node (s, τ') defined as the departure node of the next arc departing from this station (lines 16 – 18). The number of passengers u that were accepted on a will reappear at the arrival node (r, σ) unless it is the destination station of the passengers (lines 19 – 20). Finally the flow function is updated in line 21.

Example

We return to the example in Figure 2. Let each of the trains assigned to the five trips have a capacity of 100, and let $\mathcal{P} = \{p_1, p_2\}$ consist of two passenger groups with origin $o_{p_1} = o_{p_2} = O$, destination $d_{p_1} = d_{p_2} = D$, size $n_{p_1} = n_{p_2} = 100$, and deadline sufficiently large. The groups enter the system at time $\tau_{p_1} = 10{:}30$ and $\tau_{p_2} = 10{:}35$ respectively. According to the traveling strategies S_{p_1} and S_{p_2} both groups prefer traveling by the path in the passenger graph that uses trip t_3. In the simulation the containers are initialized to

$$S = \{(p_1, (O, 10{:}30), 100, \phi), (p_2, (O, 10{:}35), 100, \phi)\}$$

and after processing the outgoing arcs of node $(O, 10{:}30)$ the set contains

$$S = \{(p_1, (O, 10{:}35), 100, \phi), (p_2, (O, 10{:}35), 100, \phi)\}.$$

When processing the arc of trip t_3 both groups will attempt to board the train and according to the assumptions on the boarding procedure 50 passengers from each group will be assigned to the arc thus setting $f(t_3, p_1) = f(t_3, p_2) = 50$. New containers are inserted at the next relevant node thus resulting in the following set of containers:

$$S = \{(p_1, (O, 11{:}00), 50, \phi), (p_2, (O, 11{:}00), 50, \phi)\}.$$

When processing the arc of trip t_4 the remaining passengers will be able to board, thus setting $f(t_4, p_1) = f(t_4, p_2) = 50$.

We can calculate the delays of the involved passenger groups using the flow f. In the above example 50 passengers from each of the groups p_1 and p_2 suffered a delay of 35 minutes compared to their initially intended journeys which results in a total delay of $2 \cdot 50 \cdot 35 = 3500$ delay minutes. We observe that the scarcity of capacity in the trains combined with the greedy traveling strategy of passengers may lead to delays. In our experiments we choose to penalize delay minutes uniformly although one may argue that longer delays are worse than several small delays.

5.3 Implementation issues

In addition to the generic assumptions on passenger behavior implemented in the simulation algorithm 1 there are several other issues to be taken into account. Those are primarily related to assumptions on special cases and implementation issues.

The traveling strategy of the passengers is implemented as a shortest path algorithm in the passenger graph. The shortest path search can be performed in linear time since the passenger graph is acyclic. In the implementation we store the desired traveling path with the container to avoid recalculating the path several times. Only when passengers are rejected in the boarding procedure we need to recompute their path.

The occurrence of a disruption is incorporated in the simulation algorithm by changing the passenger graph at the appropriate point in the simulation. All containers in the set S then have their preferred path recomputed.

As the simulation algorithm is stated, it returns a function $f : A \times \mathcal{P} \to \mathbb{R}$ denoting the number of passengers from each group traveling by each arc. However, it is necessary for the feedback mechanism described in Section 6 to have the path decomposition of the flows rather than the flows represented by f. One can derive a greedy path decomposition from f in time $O(|\mathcal{P}| \cdot |V| \cdot |A|)$, but we need the path decomposition that represents the actual paths traveled by passengers. We can create it on the fly by storing the path history with each container in the algorithm instead.

We remark for the computational complexity of the simulation algorithm, that up to two containers are added to S whenever one container is removed. This implies that in the worst case each passenger group is split a number of times that is exponential in the number of trips. It is trivial to construct instances that demonstrate asymptotic worst case behavior, but as we observe later, the computational behavior of the simulation on realistic instances is not a bottle-neck in the process.

We note that we could potentially obtain a speedup of the algorithm by merging containers that represent the same passenger group if they meet at a common node. This would even prevent the worst case exponential complexity. However, we would then loose the history of paths traveled thus far in the simulation and we would loose information on the traveling patterns of groups that were split.

6 Feedback

The purpose of the feedback mechanism is to interpret the passenger flow returned by the simulation algorithm, and provide an optimization direction that is likely to improve the solution in the next iteration. This is performed by estimating the effect of the train capacities on the total passenger inconvenience.

More formally, the feedback mechanism is a function $F : \mathcal{T} \times \mathbb{Z}_+ \to \mathbb{R}$ that maps the trips $\mathcal{T}$ in the timetable and the non-negative integers to values in $\mathbb{R}$. For trip $t \in \mathcal{T}$ and $c \in \mathbb{Z}_+$ the value $F(t, c)$ denotes the penalty to the potential decision of assigning rolling stock with capacity c to trip t.

Consider a passenger flow $f : A \times \mathcal{P} \to \mathbb{R}$ returned by the simulation algorithm in Section 5. Then consider a trip $t \in \mathcal{T}$ and let $a \in A$ be the trip arc representing trip t. We achieve this by considering how many passengers wanted to travel with t in the simulation and estimating the delay of any passengers who were unable to board the train at its departure.

From the simulation algorithm we know which passenger groups attempted to travel with arc a. Let $p_1, \ldots, p_k$ be those groups. Let $u_i = f(a, p_i)$ be the number of passengers from group p_i that traveled with arc a and let r_i be the number of passengers from group p_i that attempted to board arc a but were rejected.

Consider a passenger group p_i. The passengers of this group follow paths from the origin node o_{p_i} according to the path decomposition from the simulation algorithm. Suppose s_i paths contain the arc a, and t_i paths result from the rejection at the boarding procedure at arc a. Then let $Q_1, \ldots, Q_{s_i}$ be the paths that contain the arc a and let $W_1, \ldots, W_{t_i}$ be the paths that result from the rejection at a. For a path P let $n(P)$ be the number of passengers that traveled along P and let $v(P)$ be the inconvenience per passenger traveling along P for the group p_i. Then the term

$$\texttt{TRAVEL}(p_i, a) = \frac{\sum_{j=1}^{s_i} (n(Q_j) \cdot v(Q_j))}{\sum_{j=1}^{s_i} n(Q_j)}$$

denotes the average inconvenience per passenger from group p_i traveling on a. Similarly, the term

$$\texttt{REJECT}(p_i, a) = \frac{\sum_{j=1}^{t_i} (n(W_j) \cdot v(W_j))}{\sum_{j=1}^{t_i} n(W_j)}$$

denotes the average inconvenience per passenger from group p_i rejected upon boarding a. Using the terms $\texttt{TRAVEL}(p_i, a)$ and $\texttt{REJECT}(p_i, a)$ we estimate the contribution per passenger to the total inconvenience by

$$\texttt{AVG\_COST}(a) = \frac{\sum_{i=1}^{k} r_i \cdot (\texttt{REJECT}(p_i, a) - \texttt{TRAVEL}(p_i, a))}{\sum_{i=1}^{k} r_i}$$

where $\texttt{AVG\_COST}(a)$ is the marginal contribution to the cost of the passenger flow by decreasing the capacity of arc a by a small value ϵ. We set the function in the feedback mechanism to the following value

$$F(t, c) = \max \left\{ 0\, ,\ \sum_{i=1}^{k} (u_i + r_i) - c \right\} \cdot \texttt{AVG\_COST}(a)$$

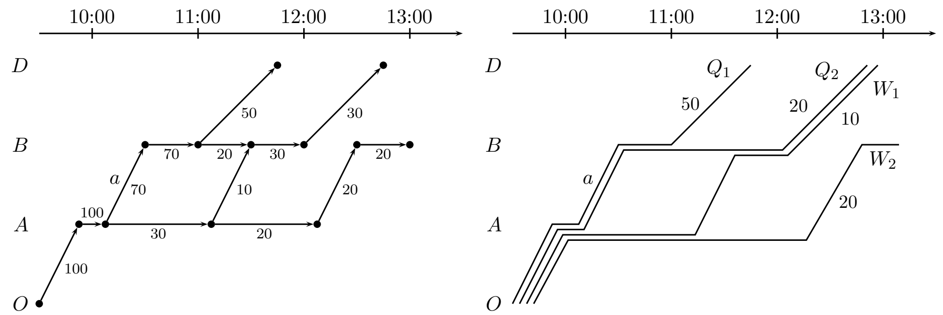


Figure 3: *Left*: The passenger flow on arcs for passenger group p_i. *Right*: The path decomposition.

which denotes the expected cost of assigning capacity c to trip t (where a is the trip arc corresponding to t). Note that if the capacity is sufficient to hold all passengers (i.e. $c \geq \sum_{i=1}^{k} (u_i + r_i)$) then it incurs no cost to assign the capacity.

