Timing of Train Disposition: Towards Early Passenger Rerouting in Case of Delays *

Martin Lemnian[1], Ralf Rückert[1], Steffen Rechner[1], Christoph Blendinger[2], and Matthias Müller-Hannemann[1]

1 **Institut für Informatik**
Martin-Luther-Universität Halle-Wittenberg, Germany
`{mlemnian,muellerh,rechner,rueckert}@informatik.uni-halle.de`

2 **DB Mobility Logistics AG, Jürgen-Ponto-Platz 1, 60329 Frankfurt am Main, Germany,** `Christoph.Blendinger@deutschebahn.com`

Abstract

Passenger-friendly train disposition is a challenging, highly complex online optimization problem with uncertain and incomplete information about future delays. In this paper we focus on the timing within the disposition process. We introduce three different classification schemes to predict as early as possible the status of a transfer: whether it will almost surely break, is so critically delayed that it requires manual disposition, or can be regarded as only slightly uncertain or as being safe. The three approaches use lower bounds on travel times, historical distributions of delay data, and fuzzy logic, respectively. In experiments with real delay data we achieve an excellent classification rate. Furthermore, using realistic passenger flows we observe that there is a significant potential to reduce the passenger delay if an early rerouting strategy is applied.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems; G.2.2 Graph Theory (Graph algorithms; Network problems)

Keywords and phrases train delays, event-activity model, timing of decisions, passenger flows, passenger rerouting

Digital Object Identifier 10.4230/OASIcs.ATMOS.2014.122

1 Introduction

Disruptions and delays frequently occur in public transport for various reasons. For passengers with planned transfers this means that there is a relatively high risk to miss some transfer. Train disposition in case of disruptions is therefore a research theme of high importance. It is about a complex online optimization problem where we are given a massive stream of messages about delays, cancellations, extra trains and the like. The task of disposition is the real-time reaction to unexpected events with the goal to minimize negative effects. Depending on the type of conflicts to solve and dispositional actions one distinguishes between timetable adjustment to the current operational conditions (often called *delay management*), rolling stock rescheduling and crew rescheduling [17]. Here we focus on timetable adjustment ("to wait or not to wait" [1]) and the crucial questions: When should we decide which transfers are to be maintained and what is the effect on passengers?

* partially supported by grant MU 1482/4-3 within the DFG program SPP 1307 *Algorithm Engineering* and Deutsche Bahn AG.

14th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'14).
Editors: Stefan Funke and Matúš Mihalák; pp. 122–137
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Passenger-oriented disposition. Passenger-oriented disposition as introduced by Berger et al. [4] requires precise knowledge about the travel routes of passengers, ideally the exact travel plans of all passengers. In this work we continue in this spirit and assume that for each passenger the planned travel connection is known. The union of all travel connections and their multiplicities yields what we call a *passenger flow*. In today's daily operations only partial information is available for dispatchers. We envision that this will change in the future. Since almost all tickets are sold electronically, a majority of planned connections could be available through the vending systems. A considerable fraction of passengers travels with travel passes allowing unlimited travel within a given time period and region. Such passengers may either register deliberately their travel plans in order to get assistance in case of delays, or staff on train collects this information when checking tickets. For the purpose of disposition anonymity can be ensured since no personal information is required to take a decision. We just need some way to inform passengers about new travel recommendations. For experiments in this paper, we use detailed realistic passenger flow data provided by Deutsche Bahn AG.

Disposition has to consider several, partially conflicting goals. Examples of possible optimization goals are to minimize the total passenger delay, cost of compensations (payments for taxis, hotels, or travel vouchers), total number of missed transfers, additional operational costs, or loss of reputation of the operating company. Even if complete information about the current operational situation is available, the prediction of the further development of delays is fairly difficult since (1) additional delays (secondary delays) can be induced, for example, by headway conditions, tight track capacities, and restrictions of the network topology, and (2) available buffer times can be used to some extent to reduce given delays. Berger et al. [5] presented a stochastic model to predict event time distributions in an online scenario. Recently, Kecman and Goverde [19] developed a microscopic, fine-grained model for train event time prediction.

Related Work. There is a rich literature on delay management. Based on so-called event-activity networks, there are various formulations as integer linear programming problems [28, 29, 30]. The computational complexity of delay management turned out to be NP-hard even under simplifying assumptions [14, 15]. While early formulations of delay management ignored track capacities, more elaborated models take them into account [25]. A typical assumption is that all train lines operate periodically, and that passengers who miss a connection wait for the next one in the next period. Delay management with rerouting of passengers has been considered by [12, 13, 27]. A major deficiency of all these delay management models is, however, that they assume that at some point in time full knowledge about all future delays and their exact sizes becomes available. Cicerone et al. [9] take the viewpoint of robust optimization and try to find waiting decisions which are robust against future delays. Various other approaches use online algorithms for train disposition [2, 3, 6, 16, 20, 21]. Biederbick and Suhl use agent-based simulation for passenger-oriented disposition [7, 8]. Waiting policies in a stochastic context are considered in [1]. Online optimization of a single line has been examined by [3, 21]. D'Ariano (and co-authors) use local rerouting and speed adjustments to solve track occupation conflicts [10, 11]. A detailed and quite realistic framework for train disposition has been developed within the DisKon project [26]. Another prototypal tool for timetable adjustment is ANDI/L which uses heuristics to reduce the search space in order to meet online capabilities [22]. Up to now, optimizing approaches for short-term train disposition are not used in daily operations (see, for example, [18]).

Uncertainty and timing of decisions. Timing when to take dispositional decisions has been neglected to a large extent. Recall that classical delay management takes a static viewpoint. It is assumed that the whole delay scenario (i.e. the size of the delay of all activities) becomes known at a certain point in time. In contrast, in online management decisions are usually taken greedily when new information about delays arrives. However, the analysis of historical delay data clearly shows a high volatility of the available data. Prediction algorithms of how delays evolve over time are facing a considerable amount of uncertainty. In particular, it is difficult to predict whether a planned transfer can be maintained or not. For the benefit of shorter travel times, train schedules are designed to realize relatively short buffer times for transfers. The downside of this optimization is that many transfers are so tight that it becomes hard to predict whether they will break or not.

