PREDICTION OF TRAIN RUNNING TIMES AND CONFLICTS USING TRACK OCCUPATION DATA

Ir. Dirk J. van der Meer[1]

Dr. Rob M.P. Goverde[1,*]

Prof.Dr.-Ing. Ingo A. Hansen[1]

[1]*Department of Transport and Planning, Delft University of Technology*
Stevinweg 1, 2628 CN Delft, The Netherlands
**Corresponding author, e-mail: r.m.p.goverde@tudelft.nl*

ABSTRACT

This paper presents a train running time prediction model, which takes into account train path conflicts and dispatching decisions. Dependencies between train events are modelled as arcs in a weighted graph. Data mining of historical track occupation data is used to find accurate estimates of the arc weights (running times, dwell times, headway) conditional on current delays and factors such as peak hours. A case study of the Dutch railway corridor Dordrecht–Rotterdam–Den Haag shows good calibration and validation results.

Keywords: running time prediction, delay propagation, track occupation data, train conflicts

INTRODUCTION

In order to reduce delays in a railway system and to prevent them from propagating through the network, a train dispatcher can apply dispatching actions, for example by changing the location of a planned overtaking, or cancelling a train connection. When deciding which dispatching measure to apply, the dispatcher needs to estimate the consequences in advance. This asks for the possibility to predict the train traffic in the near future (e.g. one hour ahead), based on the current state of the railway system. This paper presents a statistical analysis of train running times between stations with the aim to make accurate predictions of the delay propagation over a railway network. The main goal is to develop and test a predictive delay propagation model that obtains its parameters by statistical analysis of historical track occupation data and takes into account possible downstream conflicts.
Over the last decades, much research has been done to improve dispatching procedures during disruptions in railway operations. Models for finding optimal dispatching decisions

have been developed based on integer programming, graph theory, or microscopic simulation using blocking time theory (Hansen and Pachl, 2008). All these methods rely explicitly or implicitly on conflict detection and resolution. However, an online tool for predicting train path conflicts and calculating the network-wide effectiveness of dispatching decisions still has to be developed. This may explain that the developed models and prototype railway traffic management systems still have to find their way into practical implementations.
A railway traffic network consists of many interdependencies due to timetable constraints, passenger connections, infrastructure constraints, logistic constraints such as rolling stock circulations, etc. Such a system can be modelled effectively using a *timed event graph*, which can also be expressed as a linear system in *max-plus algebra* (Goverde, 2005). The graph structure of such a model allows fast calculations of delay propagation (Goverde, 2010), making the model suitable for online use in decision support systems for dispatchers or in dynamic passenger information systems.
However, train operations are affected by many varying factors, such as driver behaviour, passenger volumes, weather conditions, etc. The need for fast running time prediction algorithms on the one hand and the complexity of railway systems on the other hand leads to a trade-off between fast online train traffic prediction and a detailed calculation of the stochastic processes and interdependencies resulting from trains sharing the same railway infrastructure. To tackle this problem an adaptive model can be used where the parameters such as running times and dwell times are estimated using feedback of current and historical operational data. Data mining of railway operations data can be used to find stochastic distributions and dependencies amongst delays, process times, and categorical factors, see e.g. Goverde et al. (2001ab), Goverde (2005), Yuan (2006), Conte (2007), Daamen *et al*. (2009), Flier *et al*. (2009), and Van der Meer *et al*. (2009).
The next section proposes a running time prediction model based on the (timed event) graph modelling with train routes and process times as input parameters. The subsequent section presents a statistical analysis of train running times to discover which parameters should be added as model parameters in the graph model. Thereafter, the model is calibrated and validated in a case study of the heavy used Dutch railway corridor Rotterdam-The Hague. The final section contains the main conclusions.

THE RUNNING TIME PREDICTION MODEL

Dutch train describer data

A train traffic prediction model requires a topological description of the underlying railway network including the measurement locations of actual train positions. A good starting point is the network configuration of train describer systems. The progress of trains over a railway network is followed by train describer systems using track occupation detection messages from the safety systems. Both incoming infrastructure messages and generated train describer messages are typically recorded in train describer logfiles in real-time. Goverde and Hansen (2000) developed the *TNV-Prepare* data mining tool for Dutch train describer records (TNV-logfiles) to match infrastructure messages to train numbers and thus obtaining train path realizations on a track section level with a precision of one second. The output of

this tool has been used for operations analysis and statistical inference (Goverde, 2001ab, 2005; Nie and Hansen, 2003; Yuan and Hansen, 2007). More recently, Daamen *et al*. (2009) developed the data mining tool *TNV-Conflict* which in addition to the train trajectories and train delays also identifies realized route conflicts and conflicting trains using object-oriented programming and signalling logic. Using this tool unhindered and hindered running times are distinguished and the effect of route conflicts can be analysed effectively. Prediction of running times and conflicts is the next logical step based on insight gained from the descriptive and inferential statistics (Van der Meer *et al*., 2009).
The data used in this report are obtained by TNV-Conflict using the train describer logfiles of the areas Rotterdam and The Hague of February 2009. The graph model is based on the topological description of the railway network according to the train describer configurations. The railway network is divided into track sections by track-free detection devices (track circuits, axle counters). Each time a train enters or clears such a track section, an infrastructure message is received by the train describer system and recorded in a logfile. Figure 1 shows an example of the division of railway infrastructure into track sections, where the underlined codes are examples of section names. Note the difference between a track section and a *block section*, which always starts and ends at a signal. The high level of detail of these train describer data and the accuracy of one second allows a detailed calculation of train running times between given elements in the network. This microscopic level is used to find statistical dependencies and parameter estimates offline. For the online prediction model an aggregated network structure is used corresponding to scheduled station events (arrivals, departures and through passages) and the train position steps from the train describers that are available online.

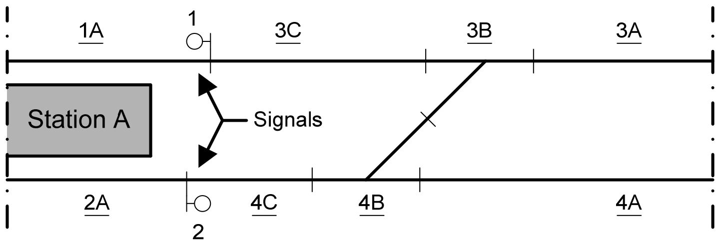


Figure 1 Small example of railway infrastructure elements: track sections and signals

Structure of the prediction model

In the model proposed in this paper, train operations are described by a set of *processes*, which take a certain amount of time, and *events*, which form the beginning or the end of a process. Examples of process times are: train running times, dwell times, waiting times caused by conflicting train routes, etc. In the online macroscopic prediction model the events are one of three types: train departures, train arrivals, or train passages at stations.
The dependencies between events and processes can be graphically represented by *timed event graphs.* As shown by Goverde (2005), this is a suitable way to model railway systems if the propagation of train delays has to be calculated quickly. For a correct prediction of train running times, it is necessary that dependencies between trains due to conflicting train routes are incorporated in the model. Such a dependency is modelled by an arc (i, j), connecting the dependent events i and j. The weight of the arc represents the minimum time duration

between events *i* and *j*. A railway system can thus be modelled by a set of events and a list of arcs, which together form a timed event graph. The graph is built by scanning the sectional train data as generated by TNV-Conflict from the raw train describer logfiles. This will be described in the next section.

Generating the model from track occupation data

By mining the track occupation data, a timed event graph of all recorded train operations can be generated. In this project, such a model will serve two purposes: the first half of the data will be used to calculate parameters, such as running times, blocking times, etc., while the second half of the data will be used to generate test cases, in order to test the accuracy of the model.