Example

Consider the example passenger flow of a group p in the left diagram of Figure 3. The group consists of 100 passengers traveling from origin O to destination D, starting at 9:00, and has deadline 13:15. Upon boarding arc $a = (A, 10{:}10)(B, 10{:}30)$ the passenger group is split where 70 passengers are able to board and the remaining 30 passengers are rejected. The group is further split at some other arcs resulting in the paths shown in the right diagram in Figure 3. Paths Q_1 and Q_2 follow from the boarding of a whereas paths W_1 and W_2 follow from the rejection at a.

Assume the passengers in group p had expected arrival time 11:45. Then we derive the following path sizes and values from the figure: $n(P_1) = 50$, $v(P_1) = 0$, $n(P_2) = 20$, $v(P_2) = 60$, $n(W_1) = 10$, $v(W_1) = 60$, $n(W_2) = 20$ and $v(W_2) = 90$. The average contribution per passenger to the total inconvenience for passengers from group p respectively traveling with or rejected at a is calculated as follows.

$$\texttt{TRAVEL}(p, a) = \frac{\sum_{j=1}^{2} (n(Q_j) \cdot v(Q_j))}{\sum_{j=1}^{2} n(Q_j)} = \frac{50 \cdot 0 + 20 \cdot 60}{70} = 17.1$$

$$\texttt{REJECT}(p, a) = \frac{\sum_{j=1}^{2} (n(W_j) \cdot v(W_j))}{\sum_{j=1}^{2} n(W_j)} = \frac{10 \cdot 60 + 20 \cdot 90}{30} = 80$$

The estimated contribution to the quality of the passenger flow per passenger rejected at arc a is thus 62.9.

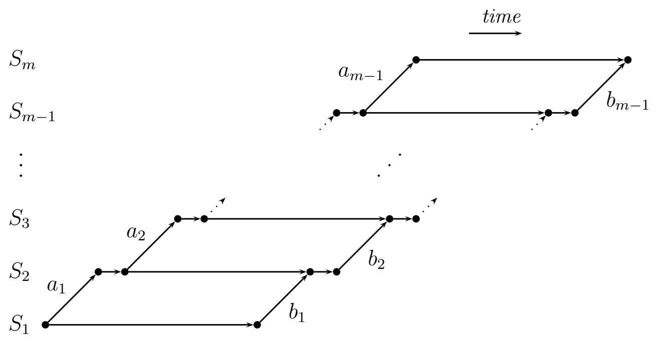


Figure 4: Example with passenger graph that needs $m-1$ iterations to avoid delay.

Observations and remarks

We note that the feedback mechanism may encourage the insertion of more capacity on an arc a resulting in more passengers traveling on that arc. However, these passengers may turn out to be delayed by the lack of capacity on other arcs instead. It may thus take several iterations before reaching an assignment of rolling stock that avoids some of the delays. Consider for example the passenger graph in Figure 4 with m stations $S_1, \ldots, S_m$. Two trip arcs a_i and b_i connect station S_i and S_{i+1} where b_i is later than a_i. Suppose a passenger group has origin S_1 and destination S_m and has some passengers rejected upon boarding at arc a_1. Those passengers will then travel with trip arcs $b_1, \ldots, b_{m-1}$.

The feedback mechanism will then penalize capacity shortage on arc a_1. Assigning more capacity to a_1 will then cause the passengers to be rejected at arc a_2 instead, leading to the same amount of delay. The feedback mechanism will then additionally penalize capacity shortage on arc a_2 in the next iteration and so on. In this example it will take $m-1$ iterations before the delay is avoided.

We also note that the feedback mechanism described in this section is a natural candidate for providing a new optimization direction based on the current solution. But there exist many other schemes for this task that could make sense.

7 Optimization

For rescheduling the rolling stock we use the Composition Model described in Chapter 3 of Nielsen (2011). Recall that the model contains variables $X_{t,p} \in \{0,1\}$ that denote whether composition $p \in \eta(t)$ is used for trip $t \in \mathcal{T}$ where $\eta(t)$ is the set of allowed compositions for t. The capacity of the rolling stock in composition p is $\texttt{cap}(p)$.

The objective of the model consists of two parts; the system related costs and the service related costs. At first we could define the service related costs in the ith iteration by the function $g_i(X)$ using the feedback mechanism $F : \mathcal{T} \times \mathbb{Z}_+ \to \mathbb{R}$.

$$g_i(X) = \sum_{t \in \mathcal{T}} \sum_{p \in \eta(t)} F(t, \mathtt{cap}(p)) \cdot X_{t,p}$$

Then the objective function $h_i(X, Z, I)$ in the ith iteration can be defined as follows.

$$h_i(X, Z, I) = c(X, Z, I) + g_i(X)$$

where $c(X, Z, I)$ are the system related costs and we assume that the feedback is initially $g_0(X) = 0$.

However, we experienced that this approach could lead to cyclical behavior as the feedback from earlier iterations is ignored. We therefore modify the part of the objective derived from the feedback to

$$g_i(X) = (1 - \alpha)g_{i-1}(X) + \alpha \sum_{t \in \mathcal{T}} \sum_{p \in \eta(t)} F(t, \mathtt{cap}(p)) \cdot X_{t,p}$$

where $0 \leq \alpha \leq 1$ is a parameter that weights the latest feedback against feedback from earlier iterations. This way the feedback from a certain iteration gradually becomes less relevant. Note that $\alpha = 1$ describes the special case where feedback from earlier iterations is ignored. The similar introduction of such an α parameter in the approach by Dumas and Soumis (2008) was shown to be crucial for the performance of their solution procedure.

We mention that neither the solution nor the g_i function necessarily converges with the applied feedback mechanism and the definition of α.

8 Lower bounds

To assess the quality of our solutions we investigate methods for constructing lower bounds on the solution value. This allows us to analyze how much of the delays is caused by the changes in the disrupted timetable and how much is caused by the shortage of capacity.

In this section we describe two methods based on relaxing some of the assumptions on the problem. The first approach is rather simple, we relax the constraints that cause the passengers to interact – namely the limitation on capacity of rolling stock. In the second approach we relax some assumptions on the passengers' traveling strategies; we assume that capacity is not utilized by passengers in a greedy manner, but is rather allocated by the operator such that total inconvenience in the system is minimized.

8.1 Lower bound based on unlimited capacity

An intuitive way to relax the underlying assumptions of the system on passenger interaction is to assume that all trains have unlimited capacity. This way no passenger is ever rejected in the boarding procedure and every passenger can follow the best path in the passenger graph according to their traveling strategy. This implies that all delays experienced by passengers are caused by changes to the timetable such as canceled trips rather than by limited capacity.

This relaxation corresponds to the situation where unlimited amounts of rolling stock are available, an arbitrary amount of rolling stock may be assigned to each train, and there are no limitations on shunting possibilities at any station. It results in a lower bound on the service related cost i.e. the total amount of delay experienced by passengers in the system. We call this lower bound the *infinite capacity lower bound* (IC-LB).