Goals and contribution. With this study we want to work towards a better understanding of the timing dimension of waiting decisions. Since the realization of decisions takes some time (train staff and staff at the station have to be informed; new travel recommendations have to be communicated to passengers), the time window for taking any decision closes about 15 minutes before the actions must be realized. In current practice there seems to be a tendency to decide as late as possible (based on interviews with experienced practitioners from Deutsche Bahn AG). In general, there is a trade-off: the longer we wait, the more accurate information will be available, but the earlier we decide, the larger is the range of alternatives. Clearly, having a larger set of alternatives is beneficial for all those passengers who have to change their travel route. Thus, the challenging task is to find an appropriate timing of decisions. We here aim at a refinement of the decision process which takes the temporal dimension into account. Important use cases are

- large delays of a single train
- unavailability of a track segment for a certain time period.

In both cases passengers may take advantage of a timely detection of problems, the earlier the better. We study the following questions:

1. How early can we predict that a planned transfer will break? And if we do so, how reliable are our recommendations?
2. How early should we inform passengers? Should we do it immediately, or should we wait with final recommendations in light of uncertain information?
3. How many passengers can take advantage of early rerouting recommendations?

We propose several approaches and evaluate them with experiments based on realistic passenger flows and historical delay data.

Overview. The remainder of this work is structured as follows. In Section 2, we explain in detail the train disposition framework. Afterwards, in Section 3 we introduce several methods for the classification of transfers with respect to the likelihood that they can be realized. We describe our experiments and report computational results in Section 4. Finally, we summarize our findings and give remarks on possible future research.

2 Train Disposition Framework

2.1 Event-Activity Networks and Passenger Flows

To model railway traffic, we use a so-called *event-activity network* $\mathcal{N} = (\mathcal{V}, \mathcal{A})$ which is a directed, acyclic graph with vertex set $\mathcal{V}$ and arc set $\mathcal{A}$. Each departure and arrival event of some train is modeled by a vertex $v \in \mathcal{V}$ and is equipped with a number of attributes:

its event type *type* (either departure *dep* or arrival *arr*), an identifier for the corresponding train *trainID*, the type of the train *trainType*, the station s where the event takes place, the originally planned scheduled event time t^{sched}, a predicted realization time t^{pred}, and the actual realization time t^{act} (as soon as available). Times are usually given in minutes after some point of reference.

An arc models a precedence relation between events. We distinguish between different types of arcs ("activities"): *driving arcs* $\mathcal{A}_{\text{drive}}$, modeling the driving of a specific train between two stations without intermediate stop, *waiting arcs* $\mathcal{A}_{\text{wait}}$, modeling a train standing at a platform, and *transfer arcs* $\mathcal{A}_{\text{transfer}}$, modeling the possibility for passengers to change between two trains. The planned duration of driving and waiting activities is implicitly given by the time difference between the scheduled event times of the corresponding events. Each activity $a \in \mathcal{A}$ has a lower bound ℓ_a specifying its minimum duration. The lower bound gives the minimum travel time between two stops of the corresponding train type under optimal driving conditions for driving activities. For a waiting arc $a \in \mathcal{A}_{\text{wait}}$, the value of ℓ_a represents the minimum dwell time which is assumed to be necessary for boarding and deboarding of passengers. For a transfer activity $a \in \mathcal{A}_{\text{transfer}}$, the lower bound ℓ_a denotes the minimum time a passenger needs to change from a feeder train to a connecting train.

For ease of exposition, we here use a path formulation of passenger flows. Associated with each passenger p is a path P_p in the event-activity network $\mathcal{N} = (\mathcal{V}, \mathcal{A})$. Such a path always represents the current passenger's route, initially the planned connection, and later, whenever updates occur, the new route. We use an additional layer of data structures to represent passenger flows on top of the event-activity network.

2.2 Prediction of Event Times and Delay Propagation

Whenever new information about realized event times or delays becomes available, it must be propagated through the network. To partially automize the train disposition process, train operators often apply standard waiting time rules: Given a number $wt_a \in \mathbb{N}_0$ for a transfer activity $a \in \mathcal{A}_{\text{transfer}}$, the connection shall be maintained if the departing train has to wait at most wt_a minutes compared to its original schedule. In addition, a dispatcher may manually overwrite this value in order to keep a connection. Note that a departure event w may have several incoming transfer arcs. By $w.t^{maxwait} := \max\{w.t^{sched} + wt_a \mid a = (v, w) \in \mathcal{A}_{\text{transfer}}\}$ we denote the maximum waiting time induced by any delayed feeder train of w.

We here assume that a permanently running process updates predictions of future event times with respect to standard waiting time rules. This process is fairly complicated and can be done at different levels of sophistication, with macroscopic [24] or microscopic models [19], or with stochastic distributions [5]. In order to be consistent, predicted event times shall satisfy the following constraints:

$$w.t^{pred} \geq v.t^{pred} + \ell_a \text{ for all } a = (v, w) \in \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}}, \tag{1}$$

which means that minimum driving times and minimum dwell times are respected, and

$$\begin{aligned} w.t^{pred} \geq v.t^{pred} + \ell_a \text{ for all } a = (v, w) \in \mathcal{A}_{\text{transfer}} \\ \text{with } w.t^{sched} < v.t^{pred} + \ell_a \leq w.t^{sched} + wt_a \end{aligned} \tag{2}$$

which ensure that a connecting train waits for its feeder trains at least as long as the standard waiting time rules specify. Moreover, we require that no train may depart before its scheduled event time, i. e. $v.t^{pred} \geq v.t^{sched}$ for all $v \in \mathcal{V}$ with $v.type = dep$.

With respect to realized event times, a transfer $a = (v, w) \in \mathcal{A}_{\text{transfer}}$ is *maintained* if $w.t^{act} \geq v.t^{act} + \ell_a$; otherwise, we say that the transfer has been broken.