Track section occupations by trains recorded in the data lead to events in the model, but to keep the model simple, not all section occupations will be translated to events. Therefore, the track occupation data is filtered for section occupations listed in a set *S*, being the sections where the events occur that are to be modelled in the final model. At some sections, the activity is dependent on the train type. For example, the occupation of a track section in a small station gives a through event for intercity trains and a departure event for local trains. To make this distinction correctly while scanning the data, a line number *Line* is assigned to each train run, while the activity for each train line *Line* at a section *S* is stored in a matrix *Act*(*S*,*Line*), with entries that can attain the values 'arrival', 'through' or 'departure'. For each section occupation by a train number, it is now clearly defined whether a train is arriving, departing or passing.

Scanning the data for occupations of track sections in *S* results in a set of events *E*. For each event *i* the following attributes are determined: scheduled event time $T_{\text{scheduled}}(i)$, recorded event time $T_{\text{recorded}}(i)$, train number *N*(*i*), event type *Activity*(*i*), and rolling stock type *Stock*(*i*).

After the track occupation data has been scanned for events an arc list *A* is generated, containing all arcs modelling delay dependencies between the events in de model. The arcs modelling running times and dwell times are generated as follows. First all events with the same train number are sorted on date and time of occurrence. After sorting, all pairs of subsequent events are connected by arcs (*i*, *j*). If *Activity* (*i*) = 'departure' and *Activity* (*j*) = 'arrival', then (*i*, *j*) is a running time arc, whereas in case of the opposite, the arc reflects a dwell time at a station. Arcs ending or starting at a 'through' event are always running time arcs.

Creating the headway arcs is more complicated, since different types of conflicting train routes (e.g. crossing routes or routes merging into the same track) result in different headways that have to be implemented in the model as headway arcs. Headway arcs reflect the minimum headway that has to be respected between two events of trains with conflicting routes. The direction of the headway arc is determined by the order in which the trains pass the shared infrastructure element. Headways between following, merging or crossing trains are distinguished. In order to check which type of headway arc has to be generated, the track occupation data is scanned for train passages at bottlenecks in the network, simultaneously to the scanning for events. The set of bottleneck sections *B* has to be defined in advance, such that the resulting model will reflect the train operations and the occurring delay propagations as accurately as possible. Note that detailed knowledge of the infrastructure is crucial to define the bottleneck sections correctly, as the quality of the models generated by the algorithm mainly depends on a tactical definition of the set *S* and the bottleneck sections in the set *B*.

Each time an occupation of a bottleneck section is found in the data, a headway arc is added to the model as shown in Figure 2, according to the following rules:

- A 'following' bottleneck models the fact that a train cannot arrive at a station as long as its (platform) track is still occupied by a previous train. This leads to a headway arc connecting the departure of the previous train with the arrival of the next train.
- A 'merging' bottleneck models the headway between trains with routes merging into the same track. This leads to a headway arc between the events of two subsequent trains, preceding the passage of the bottleneck.
- A 'crossing' bottleneck models the delay dependency between crossing trains, where a delay of the preceding train is propagated to a following train. This leads to an arc between the predecessor event of the first train and the successor event of the following train, with regards to the bottleneck passage. For example, in Figure 2 (right picture) two trains 1 and 2 pass through station A, cross each other, and then pass through station B. Here, a delay of train 1 at station A can be propagated to train 2 at station B.

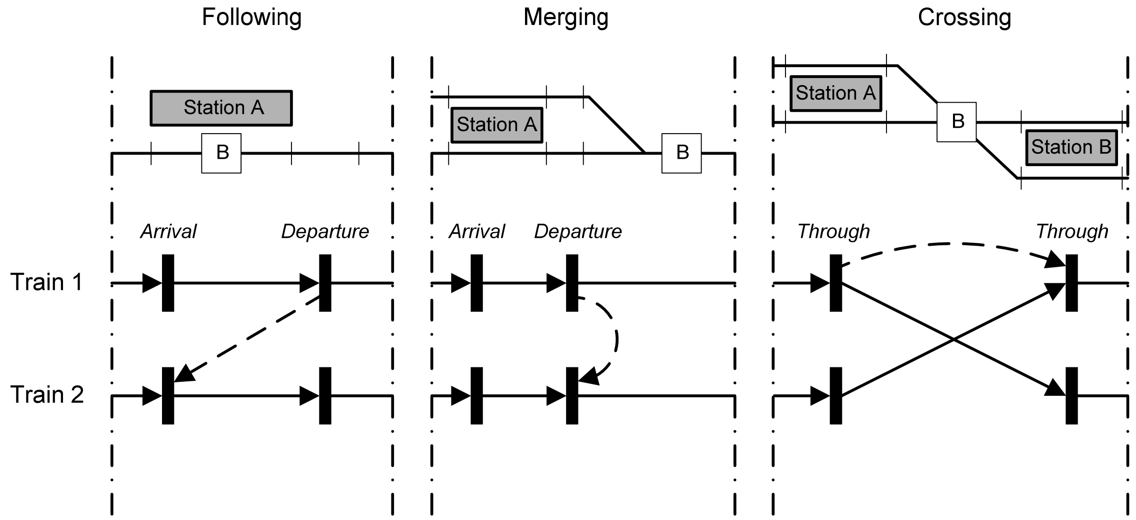


Figure 2 Bottlenecks (denoted by 'B') and the resulting headway arcs in a timed event graph (events are denoted by black bars, running time and dwell time arcs are solid, headway arcs are dashed)

Algorithm 1 gives the pseudo-code for the model construction. Arcs are generated while scanning the set of events *E* and checking for bottleneck passages (in line 4). If two subsequent events in the (sorted) set *E* are associated to the same train number, then a running/dwell time arc is constructed (line 3). Headway arcs caused by the different types of bottlenecks are constructed in lines 6 – 14. If a section code is listed both in set *S* (for defining events) and in set *B* (for defining bottleneck passages), then the algorithm treats the bottleneck passage as event *i*, and the subsequent event as event *j*. This situation can be seen in the example of a 'following' bottleneck in Figure 2.

Event dependencies due to passenger transfers, turning trains, coupling and uncoupling trains, crew schedules, etc., can be modelled using the same principle.

Algorithm 1 **(GENARCS)**

```
Input:
   E      = list of events, sorted by occurrence time t, then to train number n
   B      = list of bottleneck sections b
Output:
   A      = arc list
for each pair (i, j) of subsequent events in E do
   if N (i) = N (j) then                          % i and j belong to the same train run
      A ← A ∪ {i, j};                             % add running/dwell time arc from i to j
      if a bottleneck b was passed between i and j then
         find previous passage of b;
         if type(b) = 'crossing' then
            find the predecessor event k of the previous passage of b;
            A ← A ∪ {k, j} ;                      % add headway arc from k to j
         else if type(b) = 'following' then
            find the successor event k to the previous passage of b;
            A ← A ∪ {k, i} ;                      % add headway arc from k to i
         else if type(b) = 'merging' then
            find the predecessor event k to the previous passage of b;
            A ← A ∪ {k, i} ;                      % add headway arc from k to i
return A;
```

After the arc list has been generated the arc weights have to be calculated. Since the arc weight of an arc (i, j) reflects the *minimum* time that has to elapse after event i before event j can occur, the most simple way to estimate an arc weight is to calculate a small percentile of all observed arc weights in the track occupation data. Different percentiles may be used for this depending on the process that has to be modelled. The remainder of the paper is devoted to find good estimates of the arc weights, with train running times analysed in more detail.

ANALYSIS OF TRAIN RUNNING TIMES

The railway traffic model presented in the previous section can be used to predict the delay propagation based on the current state of the railway traffic. However, apart from the current delays, other factors may influence the running times and dwell times of trains, thereby affecting the delay propagation. The influence of a number of parameters on train running time is investigated in this section, but first the different components of scheduled running time are introduced.