Calculating the infinite capacity lower bound simply amounts to simulating the passenger groups $\mathcal{P}$ in a passenger graph $G = (V, A)$ where all arcs $a \in A$ have capacity $\texttt{cap}(a) = \infty$.

8.2 Lower bound based on centralized passenger flow

As described earlier, the passenger flow constitutes a multi commodity flow in the passenger graph. The quality of the flow is measured by the total inconvenience experienced by the passengers, i.e. the amount of delay and the number of passengers who do not reach their destination before their deadline. Since the passengers themselves choose their routes in the network the total flow may be suboptimal with respect to this quality measure. Suppose we relax the assumptions on the route choice to let the operator decide which passengers travel with which trains.

For passenger group $p \in \mathcal{P}$ we denote by π_p the set of paths the passengers in the group can travel with. The paths originate from the node (o_p, τ_p) in the passenger graph $G = (V, A)$ and can lead to any station in the network – not only to the destination station. This includes an empty path that represents staying at the origin station.

Define for each passenger group $p \in \mathcal{P}$ and possible traveling path $q \in \pi_p$ a variable $Y_{p,q} \in \mathbb{R}_+$. The variable states the number of passengers from group p that travel with path q. Let $c_{p,q}$ denote the inconvenience by one passenger from group p traveling with path q. Then the passenger flow with operator control may be expressed as the following linear program.

$$\min \sum_{p \in \mathcal{P}} \sum_{\substack{q \in \pi_p: \\ a \in q}} c_{p,q} Y_{p,q} \tag{1}$$

subject to

$$\sum_{q \in \pi_p} Y_{p,q} = n_p \qquad \forall p \in \mathcal{P} \tag{2}$$

$$\sum_{p \in \mathcal{P}} \sum_{\substack{q \in \pi_p: \\ a \in q}} Y_{p,q} \leq \mathtt{cap}(a) \qquad \forall a \in A \tag{3}$$

$$Y_{p,q} \in \mathbb{R}_+ \qquad \forall p \in \mathcal{P}, q \in \pi_p \tag{4}$$

The objective function (1) consists of the sum over the paths of the number of passengers traveling with a possible path times the contribution per passenger of the path to the quality of the flow.

Constraints (2) state that all passengers in a group must travel by one of the possible paths. The capacity constraints (3) denote that the number of passengers traveling on an arc is limited by the arc capacity. Finally constraints (4) state the domains of the variables.

The program (1) – (4) models the operator controlled passenger flow problem as a continuous minimum cost multi commodity flow problem. However, the model assumes that capacities on arcs are given as input (right hand side of constraints (3)). The arc capacities are decided through the rolling stock rescheduling process and are themselves subject to constraints on rolling stock availability and shunting possibilities. In order to incorporate the aspect of rolling stock rescheduling, we extend the model to (5) – (9) below. Here, the variables $X_{t,p} \in \{0,1\}$ model the assignment of composition p to trip t as in the Composition Model in Chapter 3 of Nielsen (2011).

$$\min \sum_{p \in \mathcal{P}} \sum_{\substack{q \in \pi_p: \\ a \in q}} c_{p,q} Y_{p,q} \tag{5}$$

subject to

$$\sum_{q \in \pi_p} Y_{p,q} = n_p \qquad \forall p \in \mathcal{P} \tag{6}$$

$$\sum_{p \in \mathcal{P}} \sum_{\substack{q \in \pi_p: \\ a \in q}} Y_{p,q} \leq \sum_{p \in \eta(t)} \mathtt{cap}(p)\ X_{t,p} \qquad \forall a \in A, t \text{ is trip of } a \tag{7}$$

$$X \in \mathcal{X} \tag{8}$$

$$Y_{p,q} \in \mathbb{R}_+ \qquad \forall p \in \mathcal{P}, q \in \pi_p \tag{9}$$

In constraints (7) the right hand side describes that the capacity of an arc is determined by the rolling stock composition assigned to the trip represented by the arc. The constraint (8) states that the vector X of composition variables must belong to the set $\mathcal{X}$ of feasible assignments of rolling stock to trips. In the railway system considered in this paper, it is equivalent to saying that it must be a feasible solution to the Composition Model. We call this the *strong operator controlled lower bound* (SOC-LB). However, we did not solve this model. But in the discussion of future research in Section 10 we propose a method for solving SOC-LB using a cut-and-branch approach.

A somewhat weaker lower bound that does not require solving the complex model (5) – (9) can be constructed by relaxing constraints (8). Rather than requiring that all variables $X_{t,p}$ correspond to a feasible solution to the rolling stock rescheduling problem, we require for each trip $t \in \mathcal{T}$ to be assigned a composition $p \in \eta(t)$ with the largest possible capacity that is feasible in some solution $X \in \mathcal{X}$.

This enables us to decide the capacity in a preprocessing step and fix the variables $X_{t,p}$. With the capacities fixed the model is equivalent to the multi commodity flow model (1) – (4). We call this the *weak operator controlled lower bound* (WOC-LB).

We solve WOC-LB in two steps. In the first step, the Composition Model is solved once for each trip $t \in \mathcal{T}$ with the simple objective

$$\mathtt{cap}_{\mathtt{max}}(t) = \max \sum_{p \in \eta(t)} \mathtt{cap}(p) X_{t,p}$$

thus computing the maximum possible capacity $\mathtt{cap}_{\mathtt{max}}(t)$ on each trip t in any feasible assignment of the rolling stock. In the second step, we solve the model (1) – (4) with the arc capacities in constraint (3) set to $\mathtt{cap}(a) = \mathtt{cap}_{\mathtt{max}}(t)$ where a is the corresponding trip arc of t in the passenger graph. The multi commodity flow problem is solved using a text-book column generation procedure (see Ahuja et al. (1993)).

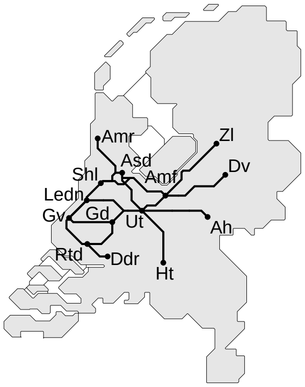
9 Computational tests

In this section we perform computational tests based on our approach. In Section 9.1 we describe how we generate test instances with different characteristics, and in Section 9.2 we report and discuss the results.

9.1 Instances

For testing our approach we constructed a number of instances from realistic data based on the Intercity network of NS. The instances involve the heavily utilized core part of the network connecting the 14 stations shown in Figure 5. This part of the network is serviced by the 16 Intercity lines listed in the figure. The lines call at the given stations and operate with the specified frequencies. On most routes there are at least four trains per hour between neighboring stations.

We note that some of the involved lines in reality continue beyond the terminal stations shown in Figure 5 i.e. to the south-west of Dordrecht (Ddr),



Line	Stations	Frequency
500	Gv Gd Ut Amf Zl	hourly
700	Shl Amf Zl	hourly
800	Amr Zd Asd Ut Ht	half hourly
1500	Asd Amf Dv	half hourly
1600	Shl Amf Dv	hourly
1700	Gv Gd Ut Amf Dv	hourly
1900	Gv Rtd Ddr	half hourly
2000	Gv Gd Ut Ah	half hourly
2100	Asd Shl Ledn Gv Rtd Ddr	half hourly
2600	Asd Shl Ledn Gv	half hourly
2800	Rtd Gd Ut Amf	half hourly
3000	Amr Asd Ut Ah	half hourly
3500	Shl Ut Ht	half hourly
8800	Ledn Ut	half hourly
20500	Rtd Gd Ut	hourly
21700	Rtd Gd Ut	hourly

Figure 5: The network considered in the test instances.

north of Alkmaar (Amr) and Zwolle (Zl), east of Deventer (Dv), south-east of 's-Hertogenbosch (Ht), and south of Arnhem (Ah). However, we perform a spatial aggregation of the network in the instances by assuming that passengers do not travel further than those six terminal stations. This is not a restriction in the verification of the approach as there are no rerouting possibilities in the peripheral parts of the network anyway.