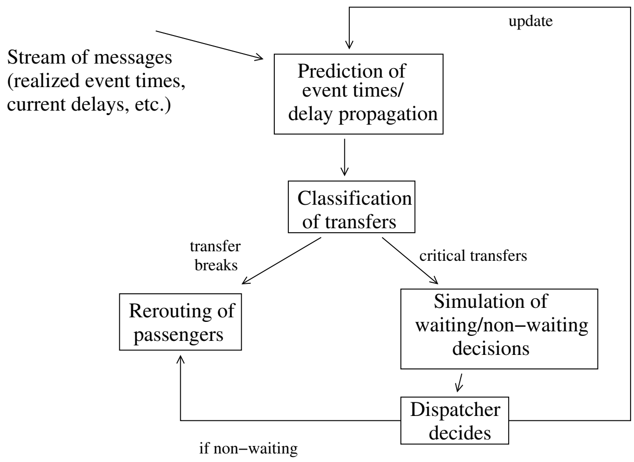


Figure 1 Schematic sketch of the train disposition process.

A fairly simple and in practice often used way to implement delay propagation is based on the following rules: (a) the current delay is not reduced on driving arcs, (b) on a waiting arc $a = (v, w)$, the slack $slack(a) = w.t^{sched} - v.t^{sched} - \ell_a$ given by the difference of planned and minimum dwell time is fully used to reduce delays, and (c) standard waiting time rules are applied on transfer arcs. The latter means that the predicted departure time $w.t^{pred}$ of a departure event w is set as the maximum of three values: $w.t^{sched}$ (its scheduled departure time), $u.t^{pred} + \ell_{(u,w)}$ where (u, w) denotes the waiting arc before the departure event w (if it exists), and $\max\{v.t^{pred} + \ell_a\}$ where the maximum is taken over $a = (v, w) \in \mathcal{A}_{\text{transfer}}$ with $v.t^{pred} + \ell_a \leq w.t^{sched} + wt_a$, the delay propagated through transfer arcs. In this paper we use this propagation scheme for our experiments.

2.3 Train Disposition

Next, we give a very brief high-level description of the train disposition process, see Fig. 1 and also [4]. Periodically, say every minute or every 30 seconds, we obtain from an external source the latest information about realized event times, current delays, train cancellations, extra trains, and the like. All these messages are parsed and taken to update the predictions of event times. Then, the predicted event times are used to classify which transfers potentially require dispositional actions (for details see the following section). In general, we obtain a set $\mathcal{A}_{\text{watch}}$ of transfer arcs which have to be "watched". Now several strategies are possible how and when to handle each of these transfers. For example, a simple strategy could be to order the transfer arcs by the time for which the departing trains is originally scheduled and to decide what to do a fixed amount of time, say 20 minutes, before the event occurs.

Here, we propose a more flexible scheme. With each transfer arc in $a \in \mathcal{A}_{\text{watch}}$ we associate a decision time $a.t^{dec}$ which denotes when the decision about this case shall be obtained. We use a priority queue with the decision times $a.t^{dec}$ as keys to obtain the order by which all cases are handled. Roughly speaking, decision times are chosen by importance (how many passengers are involved) and criticality (how likely is it that the transfer breaks). The crucial idea is that the "clearly broken cases", i. e. those where the feeder train is so severely delayed that maintaining certain connections is unreasonable or operationally infeasible, shall be handled immediately. In such cases we instantly try to reroute all passengers which currently have planned to use a transfer that will break. If we

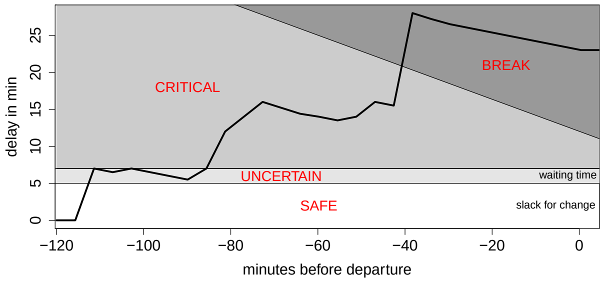


Figure 2 Delays fluctuate over time. The figure sketches exemplarily the development of the delay in minutes of a single feeder train, two hours before the departure of a connecting train of a planned transfer. Depending on the size of the delay and the potential to regain, the transfer is classified as "SAFE", "UNCERTAIN", "CRITICAL" or "BREAK".

succeed, we can eliminate the corresponding transfer arc from $\mathcal{A}_{\text{watch}}$. All remaining transfer arcs are periodically reclassified. For critical cases where the feeding train is delayed by so much that the standard waiting time rules do not apply anymore, but maintaining the transfer is still an option, the alternatives "waiting" or "not waiting" have to be evaluated with respect to the effect on the passenger flow. To this end, for each considered alternative the disposition module tentatively first computes new event time predictions, and second updates the passenger flow. Flow updates are necessary if a transfer arc breaks which carries non-zero flow in the given scenario. In such a case all passengers on such a transfer arc have to be rerouted, that means, a new train connection has to be computed for each passenger.

The hypothetical change of flow can be evaluated in terms of one or several objectives (for example, total passenger delay in minutes). The dispatcher then has to decide between the given alternatives. Whenever a final disposition decision is taken, it must be implemented. Within our framework each decision requires to update event times and passenger flows.

3 Classification of Transfers

In this section we propose classification schemes for transfers. The goal is to classify each transfer as early as possible, i. e., to determine whether it will be maintained or will break, see Fig. 2. The semantics of the classes used by our classification are as follows.