The scheduled running time between two main stations consists of four components (Hansen and Pachl, 2008):

- *Minimum running time between scheduled stops*
  This is the shortest possible running time of a train between two stops. This pure running time can be derived from running time calculations or from practical experience.
- *Dwell time at intermediate scheduled short stops*
  In case of intermediate stops, dwell time is included in the scheduled running time.
- *Running time supplement*
  A running time supplement is added to the minimum running time to enable trains to make up small delays. Usually, the running time supplement is a certain percentage of

the minimum running time. When a train runs according to the schedule, the running time supplement can be used differently depending on the train driver. The usual approach is to adopt a more energy-efficient driving style by reducing maximum speed or coasting, thus avoiding early arrivals and the risk of conflicts at the next station. Note that the running time supplement can not always be used fully to recover from delays, since sometimes it has to be used to make up for less ideal circumstances, such as bad weather, etc.

- *Scheduled waiting time*
  Scheduled waiting time is sometimes added to the schedule in order to solve a train path conflict or to synchronize train departures to enable passenger transfers. Scheduled waiting time may therefore be added to either dwell time or running time.

A departure delay may influence the running time of that train. Two mechanisms causing this influence can be distinguished: (a) the train driver of a delayed train will try to use the running time supplement by running at the maximum speed allowed in order to reduce the delay, and (b) at intermediate stops delayed trains require less time (if at all) waiting for their scheduled departure time. The influence of current delays is studied below for the successive cases of non-stop running times, dwell times, and running times including intermediate scheduled short stops.

Influence of delay on non-stop train runs

To show the influence of the first mechanism, the running times of intercity trains from Zwijndrecht (a through station between Dordrecht and Rotterdam) to Rotterdam Centraal are investigated. This route has been chosen for two reasons: (1) two tracks per direction are available all the way from Zwijndrecht to Rotterdam Centraal, so running times are less likely to be influenced by train conflicts, and (2) at Zwijndrecht the trains just crossed the movable railway bridge near Dordrecht, so the running times are not influenced by bridge openings.
The running times are calculated by subtracting the occupation time of the platform track at Rotterdam Centraal (section 218BT) and the track section directly after the platform track at Zwijndrecht (1335AT). Hindered train runs as detected by TNV-Conflict analysis have been filtered out. Table I lists the statistical characteristics of the data sets for IC1900, IC2100 and IC2400 trains.

Table I Summary of running times for intercity trains between Zwijndrecht and Rotterdam Centraal

Train line	**1900**	**2100**	**2400**
Train route	Vl – Gvc	Vs – Asd	Ddr – Asd
Number of data points	594	340	327
Scheduled running time [s]	660	720	720
Mean [s]	545	557	564
Median [s]	534	538	543
Standard deviation [s]	49	63	58
R^2 of robust regression line	0.0029	0.0517	0.0237

Note that the scheduled running time is calculated until the actual arrival (i.e. standstill at the platform) of the train, whereas the observed running times are calculated using the section occupation time, which causes the observed running times to be shorter than the scheduled running times. Hence, the running times of the arriving trains on the platform section itself (i.e. from the occupation of the last platform track section to standstill) is not taken into account. A lower bound for this value is the time that elapses between the occupation of the platform section and the release time of the last section before the platform, which is the running time over the train length from the first axle to the last axle passing the beginning of the platform track section. For the IC1900 trains for instance, the median of these running times is 35 seconds. Note that the release time of the pre-platform track section is a more accurate arrival time estimate but this depends on the train length. Calculating all running times over the same distance as the difference of section occupation times avoids an additional dependency on train lengths.

Figure 3 shows a scatter plot of the running times of the IC1900 trains depending on the through delays with respect to the scheduled passing time in Zwijndrecht. A robust regression line resisting 25% of the outliers (Rousseeuw and Van Driessen, 2006) has been included to investigate the dependency between the delay and the running time. No such dependency can be recognized, as is also shown by the low value of explained variation R^2 = 0.0029. Apparently, between Zwijndrecht and Rotterdam Centraal, the IC1900 trains cannot use any recovery time to reduce their delays, which means that either there is not enough recovery time in the schedule or that the available recovery time is used for other purposes.

Figure 4 shows a scatter plot and histogram of the IC2100 trains with a scheduled running time of 720 seconds, which is one minute longer than the IC1900. Again, very little linear relationship between the running time and the delay can be found. However, the running times of the punctual IC2100 trains show more variation, which can be explained by the different anticipating behaviour of train drivers. From the scatter plot can be concluded that some train drivers adopt a more slow and energy-efficient driving style if they know their train is running punctual or ahead of schedule while other drivers always run at the maximum allowed speed, depending on their experience, willingness to avoid too early arrivals at Rotterdam Centraal, save energy, etc. The IC2400 trains show the same behaviour as the IC2100 trains.

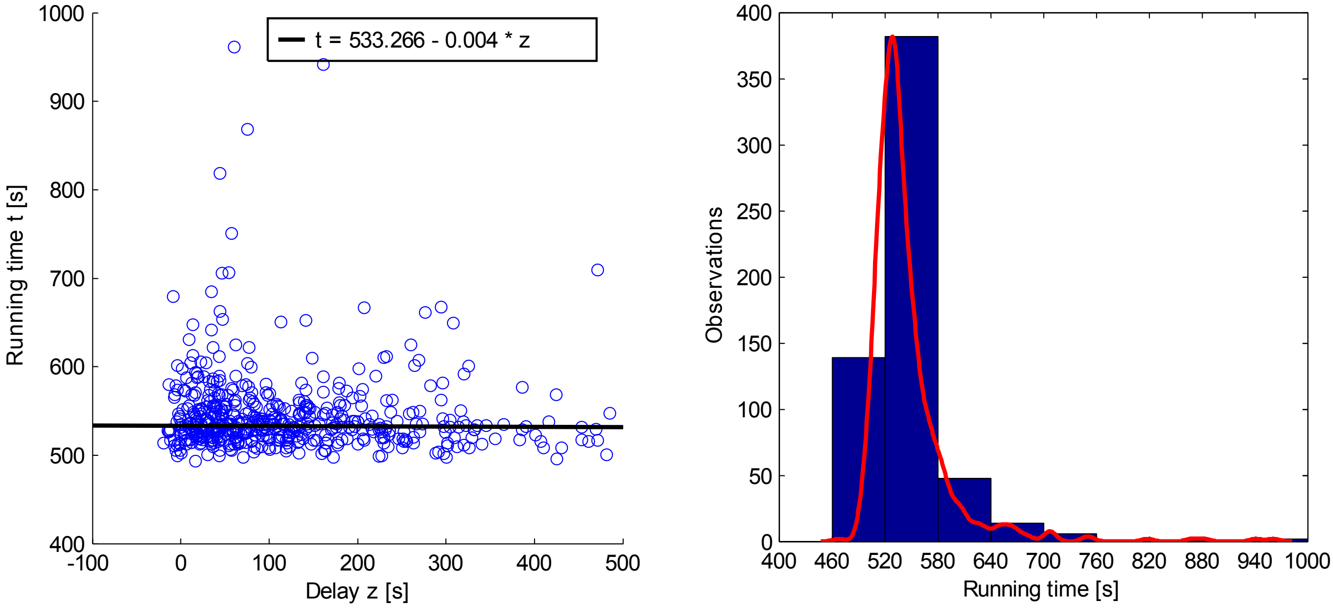
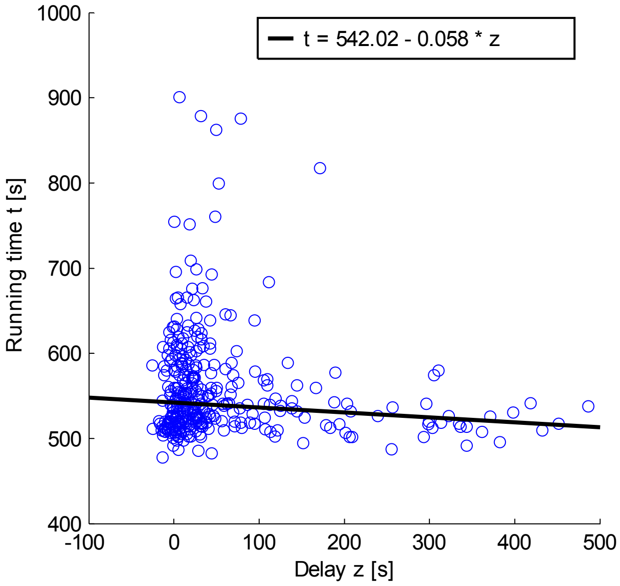