The timetable is from a weekday and contains 2324 trips. All trips are assumed to be served with rolling stock of the type VIRM which is available in two variants with 4 and 6 carriages, respectively. The two variants have technical maximum capacities of 572 and 847 passengers per unit respectively. The maximum train length is assumed to be 14 carriages on all trips and it is possible for all lines to perform shunting operations at the terminal stations except Utrecht (Ut).

We created the passenger groups for the instances by matching passenger counts on each trip with OD data which resulted in 11415 passenger groups for the full day instance. To construct the deadlines of the passenger groups we assume that passengers are willing to accept at most an increase in traveling time of 50% plus 90 minutes.

Figure 6 shows the number of people in the system during the day in the instances. We observe that the peak hours around 8:00 and 17:00 are the busiest and therefore the most likely periods to experience capacity problems.

The disruptions considered in the computational tests all concern situations where a certain part of the network is unavailable for several hours. The timetable is updated according to current practice by canceling affected trips and turning trains on either side of the disruption.

The instances are described in Table 1. Each instance is named by a string like "D2T1O1R2P1" characterizing the instance. The name consists

			Assigned	Reserve		Passenger
Instance	Disruption	Time	units	units	Passengers	groups
D1T1O1R1P1	Rtd-Gvx	16:00 - 19:00	114, 44	0	422022	11415
D1T1O1R2P1	Rtd-Gvx	16:00 - 19:00	114, 44	1, 2	422022	11415
D1T1O1R3P1	Rtd-Gvx	16:00 - 19:00	114, 44	4, 2	422022	11415
D1T1O2R1P1	Rtd-Gvx	16:00 - 19:00	106, 41	0	422022	11415
D1T1O1R2P2	Rtd-Gvx	16:00 - 19:00	114, 44	1, 2	485375	11415
D1T2O1R1P1	Rtd-Gvx	11:00 - 15:00	114, 44	0	422022	11415
D2T1O1R1P1	Gd-Ut	16:00 - 19:00	114, 44	0	422022	11415
D2T1O1R2P1	Gd-Ut	16:00 - 19:00	114, 44	1, 2	422022	11415
D2T1O1R3P1	Gd-Ut	16:00 - 19:00	114, 44	4, 2	422022	11415
D2T1O2R1P1	Gd-Ut	16:00 - 19:00	106, 41	0	422022	11415
D2T1O1R2P2	Gd-Ut	16:00 - 19:00	114, 44	1, 2	485375	11415
D2T2O1R1P1	Gd-Ut	11:00 - 15:00	114, 44	0	422022	11415
D3T1O1R1P1	Ut-Amf	16:00 - 19:00	114, 44	0	422022	11415
D3T1O1R2P1	Ut-Amf	16:00 - 19:00	114, 44	1, 2	422022	11415
D3T1O1R3P1	Ut-Amf	16:00 - 19:00	114, 44	4, 2	422022	11415
D3T1O2R1P1	Ut-Amf	16:00 - 19:00	106, 41	0	422022	11415
D3T1O1R2P2	Ut-Amf	16:00 - 19:00	114, 44	1, 2	485375	11415
D3T2O1R1P1	Ut-Amf	11:00 - 15:00	114, 44	0	422022	11415
D4T1O1R1P1	Gvx-Ledn	16:00 - 19:00	114, 44	0	422022	11415
D4T1O1R2P1	Gvx-Ledn	16:00 - 19:00	114, 44	1, 2	422022	11415
D4T1O1R3P1	Gvx-Ledn	16:00 - 19:00	114, 44	4, 2	422022	11415
D4T1O2R1P1	Gvx-Ledn	16:00 - 19:00	106, 41	0	422022	11415
D4T1O1R2P2	Gvx-Ledn	16:00 - 19:00	114, 44	1, 2	485375	11415
D4T2O1R1P1	Gvx-Ledn	11:00 - 15:00	114, 44	0	422022	11415
D5T1O1R1P1	Asd-Ut	16:00 - 19:00	114, 44	0	422022	11415
D5T1O1R2P1	Asd-Ut	16:00 - 19:00	114, 44	1, 2	422022	11415
D5T1O1R3P1	Asd-Ut	16:00 - 19:00	114, 44	4, 2	422022	11415
D5T1O2R1P1	Asd-Ut	16:00 - 19:00	106, 41	0	422022	11415
D5T1O1R2P2	Asd-Ut	16:00 - 19:00	114, 44	1, 2	485375	11415
D5T2O1R1P1	Asd-Ut	11:00 - 15:00	114, 44	0	422022	11415

Table 1: Instances for computational results.

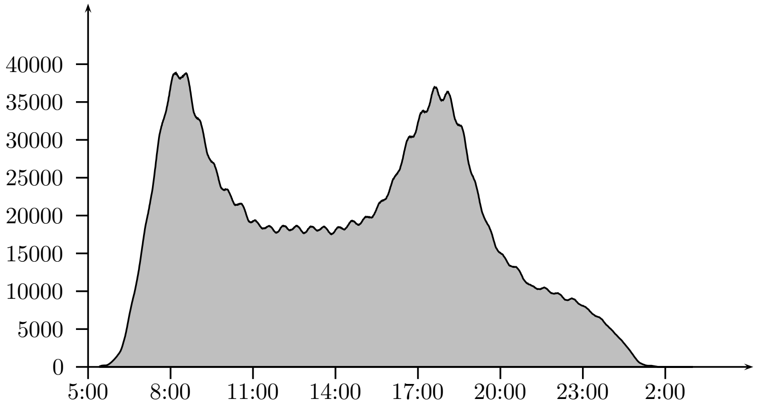


Figure 6: Number of passengers in the system during the day in the undisrupted situation.

of five parts where the first part “Di” describes the place of the disruption; “D1” is between Rotterdam (Rtd) and The Hague (Gv), “D2” is between Gouda (Gd) and Utrecht (Ut), “D3” is between Utrecht (Ut) and Amersfoort (Amf), “D4” is between The Hague (Gv) and Leiden (Ledn), and “D5” is between Amsterdam (Asd) and Utrecht (Ut).

The second part “Ti” describes the time of the disruption; for instances with “T1” the blockage lasts from 16:00 to 19:00, and for instances with “T2” the blockage lasts from 11:00 to 15:00.

The third part “Oi” describes the original rolling stock schedules used for the undisrupted situation. Instances with “O1” are based on a circulation with 114 rolling stock units of the variant with 4 carriages and 44 rolling stock units of the variant with 6 carriages. The circulation is planned such that enough capacity is assigned to all trips to accommodate all passengers in the undisrupted situation, furthermore every train has significant slack capacity compared to the “full” capacity. Instances with “O2” use only 106 and 41 units of the two types respectively.

The fourth part “Ri” refers to the number of available reserve units. Instances with “R1” do not have reserve units. Instances with “R2” have three reserve units allocated in the network as one unit with 4 carriages at Amsterdam (Asd), one unit with 6 carriages at each of the stations of The Hague (Gv) and Rotterdam (Rtd). Instances with “R3” have six reserve rolling stock units distributed as two units with 4 carriages in Amsterdam (Asd), one unit with 4 carriages in Amersfoort (Amf), one unit with 6 carriages in The Hague (Gv), and one of each type of unit in Rotterdam (Rtd).

The fifth part “Pi” specifies the number of passengers in the system. Instances with “P1” have 422022 passengers distributed over the day as shown in Figure 6. Instances with “P2” have 15% more passengers in all passenger groups (total 485375 passengers) relatively distributed in the same way as in the "P1" instances.

Objectives

The objective in the instances consists of the system related goals and the service objectives. As described earlier the service related objectives are measured by the total inconvenience experienced by the passengers. More specifically, we minimize the sum of the delay minutes and the penalty for passengers who leave the system.