- **SAFE** — the transfer seems to be safe. A transfer is regarded as safe if the slack time for the transfer is positive, i. e. the connecting train will be reached unless unexpected delay of the feeder train occurs.
- **UNCERTAIN** — the transfer is uncertain but likely to hold. In this case the feeder train is delayed, but by applying the standard waiting time rule, this transfer will be maintained.
- **CRITICAL** — the transfer is likely to break unless an explicit waiting decision is taken. Here the delay of the feeder train is so large that the desired transfer can only be maintained if the dispatcher decides that the transfer will be maintained, the standard waiting time rule is not sufficient.
- **BREAK** — the transfer will break. Even under best conditions, the delayed feeder train will arrive too late to reach the connecting train (which itself may be delayed). The delay is so large that waiting of the connecting train is no feasible option, or is even meaningless if the feeder train has been cancelled. Whether waiting has to be considered as a feasible option may depend on whether passengers have reasonable alternatives, in

particular in the late evening. This classification assumes that the connecting train will not be (further) delayed in the future.

The focus of our classification algorithms is on identifying breaking transfers as early as possible. We develop three different classification schemes.

3.1 Classification by Lower Bounds

The rationale behind this classification scheme is a conservative view, that is, we primarily want to detect those cases where the transfer is going to break. For each travel arc, we use historical data to derive lower bounds for the travel time. If no historical data is available we assume that the minimum travel time is 7% lower than the scheduled travel time. Current delays are propagated with respect to these lower bounds. For each arrival event $arr \in \mathcal{V}$ we so obtain a lower bound $arr.t^{lb}$ on its realization time.

For a transfer arc $a = (arr, dep) \in \mathcal{A}_{\text{transfer}}$, we classify it as BREAK if $arr.t^{lb} + \ell_a > \max\{dep.t^{pred}, dep.t^{sched} + wt_a\} + \delta$. The parameter $\delta > 0$ specifies a "safety margin" by which the classification can be tuned towards a conservative classification by increasing its value. Such an arc is classified as SAFE, if $arr.t^{pred} + \ell_a \leq dep.t^{pred}$. It is classified as UNCERTAIN if it is not SAFE but $arr.t^{pred} + \ell_a \leq dep.t^{sched} + wt_a$. In all remaining cases it is regarded as CRITICAL.

3.2 Classification by Transfer Probabilities

Again we make heavily use of historical data to derive for each transfer arc a probability distribution for its realizability. For a train of type *trainType* which is now delayed by d minutes, and a time horizon of h minutes we have an empirical density function $f_\Delta^{trainType,h,d} \colon \mathbb{Z} \mapsto [0,1]$ for the probability that the current delay will change by x minutes in h minutes from now. For example, $f_\Delta^{ICE,30,5}(2)$ gives the probability that an ICE with a current delay of five minutes will have an additional delay of two minutes half an hour later. Note that these probabilities only depend on the type of a train but not on its specific route. With the help of this function we can derive the probability that a future event v for this train will occur at time $t = v.t^{sched} + d + x$, where $d + x \geq 0$ holds. Namely, we define

$$f^{trainType,h,d} \colon \mathbb{Z} \mapsto [0,1]\,, \text{ with } \quad f^{trainType,h,d}(t) = f_\Delta^{trainType,h,d}(x), x \in \mathbb{Z}.$$

Hence, we can compute the distribution of departure and arrival times for all future events with respect to the current delay scenario.

For a future transfer arc $a = (arr, dep) \in \mathcal{A}_{\text{transfer}}$ we further derive the probability distribution that the transfer will be maintained as follows. As before denote by ℓ_a the minimum transfer time and by $dep.t^{maxwait}$ the maximum waiting time of the departing train. Thus, unless the departing train itself is delayed, it will wait at most until $t_{max} = dep.t^{maxwait}$. The probability p that a transfer will be maintained if the feeder train arrives before t_{max} is

$$p = \sum_{t=0}^{t_{max}-\ell_a} f^{arr.trainType,arr.h,arr.d}(t).$$

The probability that a transfer will be maintained after t_{max} depends on the distribution of the departing train as well. In this case

$$p = \sum_{t_1=t_{max}-\ell_a+1}^{\infty} \sum_{t_2=t_1+\ell_a}^{\infty} f^{arr.trainType,arr.h,arr.d}(t_1) \cdot f^{dep.trainType,dep.h,dep.d}(t_2),$$

Table 1 Classification rules based on transfer probability p.

class	rule
SAFE	$p \geq 0.96$
UNCERTAIN	$0.60 \leq p < 0.96$
CRITICAL	$0.05 \leq p < 0.60$
BREAK	$p < 0.05$

Table 2 Fuzzy inference rules.

current delay	regain potential	transfer
on time		SAFE
small delay	possible	SAFE
small delay	impossible	UNCERTAIN
strong delay	possible	CRITICAL
strong delay	impossible	BREAK

where $arr.h$ and $dep.h$ denote the current time horizon for the arrival event arr and the departure event dep, and where $arr.d$ and $dep.d$ denote the current delays of the arriving and departing train, respectively. The resulting probability value p is used to classify the transfer according to empirically chosen thresholds given in Table 1. Note that transfer probabilities as described, but with different thresholds are used by Deutsche Bahn AG for online timetable information.

3.3 Classification by Fuzzy Logic

To classify a transfer with respect to uncertainty, we use a classifier based on fuzzy logic. We consider three linguistic variables for the transfer's arrival event:

- the *current delay* with possible values *on time*, *small delay* and *strong delay*,
- the *regain potential* with possible values *possible* and *impossible*,
- the *state* of a transfer with values SAFE, UNCERTAIN, CRITICAL, and BREAK.

Figure 3 shows for an arrival event arr how the variables $arr.t^{lb}$ are fuzzified into linguistic variables *current delay* and *regain potential* with certainty p_d and p_r, respectively. We use the interference rules shown in Table 2 to determine the state of a transfer. The lines of the table are to be read as

$$\text{IF current delay=}\ldots\text{AND regain potential} = \ldots\text{THEN transfer=}\ldots.$$

We use the maximum of p_d and p_r to compute the certainty p_t of a transfer.

In the sequel, we consider the classification by lower bounds as our STANDARD classification scheme (justified by the experiments below), and call the classification by transfer probabilities simply STOCHASTIC, and the one based on fuzzy logic FUZZY.