Figure 3 Delay dependency of running times Zwd - Rtd for IC1900 trains



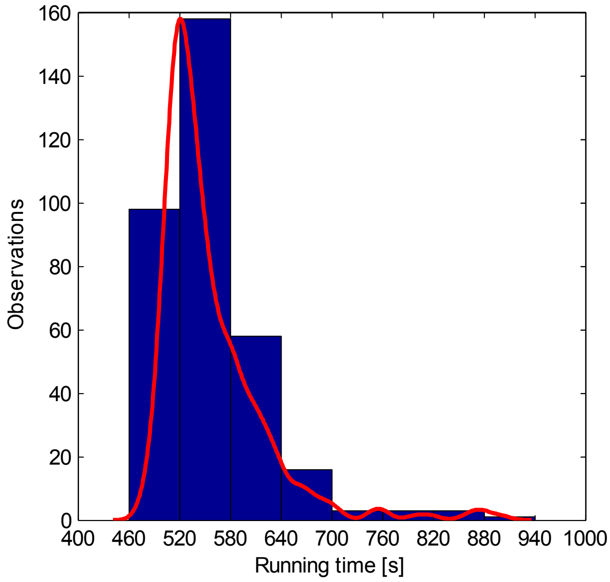


Figure 4 Delay dependency of running times Zwd - Rtd for IC2100 trains

Influence of delay on dwell times

When trains run ahead of schedule, they have to wait on their scheduled departure time at station stops. Therefore, delays have a large influence on dwell times of trains at stations. In this section, the dwell times of the intercity trains at Rotterdam Centraal are calculated by subtracting the occupation times of infrastructure elements 218BT and 278, being the platform track section of track 9 and the signal directly after the platform respectively. The dwell times of the other trains are calculated using the occupation times of sections A280T and 283BT, being the platform track section of track 8 and the first section after the platform.

Table II Dwell times at Rotterdam Centraal

Train line	**IC 1900**	**IC 2100**	**IC 2400**	**S 2200**	**ST 5100**	**ST 5000**
Train route	Vl-Gvc	Vs-Asd	Ddr-Asd	Bd-Asd	Rsd-Gvc	Ddr-Ledn
Number of data poins	596	341	345	549	946	345
Scheduled dwell time [s]	180	120	120	240	180	60
Mean [s]	258	295	326	351	232	173
Median [s]	262	301	340	338	232	155
Standard deviation [s]	75	91	76	104	89	59
R^2 of robust regression line	0.90	0.94	0.97	0.91	0.92	0.61
Estimated min dwell time [s]	179	157	155	140	126	131

Table II summarises the statistical properties of the dwell times at Rotterdam Centraal of trains heading in the direction of Den Haag. From the scatter plots in Figure 5 it can be concluded that a strong relationship exists between the arrival delays and the dwell times of trains, which is also confirmed by the slope of the robust regression line (again, 25% of the outliers were resisted). However, for the model based on timed event graphs, the *minimum* dwell times of trains are important, since the arc weights model the *minimum* time that has to elapse between two connected events. Therefore, a separation has to be made between trains that actually experienced a minimum dwell time, and trains that had to wait for their departure time, leading to longer dwell times.

A robust method to obtain such a separation makes use of the fact that trains experience their minimum dwell time only if they arrived with a certain delay. Good results were obtained by using the scheduled dwell time minus 60 seconds as a limit for this minimum arrival delay. The red dashed vertical line in Figure 5 denotes this limit. The median of the dwell times of all trains with an arrival delay beyond this limit (i.e. right of the vertical, dashed line) is the robust estimation of the minimum dwell time, indicated by the horizontal dashed line. For short stops a small percentile of all dwell times can be used as a robust estimator of the minimum dwell time. Note that the use of section occupation times leads to relatively high values for the minimum dwell times as they include the running time on the platform track.

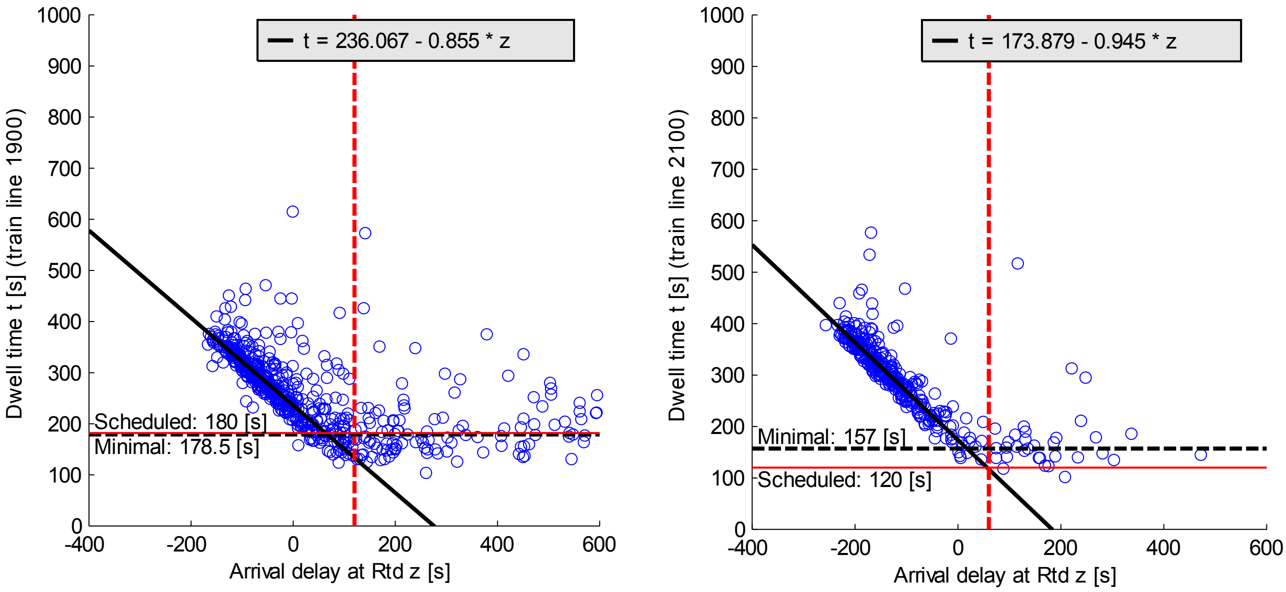


Figure 5 Scatter plots of dwell times at Rotterdam Centraal of IC 1900 (left) and IC 2100 (right) trains heading in the direction of Den Haag

Influence of delay on train runs with intermediate scheduled short stops

The previous sections showed that delays affect dwell times of trains, whereas pure running times may be almost unaffected by delays. In this section, the influence of delays on train runs with intermediate stops will be investigated for the S2200, ST5100 and ST5000 trains between Zwijndrecht and Rotterdam Centraal. The results are summarized in Table III.