For the system related objectives we minimize with highest priority the number of canceled trips and with secondary priority the number of changes to the shunting process and the number of off-balances (see Section 2.6 of Nielsen (2011)). More specifically we use a cost of 500 for introducing a new shunting operation or changing the type of operation performed. Shunting a different number of units or canceling a shunting operation is penalized by 100, while off-balances cost 400. Finally, we use a penalty of 0.0001 for carriage kilometers to ensure that for two solutions with the same value for all other objective terms, the one with lower operating cost is used. Note that all other objective parameters outweigh the total contribution from carriage kilometers.

We note that for a concrete application of the approach it is up to the decision maker to decide on the trade-off between service and system objectives. However, we limit this study to a trade-off that favors the service oriented part of the rolling stock rescheduling process and therefore we apply relatively low costs on changes to the system.

9.2 Results

We first investigate the performance of the approach for different values of the parameter α on a subset of the instances. Recall from Section 7 that α is a parameter that weighs the feedback from the current iteration against feedback from earlier iterations. We then run our approach on the remaining instances with a fixed value of α and discuss the results.

Parameter α

To analyze the effect of α we use the subset of instances named "DiT1O1R2P1" where $i = 1, \dots, 5$. In Table 2 we report the performance of the approach on the instances for distinct values of α in the interval $0 \leq \alpha \leq 1$.

In each run we performed 30 iterations. The table shows the service and system cost of the best solution found as well as in which iteration the best solution was found. Finally the table shows in the last column how many unique solutions were found in the 30 iterations.

Instance	α	Service	System	Iteration	Unique solutions
D1T1O1R2P1	0.20	514983	3949	22	27
	0.35	514983	3949	20	30
	0.50	514869	4649	24	29
	0.65	518450	2749	24	30
	0.80	517241	4550	17	23
	1.00	536076	4449	3	5
D2T1O1R2P1	0.20	1241867	16252	14	30
	0.35	1244684	14551	16	30
	0.50	1245766	13851	22	30
	0.65	1244286	14451	25	30
	0.80	1244286	14651	15	30
	1.00	1250848	14951	3	7
D3T1O1R2P1	0.20	429898	3350	16	17
	0.35	429898	3350	9	19
	0.50	429898	3350	6	20
	0.65	429898	3350	5	20
	0.80	429898	3350	8	17
	1.00	429898	4150	1	3
D4T1O1R2P1	0.20	689086	7951	21	28
	0.35	689086	7851	17	29
	0.50	689086	7851	25	30
	0.65	689086	9952	2	29
	0.80	692419	8051	17	23
	1.00	717539	7652	7	8
D5T1O1R2P1	0.20	607200	9851	18	30
	0.35	607117	9151	13	30
	0.50	606274	11451	6	30
	0.65	611579	10351	21	30
	0.80	619260	8651	9	30
	1.00	644427	11051	21	30

Table 2: Results for instances “DiT1O1R2P1” with $i = 1, \ldots, 5$ for different values of α. Each row contains for a certain instance and α value the service and system objective of the best solution found and the iteration in which it was found. The last column shows the number of unique solutions found in the entire procedure.

For the instance “D1T1O1R2P1” we plotted the traversal of the algorithm in the two-dimensional objective space in separate diagrams for each α value in Figure 7. Similarly Figure 8 contains diagrams for the tests on instance “D5T1O1R2P1”.

We observe that the tests with an α value of 1.00 are consistently worse than all other values for all instances. This is not surprising since only the feedback from the last iteration is taken into account in each step, and we therefore do not use the information obtained in the earlier iterations. In fact, on “D1”-“D4” we observe cyclical behavior of the algorithm. After a few iterations the algorithm alternates between the same two iterations. This behavior can be observed in Figure 7.

For the other values of α we observe that with the parameter $\alpha = 0.20$ the algorithm often takes longer to reach the best solution in the procedure. In Figure 8 it is particularly observable that the procedure takes smaller steps in the objective space with this α value. Setting the parameter to $\alpha = 0.80$ also seems inferior to the remaining values. For the tests with $\alpha = 0.35$, 0.50, and 0.65 we observe only limited difference in the performance of the algorithm. We therefore use the parameter value $\alpha = 0.35$ for the remaining computational tests.

All instances

We tested the approach on all instances using the fixed parameter value $\alpha = 0.35$. The results are shown in Table 3. The first column contains the instance names, and the next three columns show the minimum, average and maximum experienced running times for the optimization module over the 30 iterations of the algorithm. The next column shows the average running time for the simulation module of the algorithm. The columns named “Service” and “System” under “Best for system” denote the service and system objectives of the solutions that minimizes the system related objective. The columns named “Service” and “System” under “Best solution” denote the service and system objectives of solutions that minimize the sum of service and system objectives. Columns “IC-LB” and “WOC-LB” indicate the two types of lower bounds on the service objective.

For all instances with disruptions “D1”, “D3” and “D4” we found solutions without canceled trips. However, for all instances with disruptions “D2” and “D5” it was necessary to cancel one trip. The solutions that only concern the system objective (in columns “Best for system”) all have low system costs and relatively high service costs.

We can make a number of observations concerning the results. First, we observe that the approach is able to improve the service quality significantly in all instances at the cost of a number of changes to the system. The improvements are between 4% and 47%. For disruptions “D1” and “D3” we generally add 3 – 6 shunting operations and cancel or change 15 – 30