4 Experimental Results

4.1 Test Instances, Environment and Software

The basis for our computational study is the German train schedule of 2013 and historical process data from 2011-2013. The schedule contains 36,772 trains, 8,592 stations and the corresponding event activity network consists of about 2 million events. Process data (realized event times, new delays, etc) have an average volume of about 1.6 GiB per day

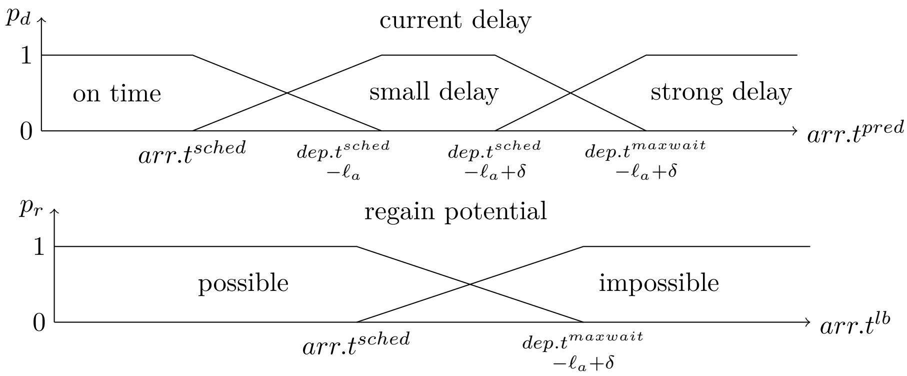
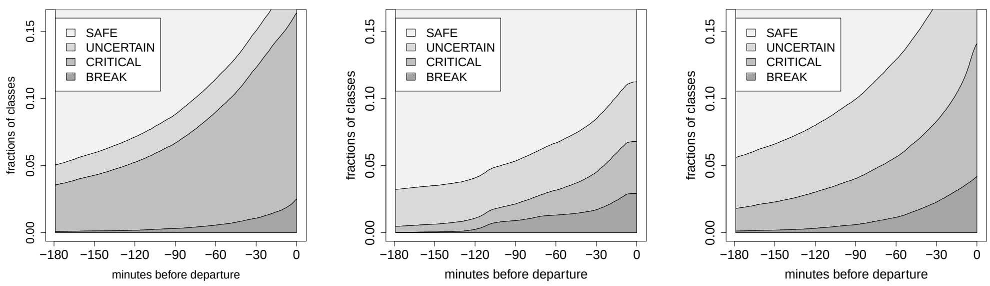


Figure 3 Fuzzification of the two variables *current delay* and *regain potential* for a transfer arc $a = (arr, dep)$.



(a) STANDARD **(b)** STOCHASTIC **(c)** FUZZY

Figure 4 Classification of transfers by the three methods: Distribution of the four classes.

(compressed XML-files). Realistic passenger flows have been provided by Deutsche Bahn AG for several test days in 2013. These flows are derived from about 2.9 million passengers and their travel connections per day; the passengers travel on roughly 400 000 different routes, with an average travel time of 119 minutes and .73 transfers on average.

All experiments were run on a PC (Intel(R) Xeon(R), 2.93GHz, 8MB L3-cache, 48GB main memory under Ubuntu Linux version 12.04 LTS). Our code is written in C++ and has been compiled with g++ 4.8.1. It is an extension of the train disposition system described in [4] and has been developed on top of MOTIS (multi-objective travel information system) using its capability of searching optimal travel routes [23] and online delay propagation [24]. If a passenger has to be rerouted, we compute a fastest alternative connection with the fewest number of transfers. Since our code is a mere prototype and efficiency is not the topic of this paper, we do not report running times. We simply remark that the achieved running times are fast enough to apply our approach in an online scenario.

4.2 Experiments

Experiment 1: Evaluation of classification schemes. How good is the predictive power of the information of current delays for the feasibility of transfers x minutes in advance? In particular, how early are we able to identify breaking transfers and how reliable is our classification?

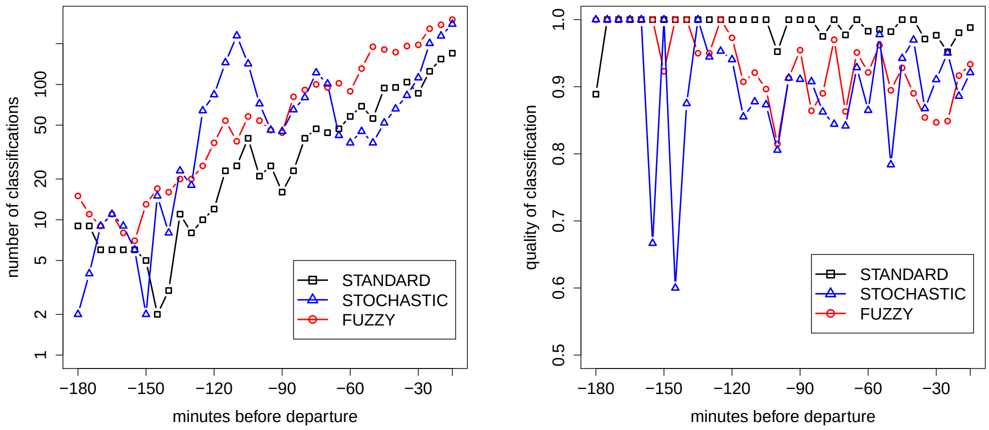


Figure 5 Number of cases (note the logarithmic scale) and accuracy of the classification schemes for class BREAK. Classifications are grouped into bins of 5-minute intervals.

To study these questions, we recorded for each classified transfer its corresponding state from the first point in time when it has been classified as non-safe (by default, we consider all planned transfers as SAFE) until its realization. The parameter δ used by STANDARD and FUZZY has been set to the value 4. Fig. 4 shows the fraction of the four classes within the last three hours before realization. Whether a transfer has been maintained or not, has been evaluated with respect to realized event times. The false negative rate of our classification schemes is very small. For example, only 0.96% of transfers are classified as SAFE by STANDARD, but eventually break. Since new delays may occur spontaneously, such misclassifications are almost unavoidable.