Table III Running times Zwijndrecht - Rotterdam Centraal of trains with intermediate stops

Train line	**S2200**	**ST5100**	**ST5000**
Train route	Bd-Asd	Rsd-Gvc	Ddr-Ledn
Number of stops	2	4	4
Number of data poins	555	948	345
Mean [s]	889	1086	1124
Median [s]	902	1082	1120
Standard deviation [s]	78	71	57
R^2 of robust regression line	0.99	0.30	0.70
Scheduled running time [s]	1020	1140	1200
Estimated min. running time [s]	787	1015	1078
Running time supplement %	23	11	10

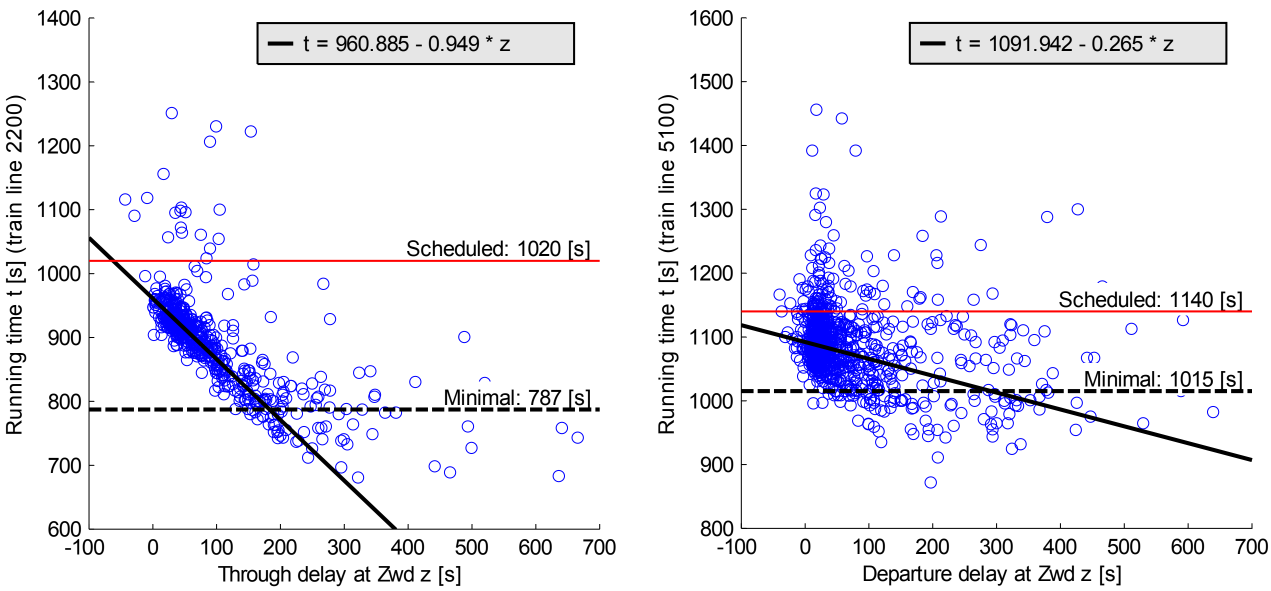


Figure 6 Delay dependency of running times Zwijndrecht – Rotterdam Centraal (left: S2200, right: ST5100)

As can be seen in the scatter plots of Figure 6, the dependency between delays and running times (including intermediate stops) is not always as strong as with dwell times. This can be explained by the different amount of slack time in the schedules of both trains. Apparently the S2200 has more slack time, enabling compensation of delays up to approximately 200 seconds. Another result is that trains of the same train line may have a different delay dependency on different segments of their route. For instance, the robust regression model of the S2200 trains between Zwijndrecht and Rotterdam has explained variation $R^2 = 0.99$, while between Rotterdam and Den Haag HS this is $R^2 = 0.18$ (Van der Meer *et al*., 2009).
Consequently, finding a robust estimator for the minimum running time using delay dependency is more difficult. It is therefore advised to use a fixed percentile as an estimator for the minimum running time. The dashed lines in Figure 6 denote the 10th percentile, which yields a good estimation for each of the train lines, as can be seen in the figures.
An estimation of the running time supplement can be made from the difference between the scheduled running time and the estimated minimum running time, which can be found in Table III as well.

Influence of peak hours

During peak hours more boarding and alighting passengers may lead to longer minimum dwell times. In the track occupation data, this effect only becomes visible for trains with short stops, or for late arriving trains where the minimum dwell time could be measured. As can be seen in Table IV, where the statistics of dwell times at Rotterdam Centraal during peak and off-peak hours are summarized, the number of trains running in the peak hours with an arrival delay more than the scheduled dwell time minus 60 seconds, is small compared to the total data set. This makes it difficult to draw statistical conclusions about the influence of peak hours on dwell times at Rotterdam Centraal. Merging the data sets of similar train lines may be a solution to this problem. For instance, the train lines ST5100 and ST5000 have the same stopping pattern, rolling stock, etc., and so have the IC2100 and the IC2400.

Table IV Peak and off-peak dwell times at Rotterdam Centraal

Train line	**IC 1900**	**IC 2100**	**IC 2400**	**S 2200**	**ST 5100**	**ST 5000**
Train route	Vl-Gvc	Vs-Asd	Ddr-Asd	Bd-Asd	Rsd-Gvc	Ddr-Ledn
Scheduled dwell time [s]	180	120	120	240	180	60
N peak and arrival delay	16	6	3	2	28	9
N off-peak and arrival delay	128	29	11	13	88	38
p-value rank-sum test	0.025	0.015	0.071	0.93	0	0.51
Med. peak and delayed [s]	215	180.5	201	142.5	148	138
Med. off-peak and delayed [s]	176.5	152	148	140	120	125

Despite the small numbers of data points, the scatter plots in Figure 7 indicate that the minimum dwell times during peak hours are longer than in the off-peak. The medians of peak-hour and off-peak dwell times, both only for trains with an arrival delay beyond the red dashed line, are given in Table IV and show that the peak-hour dwell times indeed tend to be longer than off-peak dwell times. A Wilcoxon rank-sum test is performed to investigate the difference between both sets of dwell times as well. At a significance level of 5%, statistical evidence for different peak and off-peak dwell times is found for the IC1900, IC2100 and ST5100 trains. It should be noted however that although the Wilcoxon rank-sum test can be used for different sample sizes, the lack of balance in the sample sizes of this analysis may lead to unreliable outcomes, particularly for the S2200 and IC2400 trains.

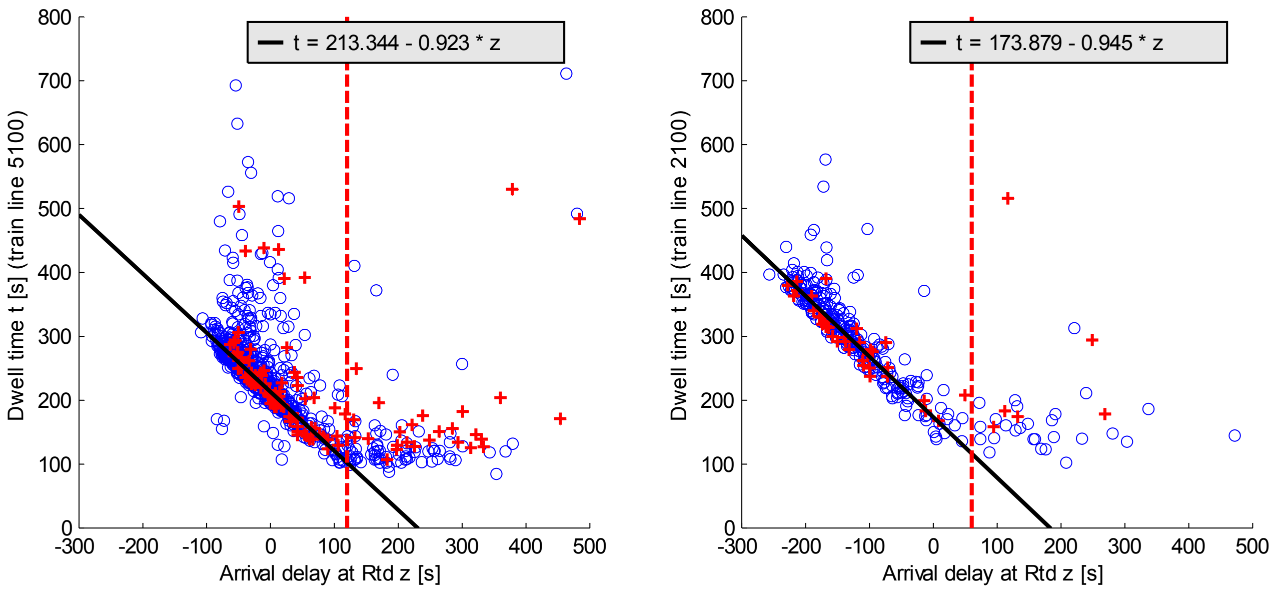


Figure 7 Dwell times in peak hours (denoted with '+'), compared to off-peak dwell times (denoted with 'o') for two train lines at Rotterdam Centraal