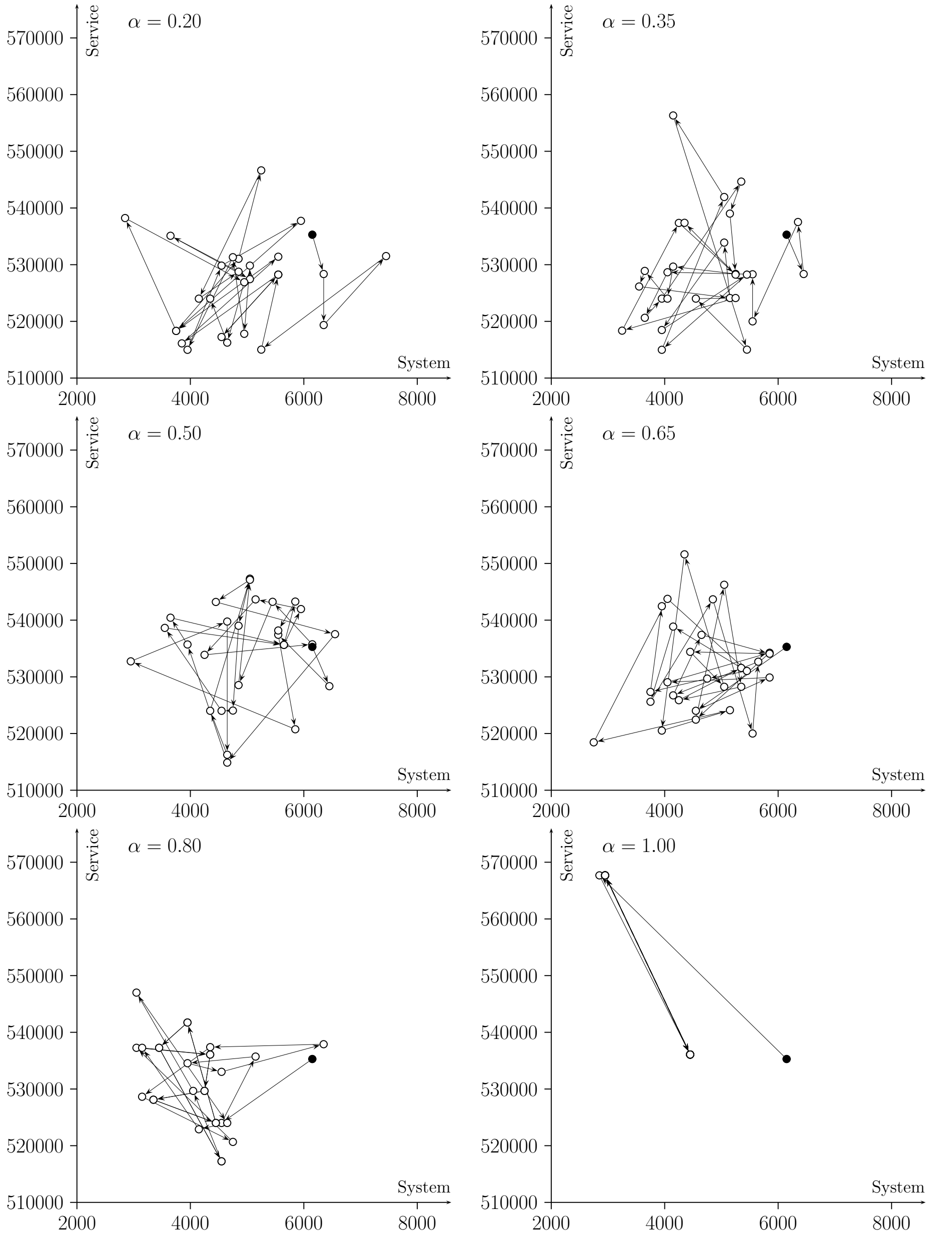


Figure 7: Traversal of the objective space for different settings of parameter α for instance "D1T1O1R2P1". The first iteration is marked by a solid dot.

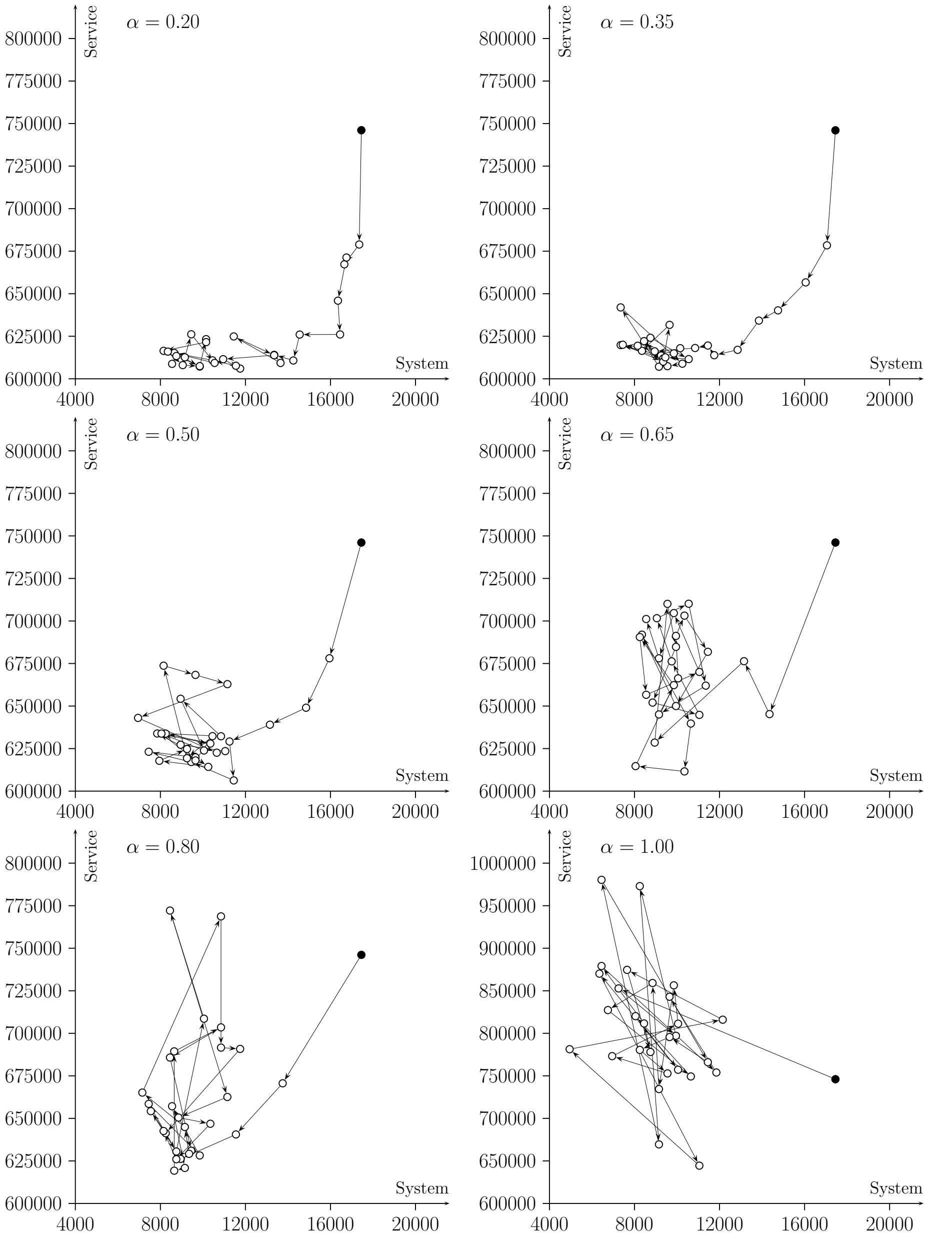


Figure 8: Traversal of the objective space for different settings of parameter α for instance "D5T1O1R2P1". The first iteration is marked by a solid dot. Note that the scale of the y-axis differs for $\alpha = 1.00$.

Instance	OPT time min	avg	max	SIM time	Best for system Service	System	Best solution Service	System	IC-LB	WOC-LB
D1T1O1R1P1	5.6	9.3	12.2	2.8	628211	1449	600017	4649	421007	544674
D1T1O1R2P1	5.6	8.6	14.6	2.6	578550	1249	514983	3949	421007	430681
D1T1O1R3P1	4.5	8.4	21.1	2.6	579315	1249	474465	4449	421007	424816
D1T1O2R1P1	4.4	7.6	12.2	2.8	660150	2049	622684	4750	421007	556666
D1T1O1R2P2	3.7	7.0	14.7	2.8	818667	1249	736754	6450	484126	532481
D1T2O1R1P1	10.2	17.8	41.9	2.8	303037	1149	287612	3049	287612	287612
D2T1O1R1P1	5.7	10.4	13.6	3.1	1818607	4248	1249484	15351	996484	1039719
D2T1O1R2P1	5.7	10.3	14.5	3.1	1818607	4248	1244684	14551	996484	1039719
D2T1O1R3P1	4.9	8.0	17.7	3.1	1818607	4248	1240150	12951	996484	1039719
D2T1O2R1P1	7.1	11.0	16.6	3.1	1805419	3849	1298913	16751	996484	1068677
D2T1O1R2P2	4.8	10.8	14.6	3.3	2567406	4248	1618741	17952	1145355	1263904
D2T2O1R1P1	22.4	203.3	4701.2	3.1	1115628	3248	743964	10051	721884	727088
D3T1O1R1P1	4.6	8.6	21.5	2.7	505971	2349	429898	3750	416595	426838
D3T1O1R2P1	4.5	7.4	21.1	2.6	505971	2049	429898	3350	416595	426838
D3T1O1R3P1	4.6	7.8	27.8	2.7	505971	2049	429898	3350	416595	426838
D3T1O2R1P1	4.7	10.0	21.7	2.6	505971	3049	429898	5750	416595	426838
D3T1O1R2P2	4.5	10.1	21.6	2.7	765566	2049	515438	6651	479347	500599
D3T2O1R1P1	15.9	32.8	70.3	3.0	625948	949	454710	4450	444589	450480
D4T1O1R1P1	3.2	5.3	12.9	2.6	839882	1349	698616	7451	666626	674203
D4T1O1R2P1	3.3	5.1	14.0	2.4	839882	1349	689086	7851	666626	670784
D4T1O1R3P1	3.7	5.4	12.8	2.6	839882	1349	686646	8551	666626	670784
D4T1O2R1P1	2.9	4.7	12.3	2.6	842661	1249	704199	8751	666626	675355
D4T1O1R2P2	3.4	7.5	14.0	2.7	1191093	1349	846999	10952	766343	785204
D4T2O1R1P1	6.8	12.7	21.1	2.8	494192	1048	460461	5650	459536	459572
D5T1O1R1P1	14.3	85.8	404.0	2.8	1116346	1449	610295	9551	520867	536397
D5T1O1R2P1	20.5	192.7	544.6	2.6	1116650	1449	607117	9151	520867	536397
D5T1O1R3P1	22.9	268.3	954.3	2.7	1116650	1449	596495	10151	520867	536397
D5T1O2R1P1	15.5	57.8	788.4	2.8	1122055	1449	665010	14251	520867	536397
D5T1O1R2P2	13.6	79.6	1612.8	3.0	1722815	1449	915054	18453	599038	635239
D5T2O1R1P1	6.6	148.8	3082.6	2.9	765585	549	417038	7451	388216	388216