Fig. 5 shows the quality of the three classification schemes with respect to false positives, i. e. cases where the classification predicts BREAK but the transfer is eventually maintained. The x-axis represents the time before the scheduled event time, while the y-axis displays the fraction of cases with a correct classification. The overall best classification rate is obtained by STANDARD, but the two other methods also work quite well. The lower accuracy of STOCHASTIC may be explained by the lack of route-specific probability distributions.

We observe a trade-off between accuracy and number of detected cases of type BREAK. High accuracy is important since one has to avoid rerouting passengers without any need. On the other hand, the potential of early rerouting can only be used if cases of type BREAK are detected. While STANDARD is the most conservative method with an excellent classification rate, it detects the overall smallest number of cases of type BREAK. Transfers classified as CRITICAL may break or may be maintained by disposition. Fig. 6 shows the number of cases where critical transfers (classified by STANDARD) break or are maintained. Since the information whether a transfer breaks is not directly available to us, we take as a proxy the following condition. We consider a transfer as maintained due to an explicit waiting decision if the departing train departs later as scheduled, but would have already been available on time. It turns out that a high fraction of transfers classified as CRITICAL eventually breaks (see Fig. 6). If early rerouting is beneficial to passengers — which we study in the second experiment —, then this finding suggests to decide about critical transfers as early as possible.

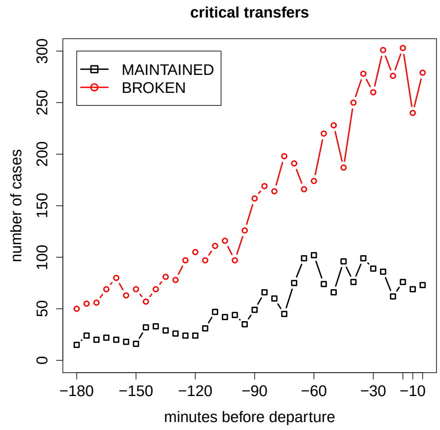


Figure 6 Critical transfers (as classified by STANDARD), where we distinguish a posteriori between maintained and broken transfers.

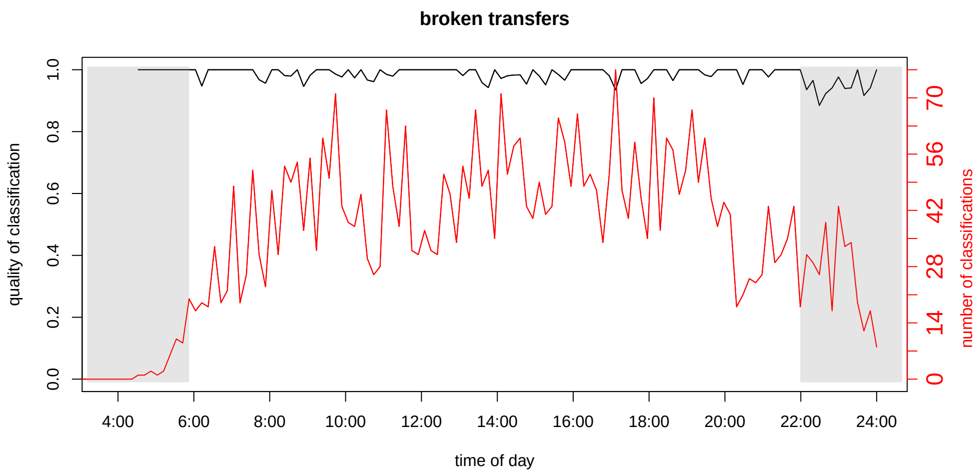


Figure 7 Accuracy of the STANDARD classification of events of type BREAK.

Fig. 7 provides a closer look at the error rate of the STANDARD classification for events of type BREAK on September 13, 2013. In particular, we are interested in the question whether the classification rate depends on the time where the broken event occurs. To exclude the effect that some transfers are manually dispatched, we here plot the remaining error rate that we obtain if we remove such cases. In the peak time between 6:00 a.m. and 10 p.m., the accuracy of the STANDARD classification is 98.9%, it slightly degrades during the night where it becomes 95.7% on average. Further analysis is required to understand why the classification rate is time-dependent.

Experiment 2: Potential of early rerouting. What is the benefit of an early rerouting strategy for the passengers?

To answer this question, we have set up the following simulation experiment. Given a realistic passenger flow for some specific traffic day, we first select the 1000 most important transfers, where importance of a transfer is just the number of passengers who plan to use

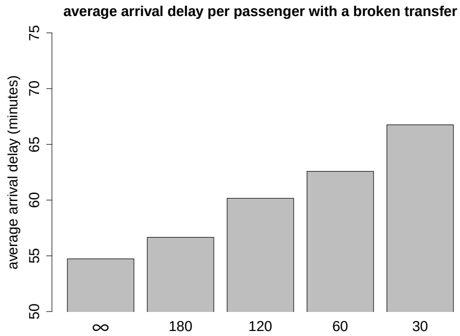


Figure 8 Average arrival delay at the destination if a planned transfer breaks. Rerouting is applied either immediately (denoted by ∞) or (at most) 180, 120, 60, or 30 minutes before the planned departure of the broken transfer.