More useful results for the analysis of the influence of peak hours were obtained from the busy station of Delft, where short stops are scheduled. No statistical evidence is necessary to show the clear peak hour dependency in Figure 8, where the dwell times of S2200 trains in February 2009 are shown. The dwell times have been estimated by subtracting the occupation time of the platform track section (14A/BT) and the section directly thereafter (34AT). It can be concluded that peak hour dependencies themselves depend on many factors, such as the train line, station and the scheduled dwell time.

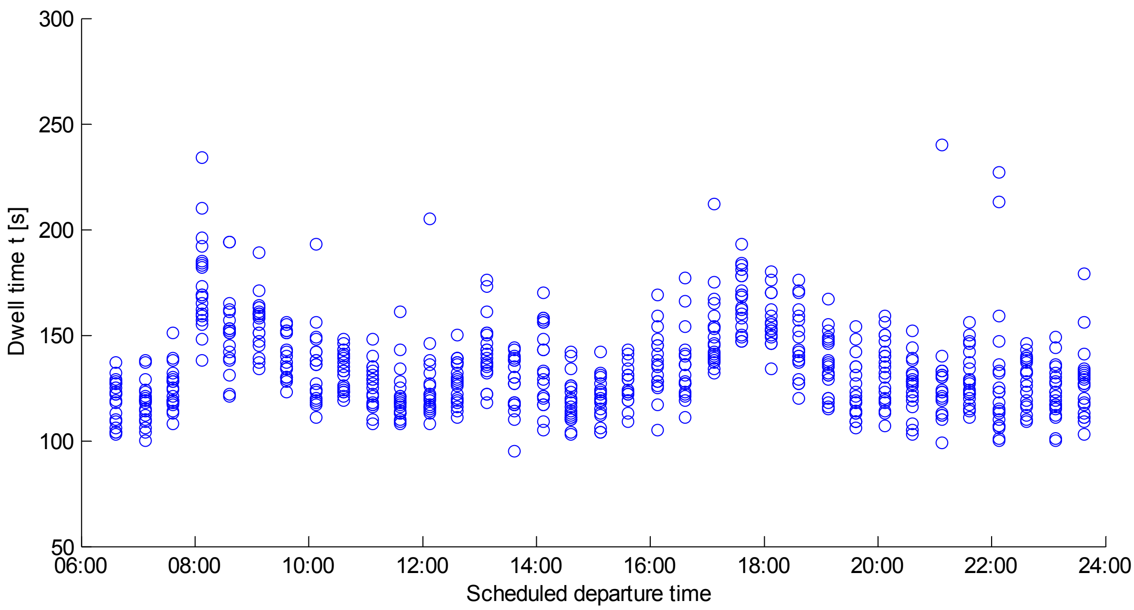


Figure 8 Dwell times (in seconds) of S2200 trains at Delft in February 2009. Note that a 'short stop' is scheduled at Delft for this train line

Influence of rolling stock and weather

The influence of different rolling stock and rainy weather has also been analysed using additional rolling stock and weather data. In both cases statistical significant differences could be found but the values are so small that implementation of these factors in a running time prediction model is not justified (Van der Meer *et al*., 2009).

VALIDATING THE PREDICTION MODEL

Delay propagation algorithm

A delay propagation algorithm has been developed in order to use the model described above for predicting train running times. Half of the available track occupation data has been used to calculate the model parameters (i.e., the set of arc weights W), whereas the other half of the data has been used as a test case for validation.
When testing the prediction model, it has to be carefully taken into account which information is assumed 'known' by the model, and which information is not, see Figure 9. Before the start of the experiment, a prediction time horizon is chosen. Based on this time horizon, the starting point of the prediction process can be found for each train that is to be predicted. For example, if the prediction time horizon is 20 minutes, and a train arrives at 17:28 according to the track occupation data, then the starting point t_{start} for the prediction of that arrival will be at 17:08. In the figure can be seen that all event times later than t_{start} are considered as 'future', and therefore they are unknown and have to be predicted. The running orders, scheduled event times and parameters like rolling stock are assumed known. The choice to regard the order in which the trains will run downstream (i.e., in the future) as known may not seem logical. However, if the model is applied in an online decision support system to evaluate

dispatching decisions, the running orders of trains are *input* for the predictive model. Hence, for use in online dispatching systems future train orders are controlled by the dispatcher (or a control system) and do not have to be predicted.

	$t < t_{start}$ 'past'	t_{start}	$t > t_{start}$ 'future'	$t_{predicted\ event}$
Event times	known		unknown	
Scheduled event times	known		known	
Running orders	known		known	
Other (e.g. rolling stock)	known		known	

Prediction horizon

Figure 9 Set up for the prediction model

The prediction horizon is taken into account in Algorithm 2, which makes the actual running time predictions by calculating the delay propagation. The delay propagation algorithm runs backwards in time. Starting with the event to be predicted the algorithm recursively goes through the predecessor events until the starting point t_{start} has been reached at which all event times are known. Then from these known event times the algorithm determines the successive event times forward in time again back to the event time to be predicted. In this procedure only the events relevant for the prediction are visited and predicted as well.
The input to the algorithm consists of the set E containing all events as found in the track occupation data, the arc list A as generated by Algorithm 1, the set of arc weights W, the delay vector z which initially contains all known delays of events that occurred before t_{start}, the minimal delays z_{min}, the event j to be predicted, and the prediction starting time t_{start}, The minimal delays z_{min} correspond to the fact that trains cannot depart before their scheduled running time. They are estimated off-line for each event by calculating a small percentile of all observed delays. This has to be done for the following reasons:

- The fact that departing trains always wait for their scheduled departure time has to be modelled. Since the model in this project uses section occupation times rather than the actual departure times, this waiting behaviour has to be modelled by a minimal delay z_{min} for the occupation time of the first section after the platform, such that the time between the actual departure and the occupation of the first section after the platform is taken into account.
- Train runs with intermediate stops may be aggregated into one arc. The fact that such trains wait for their scheduled departure time at their intermediate stops, when running ahead of schedule, is modelled by the value of z_{min} at the arrival events of such trains.

Note that for non-stop train runs z_{min} may be redundant, as the minimum running times of such trains directly imply a minimal delay at their next downstream arrivals.
The algorithm works as follows. In line 1, the delay of the current event j is updated according to the minimal delay z_{min}. Line 2 checks whether the current event occurred in the past (and is known) or in the future. $T_{recorded}(j)$ is the observed occurrence time of event j, as recorded in the track occupation data. If the event occurred in the past, its delay can easily be calculated in line 3, and the algorithm returns. For events in the future, the prediction process starts at line 5 in a loop for all incoming arcs of event j, since all predecessor events i may propagate a delay to event j. In line 6, the appropriate arc weight w is found in the set

W of arc weights that have been calculated offline. In line 7, the algorithm calls itself, but now for the preceding event i, such that all events preceding the event to be predicted are found, until t_{start} is reached. The actual delay propagation takes place at line 8, where the delay of event j is updated using the delay of the preceding event i and the arc weight w. The output of the algorithm is an updated delay vector z.