Table 3: Results for all instances. The table contains computation times for the optimization and simulation module. It contains the best solutions found for the system and the best overall solutions. The last two columns contain the lower bounds based on infinite capacity and operator control respectively.

others. The solutions have 1 or 2 off-balances. For disruptions "D2", "D4" and "D5" we generally experience more changes to the system; 10 – 17 new shunting operations are added while 20 – 35 are changed or canceled. The solutions for these disruptions have up to 7 off-balances. These changes are relatively few from a practical point of view considering the fact that the system contains 14 stations.

Second, we notice that we are generally able to alleviate more passenger inconvenience in the instances with more available reserve units. This concerns instances "R1" with no reserve units, "R2" with three reserve units, and "R3" with six reserve units. The results for these instances are the first, second and third rows respectively in each block in Table 3. Especially for disruption "D1" (between The Hague (Gv) and Rotterdam (Rtd)) the extra reserve units seem to be highly beneficial for the service objective. This indicates that the reserve units at those stations can be allocated where the capacity is needed.

Third, we notice that there is a significant difference between disruptions in the peak hours and in the off-peak hours. This is apparent when comparing the disruptions at time 16:00-19:00 (named "T1") to the disruptions at time 11:00-15:00 (named "T2") – first and sixth row in each block in Table 3 respectively. Generally, less delay minutes occur in a four hour off-peak disruption compared to a three hour disruption during the peak. Also, it is possible for all disruptions "D1",...,"D5" to bring the service objective close to the IC-LB in the off-peak instances. This indicates that capacity is not a bottleneck in those situations.

Fourth, we observe that the instances with less slack capacity in the original plan generally lead to rescheduled solutions with worse service objectives. This concerns "O2" (fourth row in each block in the table) which is planned using much less rolling stock than "O1" (first row), and "P2" (fifth row) which has 15% more passengers than "P1" (second row) but uses the same amount of rolling stock. The results imply that less slack capacity potentially leads to more inconvenience during disruptions.

Fifth, for disruptions "D3" and "D4" we see that the service objective of the best solutions found is quite close to the lower bound given by IC-LB. This means that most of the experienced inconvenience is due to the changed timetable rather than due to lack of capacity. For all other instances we experience that the changes to the timetable are still by far the major contributor to passenger delays, but lack of capacity often contributes around 10%-30% of the delays.

Sixth, in the lower bounds given by WOC-LB, the passenger traveling paths are decided so that the sum of the delays is minimized. In contrast to the IC-LB the arcs have limited capacity and the WOC-LB thus explains some of the delays by the lack of capacity. For several instances this lower bound closes a significant part of the gap between the service objective of the best found solution and the lower bound on the service objective.

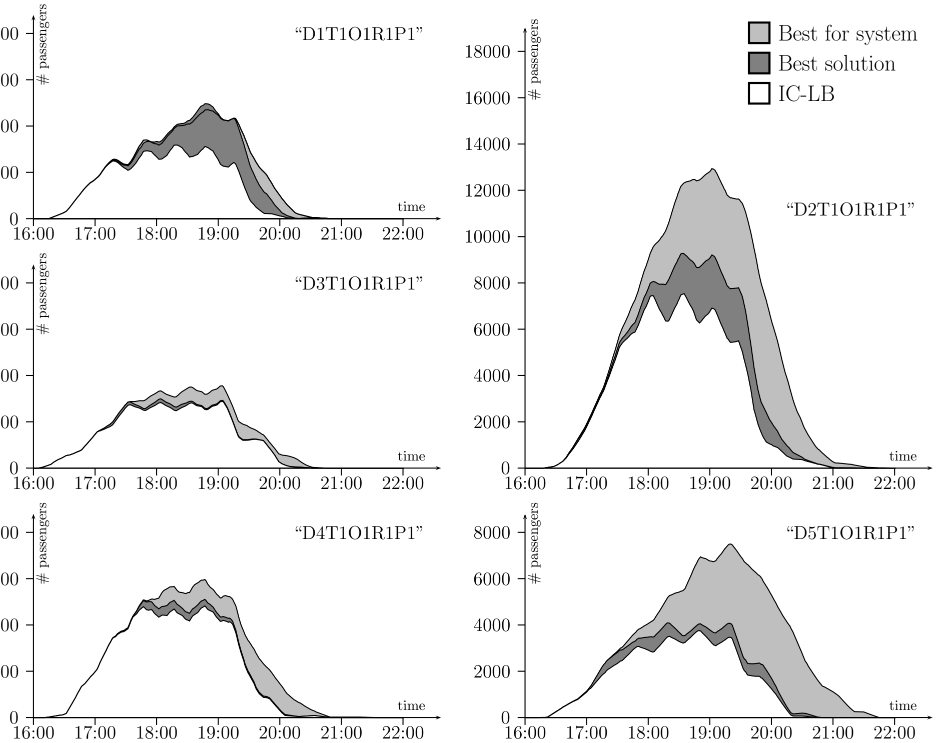


Figure 9: Number of additional passengers in the system during and after the disruption for instances "D1T1O1R1P1", . . ., "D5T1O1R1P1". The area under a curve is the cumulative delay.

In a separate set of computational tests we solved the instances with greatly different weights on the system objectives, i.e. divided or multiplied by a factor ten. But the best solutions found seemed to be comparable to the solutions reported for the current weights. This is probably due to the system costs being relatively small compared to the service costs.

The numbers of passengers that are delayed at a given time are shown in the diagrams of Figure 9. More specifically, the numbers of passengers in the system in addition to the passengers in the corresponding undisrupted situation are shown. The diagrams concern the solutions to the first instance in each block in Table 3. The number of passengers in the system is always at least as large as the number of passengers in the system in the undisrupted situation. This is because people depart at the same time and they arrive no earlier than they are able to in the undisrupted situation. Each diagram contains the IC lower bound which shows how many passengers are delayed because of the changed timetable. Furthermore, each diagram contains the curve for the best found solution (dark gray) and the solution that is best for the system (light gray). We notice that for the complicated disruptions

"D2" and "D5" there is a significant gap between the best found solution and the solution that is best for the system. For disruptions "D3" and "D4" the best solution is almost identical to the lower bound. Finally, we observe that for disruption "D1" the best solution contains more delayed passengers that the best system solution at some time instants. However, after the disruption the best solution brings the delayed passengers out of the system faster than the best system solution.

Computation times

The computation times of the optimization module and the simulation module for each instance are shown in Table 3. For the optimization module the minimum, the average, and the maximum computation time over all iterations is shown in seconds. For the simulation time only the average time is shown since the computation time for that module is very consistent. Running times for the feedback mechanism are not shown as they are neglectable in comparison with the computation time of the other modules.

The tests were performed on an Intel Core 2 duo 3.33 GHz desktop computer with 3 GB of RAM. For the optimization we used CPLEX 11.0 on a single processor while the code for the simulation and the feedback mechanism was written in Java.