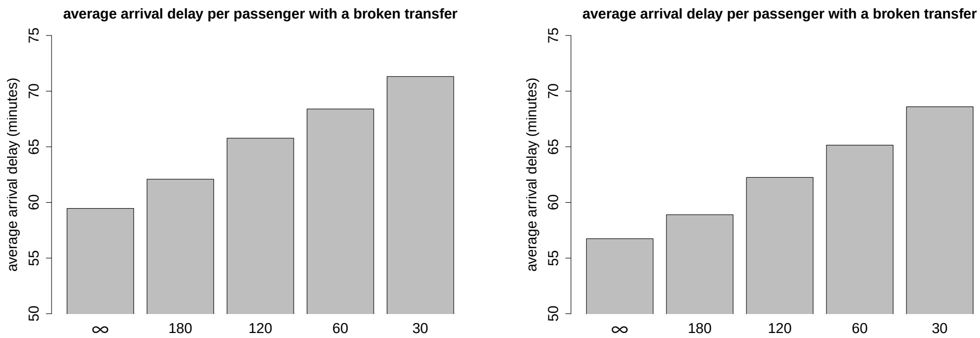


Figure 9 Experiment 2, further test days: April 16 (left) and September 12 (right), 2013. Average arrival delay at the destination if a planned transfer breaks. Rerouting is applied either immediately (denoted by ∞) or (at most) 180, 120, 60, or 30 minutes before the planned departure of the broken transfer.

it. For each selected transfer we introduce a single artificial delay, chosen large enough that this transfer breaks. Then, for each case we run several alternative strategies. The first alternative immediately reroutes all passengers who have planned to use the selected transfer when the delay information becomes available. The other alternatives wait with the rerouting until x minutes (or less) before the event takes place, where we use $x \in \{30, 60, 120, 180\}$. In all cases we measure the final delay of passengers at their destination. Fig. 8 shows exemplarily for one specific day (September 13, 2013), that the average delay is increasing the longer we wait with rerouting. The best average value of 54 minutes is obtained when rerouting is applied immediately, while waiting until 30 minutes before the planned departure of the connecting train leads to an average delay of 66 minutes. While the average delay values depend on the chosen day, the clear trend is confirmed for other test days, too (Fig. 9): the earlier we are able to reroute, the more beneficial it is for passengers.

On an ordinary test day with real delays (September 13, 2013) more than 20,000 passengers have been rerouted due to transfers classified as BREAK. Again we applied the strategies described above: Either we reroute passengers immediately (denoted by ∞) or

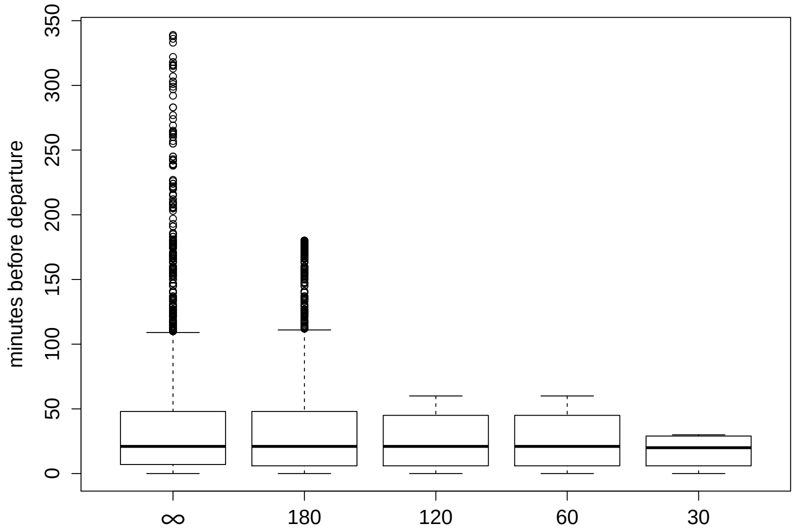


Figure 10 Rerouting is applied either immediately (denoted by ∞) or (at most) 180, 120, 60, or 30 minutes before the planned departure of the broken transfer. For each strategy, the box-plots show the distributions of the moment in time before the planned departure where rerouting is applied for real delay data on September 13, 2013.

(at most) 180, 120, 60, or 30 minutes before the planned departure of a transfer that has been classified as BREAK. For this scenario we observed only relatively small *average* savings in travel time for the rerouted passengers. Therefore, we evaluated how many minutes in advance the corresponding strategy has actually done the rerouting operations. Fig. 10 shows the corresponding distributions for each strategy as box plots (showing quantiles and outliers). We observe that for all strategies, more than 75% of all reroutings have been applied within the last 50 minutes before the planned transfer. It means that for the majority of passengers all strategies do essentially the same. This at least partially explains why the average savings in travel time by early rerouting (the average taken over all rerouted passengers) is small. About 1300 passengers can be rerouted at least 120 minutes in advance. Hence, several hundreds of passengers have the chance to profit from early rerouting.

5 Conclusion and Further Research

Timing of decisions is an important challenging aspect of train disposition. This study can be seen as a first step towards rethinking the disposition process. Our computational results suggest that rerouting of passengers should be applied as soon as possible whenever qualified information that a transfer is going to break or that a train is cancelled becomes available.

In this study, we assume for simplicity that in case of severe delays passengers can be rerouted without any restriction. In practice, there are some legal issues with respect to rerouting of passengers. Namely, booked tickets may be only valid for the reserved train or for a subset of train classes.

Another important aspect neglected in this study concerns train capacities. We simply reroute passengers to the quickest connection towards their destination. This may be problematic in case of already crowded trains. We clearly should avoid to reroute passengers to overfull trains whenever possible. From an algorithmic point of view, capacity-aware rerouting can be achieved with network flow techniques. Moreover, one may consider reliability as a further criterion for the search of alternative routes. Finally, we plan to improve the classification methods by adapting them further to specific train routes and time of the day.

Acknowledgements. The authors have to thank Deutsche Bahn AG for supporting this research and providing test data for scientific use. Our thanks also go to the group of Karsten Weihe and Mathias Schnee at Technische Universität Darmstadt for sharing their code of MOTIS with us.

References

1 L. Anderegg, P. Penna, and P. Widmayer. Online train disposition: to wait or not to wait? *ATMOS'02, ICALP 2002 Satellite Workshop on Algorithmic Methods and Models for Optimization of Railways, Electronic Notes in Theoretical Computer Science*, 66(6), 2002.

2 R. Bauer and A. Schöbel. Rules of thumb — practical online strategies for delay management. Technical report, NAM Report, Göttingen, 2012.

3 M. Bender, S. Büttner, and S.O. Krumke. Online delay management on a single train line: beyond competitive analysis. *Public Transport*, 5:243–266, 2013.