Algorithm 2 **(PROPAGATE)**

Input:
E = set of events
A = arc list
W = set of arc weights
z_{min} = vector of smallest delays
z = delay vector
j = event time to be predicted
t_{start} = starting time of prediction
Output:
z = predicted delay vector

1. $z(j) \leftarrow \max(z(j), z_{min}(j))$;
2. **if** $T_{recorded}(j) <= t_{start}$ **then**
3. $\quad z(j) = T_{recorded}(j) - T_{scheduled}(j)$;
4. **else**
5. $\quad$ **for** each incoming arc (i, j) in A **do**
6. $\quad\quad$ find the arc weight w in W based on events i and j;
7. $\quad\quad$ $z \leftarrow$ propagate(E, A, W, z_{min}, z, i, t_{start}); % call algorithm for preceding event i
8. $\quad\quad$ $z(j) \leftarrow \max(z(j), z(i) + w)$;
9. **return** z;

Test case Rotterdam Centraal – Den Haag HS

As a test case for the model, the railway line Rotterdam Centraal – Den Haag HS is used. The train running times on this line (IC trains: approximately 20 minutes, local trains approximately 30 minutes) allow a realistic prediction time horizon of 20 – 25 minutes. Furthermore, Rotterdam Centraal – Den Haag HS is one of the busiest railway lines in the Netherlands, and train conflicts at Rotterdam Centraal will arise easily in case of delays since 4 tracks merge into 2 tracks here, making the case interesting for running time predictions with regards to blocking times and conflicts being taken into account. The model data consists of track occupation data of both the Rotterdam and the Den Haag area of May 2007. The model has been generated using Algorithm 1, as described above. Figure 10 shows the infrastructure elements used to construct the model. To keep the example clear, the surrounding tracks, switches and stations are not shown.

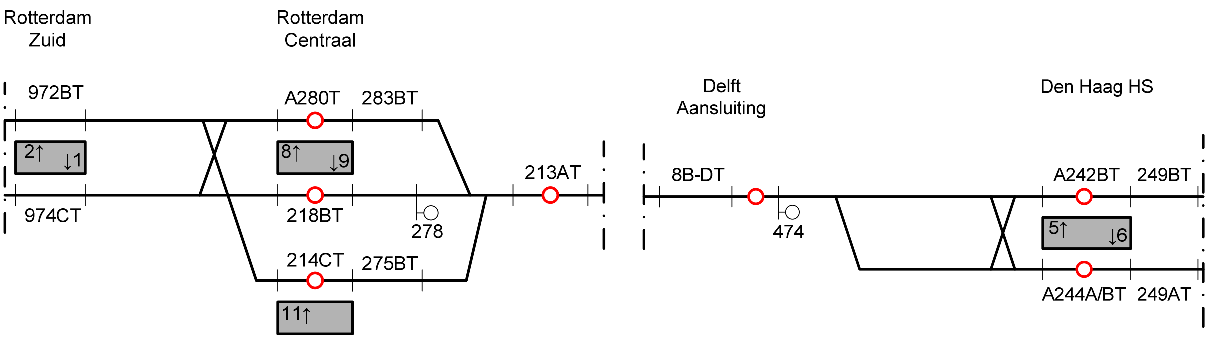


Figure 10 Test case Rotterdam Centraal - Den Haag HS.

Events are found by scanning the track occupation data for section occupations, as mentioned before. The set *S*, i.e. the codes of the infrastructure elements used for this are listed in Table V, along with the activities coupled to the section occupations. Signal 278 has been used to record the departure times of trains from track 9 at Rotterdam Centraal, since the first section after this platform is a shared switch with other train routes, which could lead to getting the wrong departure event times for track 9 corresponding to a train from another platform track. The red circles in Figure 10 denote the bottleneck sections, which are listed in Table VI.

Table V Infrastructure elements defined as events

Section codes	Trains	Modelled activity
972BT, 974CT	Local trains 5000 and 5100	arrival at Rotterdam Zuid
	Express train 2200	passing through Rotterdam Zuid
	IC and international trains	passing through Rotterdam Zuid
A280T, 218BT, 214CT	All trains	arrival at Rotterdam Centraal
283BT, 278, 275BT	All trains	departure from Rotterdam Centraal
8B-DT	All trains	passing through Delft Aansluiting
A242BT, A244A/BT	All trains	arrival at Den Haag HS
249BT, 249AT	All trains	departure from Den Haag HS

Table VI Infrastructure elements defined as bottlenecks.

Section codes	Modelled conflicting operations
A280T, 218BT, 214CT	Following conflict
213AT	Merging conflict
8B-DT	Following conflict between passing events
A242BT, A244A/BT	Following conflict

Predictions with fixed arc weights

The first test of the prediction model is carried out with fixed arc weights. This means that all minimum process times in the model are independent of delays, peak hours, etc. The arc weights, being the shortest times in which the according processes can be completed, have been calculated using the data 1 may – 18 may 2007. The minimal headways between trains are estimated by the 10^{th} percentile of selected groups of headways observed in the data, and are shown in Table VII. The group numbers in this table refer to the following groups of headways that have been aggregated (see Figure 10 for the track numbers):

1. Headways between a departure and the next arrival at the same track at Rotterdam Centraal (track 8, 9 or 11).
2. Headways between a departure from track 9 and the departure of the next train from track 8, 9 or 11.
3. Headways between a departure from track 8 or 11 and the departure of the next train from track 8, 9 or 11.
4. Headways between trains passing Delft Aansluiting.
5. Headways between a departure and the next arrival at the same track at Den Haag HS (track 5 or 6).

Note that departures from track 9 have been separated from departures from tracks 8 or 11 at Rotterdam Centraal (groups 2 and 3). The reason for this is that trains departing from track 9 can accelerate directly to 80 km/h, whereas the routes from tracks 8 or 11 in the direction of Den Haag have a maximum speed of 40 km/h until the end of the interlocking area, leading to a longer headway time for the next train leaving Rotterdam Centraal after such a route has been used. Headways between different types of trains have not been calculated separately. The reason for this is that the estimation of the minimal headways may become unreliable for combinations of trains that occur only rarely (i.e. unscheduled combinations of following trains due to disruptions or rescheduling actions).

Table VII Arc weights w for headway arcs (i, j)

Leading event *i*	Following event *j*	Group	Min. headway [s]
Departure at 283BT	Arrival at A280T	1	137
Departure at 278	Arrival at 218BT		
Departure at 275BT	Arrival at 214CT		
Departure at 283BT of 275BT	Departure at 283BT, 278 or 275BT	2	141
Departure at 278	Departure at 283BT, 278 or 275BT	3	130
Through at 8B-DT	Through at 8B-DT	4	134
Departure at 249BT	Arrival at A242BT	5	120
Departure at 249AT	Arrival at A244A/BT		

The running times and dwell times are estimated by the 20th percentile of the times observed in the data, which can be found in Table VIII. The only exceptions are the trains which have a scheduled dwell time longer than 1 minute. Recall that the minimum dwell times of trains with a long scheduled dwell time can be estimated by calculating the median of observed dwell times for trains with a sufficiently large arrival delay, which is the approach followed here as well.

Table VIII Arc weights w for running and dwell time arcs (in seconds)

Group **Operation**	Local trains 5100, 5000	Express train 2200	Intercity 1900	Intercities 2100, 2400	International and Thalys
Rtz – Rtd	381	319	158	154	156
Stop Rtd	133	163	174	152	197
Rtd – Dta	926	812	845	538	617
Dta – Gv	441	230	225	222	219
Stop Gv	133	123	142	134	159

The minimal delays z_{min} of all events are determined by calculating the 10th percentile of all observed delays for each (group of) train lines, and can be found in Table IX. These values clearly show that the z_{min} reflect the fact that a departure event can never occur before the scheduled departure time, since all departure events have a positive z_{min}.