The computation time of the optimization module seems to be instance dependent. For all instances involving the disruptions "D1", "D3" and "D4" we experienced computation times of up to 70 seconds for an iteration, and on average the optimization performed each iteration in around 5 – 10 seconds. Instances involving the early disruption "T2" take longer, as a larger part of the day is rescheduled. The instances involving disruption "D2" need longer computation times, especially for the instance "D2T2O1R1P1" the computation time is on average around three minutes per iteration.

The instances involving disruption "D5" generally require longer computation times for the optimization. In fact, the iterations for instance "D5T1O1R3P1" all took at least 22 seconds and on average more than 4 minutes. We attempted to reduce the running time by altering some of the basic CPLEX parameters and adding some valid cuts. This worked on some instances but increased the running time on others. We were thus unable to find settings that consistently improved running times compared to the default settings. However, tuning CPLEX for this set of instances is out of the scope of this study. We did, nevertheless, try another MIP solver for the optimization module. We used Gurobi 3.0.0 which was faster on especially the "D5" instances but was somewhat slower on a number of other instances. A comparison between the running times of the two tested MIP solvers would be unfair since Gurobi 3.0.0 utilizes both cores in the processor and is also a much newer release.

10 Conclusions and future research

In this paper we described a heuristic approach for improving the service aspect of the rolling stock schedule during disruptions. The improvements in service quality comes at the cost of changes to the system. For all instances in the computational tests we were able to improve the service objective and in some cases even reach the lower bound.

In our approach we apply a number of assumptions on the behavior and interaction of passengers. Passengers are assumed to want to arrive at their destinations as quickly as possible and are assumed to leave the system if their delays exceed certain thresholds. Also, passengers are assumed to compete for the scarce capacity in the sense that capacity is assigned to groups of passengers based on their size. We claim that these assumptions reasonably reflect the real situation. But the approach is modular and can be adapted to a system with significantly different assumptions on passenger behavior by changing single components in the iterative approach.

The lower bounds provided in this paper are admittedly rather weak for some instances. For future research we suggest investigating solution approaches for the lower bound SOC-LB. It is appealing to develop a cut-and-price approach for solving model (5) – (9). Such an approach would combine column generation for the paths of passengers and row generation based on Benders decomposition to add valid cuts for the assignment of capacity.

The computation time on most instances is appealing for real-time use, although the running time seems to depend on the structure of the solution. The approach provides a feasible solution in every iteration and may therefore be terminated when a satisfactory solution is reached or when the available computation time is up.

Another means to reducing computation time would be to utilize the rolling horizon framework introduced in Chapter 4 of Nielsen (2011). This integration would also allow us to account for the uncertainty of the system although it requires a more realistic notion of informedness of passengers. We have, however, chosen not to utilize the rolling horizon framework for this study of rolling stock rescheduling with dynamic passengers since it would obscure the contributions of the iterative framework.

References

Abara, J. (1989). *Applying Integer Linear Programming to the Fleet Assignment Problem.* Interfaces, 19(4):20–28.

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

Bratu, S., Barnhart, C. (2006). *Flight Operations Recovery: New Approaches Considering Passenger Recovery.* Journal of Scheduling, 9(3):279–298.

Dumas, J., Aithnard, F., Soumis, F. (2009). *Improving the Objective Function of the Fleet Assignment Problem.* Transportation Research Part B: Methodological, 43(4):466–475.

Dumas, J., Soumis, F. (2008). *Passenger Flow Model for Airline Networks.* Transportation Science, 42(2):197–207.

Ginkel, A., Schöbel, A. (2007). *To Wait or Not to Wait? The Bicriteria Delay Management Problem in Public Transportation.* Transportation Science, 41(4):527–538.

Giovanni, L.D., Heilporn, G., Labbé, M. (2007). *Optimization Models for the Delay Management Problem in Public Transportation.* European Journal of Operational Research, 189(3):762–774.

Hane, C.A., Barnhart, C., Johnson, E.L., Marsten, R.E., Nemhauser, G.L., Sigismondi, G. (1995). *The Fleet Assignment Problem: Solving a Large-Scale Integer Program.* Mathematical Programming, 70(1):211–232.

Nielsen, L.K. (2011). *Rolling Stock Rescheduling in Passenger Railways.* Ph.D. thesis, Erasmus University, Rotterdam, The Netherlands.

Oppenheim, N. (1995). *Urban Travel Demand Modeling: From Individual Choices to General Equilibrium.* Wiley-Interscience.

Schachtebeck, M. (2010). *Delay Management in Public Transportation: Capacities, Robustness, and Integration.* Ph.D. thesis, Universität Göttingen, Germany.

Publications in the Report Series Research* in Management

ERIM Research Program: “Business Processes, Logistics and Information Systems”

2010

Linearization and Decomposition Methods for Large Scale Stochastic Inventory Routing Problem with Service Level Constraints
Yugang Yu, Chengbin Chu, Haoxun Chen, and Feng Chu
ERS-2010-008-LIS
http://hdl.handle.net/1765/18041

Sustainable Passenger Transportation: Dynamic Ride-Sharing
Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang
ERS-2010-010-LIS
http://hdl.handle.net/1765/18429

Visualization of Ship Risk Profiles for the Shipping Industry
Sabine Knapp and Michel van de Velden
ERS-2010-013-LIS
http://hdl.handle.net/1765/19197

Intelligent Personalized Trading Agents that facilitate Real-time Decisionmaking for Auctioneers and Buyers in the Dutch Flower Auctions
Wolfgang Ketter, Eric van Heck, and Rob Zuidwijk
ERS-2010-016-LIS
http://hdl.handle.net/1765/19367

Necessary Condition Hypotheses in Operations Management
Jan Dul, Tony Hak, Gary Goertz, and Chris Voss
ERS-2010-019-LIS
http://hdl.handle.net/1765/19666

Human Factors: Spanning the Gap between OM & HRM
W. Patrick Neumann, and Jan Dul
ERS-2010-020-LIS
http://hdl.handle.net/1765/19668

AUK: a simple alternative to the AUC
Uzay Kaymak, Arie Ben-David, and Rob Potharst
ERS-2010-024-LIS
http://hdl.handle.net/1765/19678

The Value of Optimization in Dynamic Ride-Sharing: a Simulation Study in Metro Atlanta
Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang
ERS-2010-034-LIS
http://hdl.handle.net/1765/20456

MIPLIB Truckload PDPTW Instances Derived from a Real-World Drayage Case
F. Jordan Srour, Tamas Mahr, Mathijs de Weerdt, and Rob Zuidwijk
ERS-2010-036-LIS
http://hdl.handle.net/1765/20883

The Value of Information in Container Transport: Leveraging the Triple Bottom Line
Rob A. Zuidwijk, and Albert Veenstra
ERS-2010-039-LIS
http://hdl.handle.net/1765/20994

Smart Grid Economics: Policy Guidance through Competitive Simulation
Wolfgang Ketter, John Collins, and Carsten Block
ERS-2010-043-LIS
http://hdl.handle.net/1765/21307

Global Diffusion of the Non-Traditional Banking Model and Alliance Networks: Social Exposure, Learning and Moderating Regulatory Effort
Alexander Cuntz, and Knut Blind
ERS-2010-044-LIS
http://hdl.handle.net/1765/21681

Rescheduling of Railway Rolling Stock with Dynamic Passenger Flows
L. Kroon, G. Maroti and L. Nielsen
ERS-2010-045-LIS
http://hdl.handle.net/1765/22613

[*] A complete overview of the ERIM Report Series Research in Management:
https://ep.eur.nl/handle/1765/1

ERIM Research Programs:

LIS	Business Processes, Logistics and Information Systems
ORG	Organizing for Performance
MKT	Marketing
F&A	Finance and Accounting
STR	Strategy and Entrepreneurship