4 A. Berger, C. Blaar, A. Gebhardt, M. Müller-Hannemann, and M. Schnee. Passenger flow-oriented train disposition. In C. Demetrescu and M. M. Halldórsson, editors, *Proceedings of the 19th Annual European Symposium on Algorithms (ESA)*, volume 6942 of *Lecture Notes in Computer Science*, pages 227–238. Springer, 2011.

5 A. Berger, A. Gebhardt, M. Müller-Hannemann, and M. Ostrowski. Stochastic delay prediction in large train networks. In *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS)*, volume 20 of *OpenAccess Series in Informatics (OASIcs)*, pages 100–111. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011.

6 A. Berger, R. Hoffmann, U. Lorenz, and S. Stiller. Online railway delay management: Hardness, simulation and computation. *Simulation*, 87(7):616–629, 2011.

7 C. Biederbick. *Computergestützte Disposition im schienengebundenen Personentransport: ein kundenorientierter Ansatz.* PhD thesis, Universität Paderborn, 2006.

8 C. Biederbick and L. Suhl. Decision support tools for customer-oriented dispatching. In F. Geraets, L.G. Kroon, A. Schoebel, D. Wagner, and C. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, pages 171–183. Springer, 2007.

9 S. Cicerone, G. Di Stefano, M. Schachtebeck, and A. Schöbel. Multi-stage recovery robustness for optimization problems: A new concept for planning under disturbances. *Information Sciences*, 190:107–126, 2012.

10 A. D'Ariano. *Improving Real-Time Train Dispatching: Models, Algorithms and Applications.* PhD thesis, Technische Universiteit Delft, 2008.

11 A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real time. *Transportation Science*, 42(4):405–419, 2008.

12 T. Dollevoet and D. Huisman. Fast heuristics for delay management with passenger rerouting. *Public Transport*, 2013. `http://link.springer.com/article/10.1007%2Fs12469-013-0076-6#page-1`.

13 T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management with re-routing of passengers. In J. Clausen and G. Di Stefano, editors, *9th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS)*, OpenAccess Series in Informatics (OASIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2009.

14 M. Gatto, B. Glaus, R. Jacob, L. Peeters, and P. Widmayer. Railway delay management: Exploring its algorithmic complexity. In T. Hagerup and J. Katajainen, editors, *Proceedings of 9th Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 3111 of *Lecture Notes in Computer Science*, pages 199–211. Springer, 2004.

15 M. Gatto, R. Jacob, L. Peeters, and A. Schöbel. The computational complexity of delay management. In D. Kratsch, editor, *Graph-Theoretic Concepts in Computer Science: 31st International Workshop (WG 2005)*, volume 3787 of *Lecture Notes in Computer Science*. Springer, 2005.

16 M. Gatto, R. Jacob, L. Peeters, and P. Widmayer. On-line delay management on a single train line. In F. Geraets, L.G. Kroon, A. Schoebel, D. Wagner, and C. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, pages 306–320. Springer, 2007.

17 J. Jespersen-Groth, D. Potthoff, J. Clausen, D. Huisman, L.G. Kroon, G. Maróti, and M.N. Nielsen. Disruption management in passenger railway transportation. In R. Ahuja, R.-H. Möhring, and C. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, volume 5868 of *Lecture Notes in Computer Science*, pages 399–421. Springer, Heidelberg, 2009.

18 P. Kecman, F. Corman, A. D'Ariano, and R. Goverde. Rescheduling models for railway traffic management in large-scale networks. *Public Transport*, 5:95–123, 2013.

19 P. Kecman and R. M. P. Goverde. Adaptive, data-driven, online prediction of train event times. In *16th IEEE Annual Conference on Intelligent Transportation Systems (ITCS 2013)*, 2013.

20 N. Kliewer and L. Suhl. A note on the online nature of the railway delay management problem. *Networks*, 57:28–37, 2011.

21 S.O. Krumke, C. Thielen, and C. Zeck. Extensions to online delay management on a single train line: new bounds for delay minimization and profit maximization. *Mathematical Methods of Operations Research*, 74(1):53–75, 2011.

22 S. Kurby. *Makroskopisches Echtzeitdispositionsmodell zur Lösung von Anschlusskonflikten im Eisenbahnbetrieb*. PhD thesis, Fakultät Verkehrswissenschaften "Friedrich List", Technische Universität Dresden, 2012.

23 M. Müller-Hannemann and M. Schnee. Finding all attractive train connections by multi-criteria Pareto search. In F. Geraets, L. Kroon, A. Schoebel, D. Wagner, and C. Zaroliagis, editors, *Proceedings of the 4th Dagstuhl conference on algorithmic approaches for transportation modelling, optimization, and systems (ATMOS)*, volume 4359 of *Lecture Notes in Computer Science*, pages 246–263. Springer Verlag, 2007.

24 M. Müller-Hannemann and M. Schnee. Efficient timetable information in the presence of delays. In R. Ahuja, R.-H. Möhring, and C. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, volume 5868 of *Lecture Notes in Computer Science*, pages 249–272. Springer, 2009.

25 M. Schachtebeck and A. Schöbel. To wait or not to wait and who goes first? Delay management with priority decisions. *Transportation Science*, 44(3):307–321, 2010.

26 T. Schaer, J. Jacobs, S. Scholl, S. Kurby, A. Schöbel, S. Güttler, and N. Bissantz. DisKon — Laborversion eines flexiblen, modularen und automatischen Dispositionsassistenzsystems. *Eisenbahntechnische Rundschau (ETR)*, 45:809–821, 2005.

27 M. Schmidt. Simultaneous optimization of delay management decisions and passenger routes. *Public Transport*, 5:125–147, 2013.

28 A. Schöbel. A model for the delay management problem based on mixed-integer programming. *Electronic Notes in Theoretical Computer Science*, 50(1), 2001.

29 A. Schöbel. *Customer-oriented optimization in public transportation.* Springer, Berlin, 2006.

30 A. Schöbel. Integer programming approaches for solving the delay management problem. In F. Geraets, L. Kroon, A. Schoebel, D. Wagner, and C. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, pages 145–170. Springer, 2007.