Table IX Minimal delays z_{min} calculated using 10th percentiles (in seconds)

Group **Event**	Local trains 5100, 5000	Express train 2200	Intercity 1900	Intercities 2100, 2400	International and Thalys
Arrival/passage Rtz	-100	-60	-35	-202	-1
Arrival Rtd	-69	-78	-104	-219	-80
Departure Rtd	21	25	32	28	45
Passage Dta	-46	-161	-148	-29	-46
Arrival Gv	-135	-158	-101	-158	-111
Departure Gv	21	29	21	21	23

The running time predictions using this model are carried out for the arrival times of the IC1900 trains at Den Haag HS, based on a prediction horizon of 20 minutes, and for the arrival times of the local trains ST5100 at Den Haag HS, based on a prediction horizon of 25 minutes. The results are shown in Figure 11, with on the left 401 predicted arrival delays of IC1900 trains, and on the right 446 predicted arrival delays of ST5100 trains.

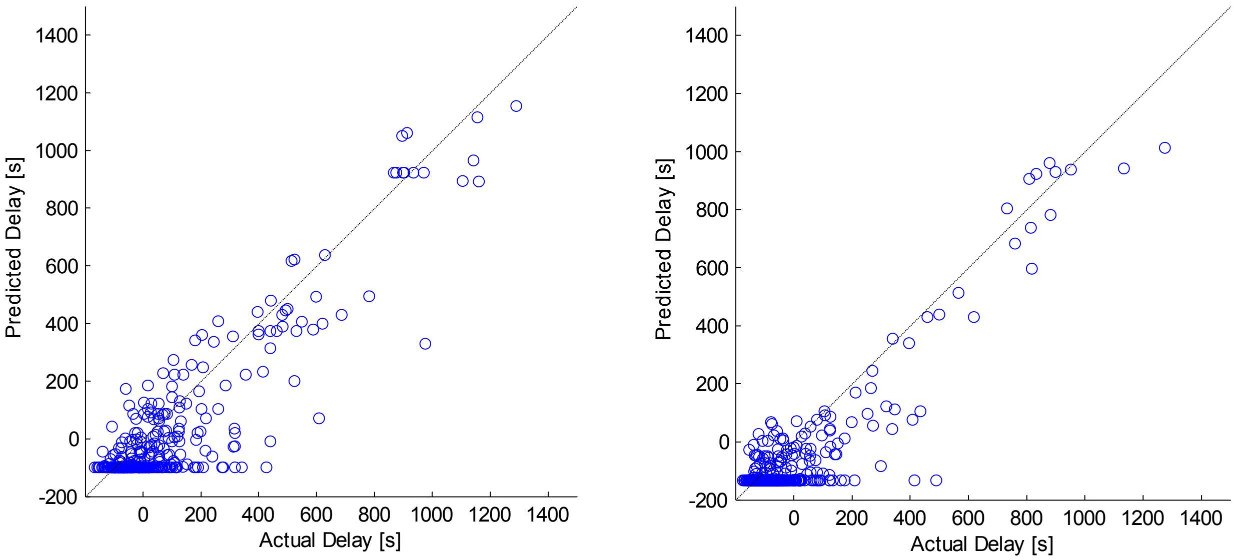


Figure 11 Predicted arrival delays at Den Haag HS (left: IC1900, with prediction horizon 20 minutes, right: ST5100 with prediction horizon 25 minutes)

As an example, Figure 12 contains the prediction for the IC 1958 at May 21. The solid lines denote the actual prediction, whereas the dashed lines denote the actual operations as recorded in the track occupation data. This train is scheduled to run between trains 2256 and 5058, but at Rotterdam Centraal the departure of IC 1958 is postponed, causing the IC 1958 to run behind train 5058. As can be seen, the headway time at Delft Aansluiting guarantees that route conflicts between the two trains are taken into account. The minimum running times of the IC 5058 and IC 1958 are slightly too short in this case.

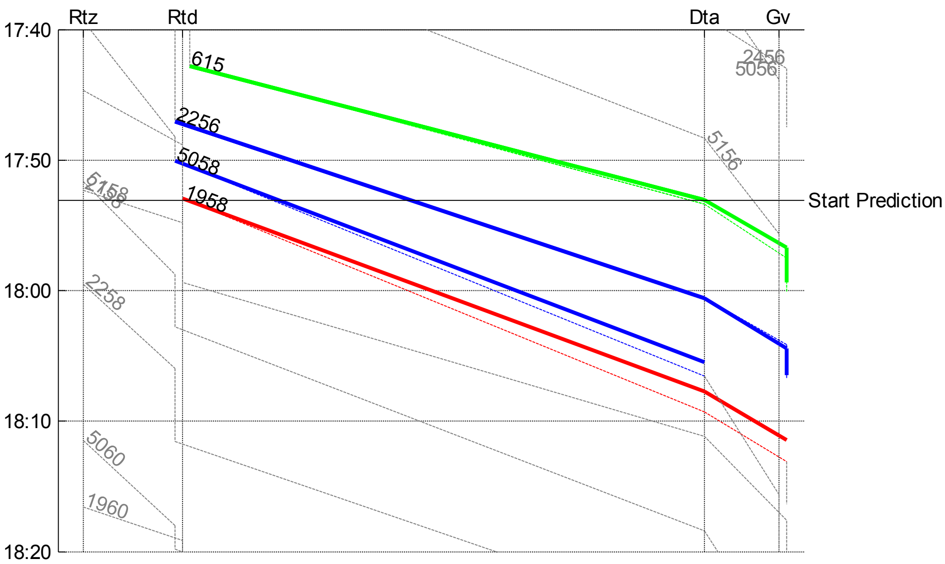


Figure 12 Prediction of the arrival of IC 1958 on May 21, 2007

2007.

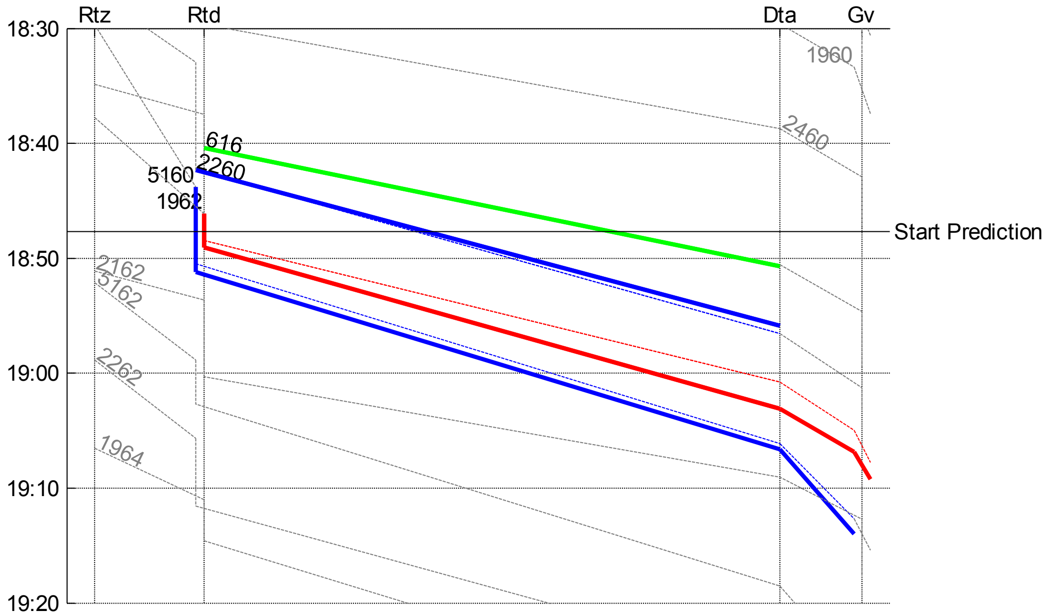


Figure 13 Prediction of the arrival of ST 5160 on May 27, 2007

Figure 13 shows the prediction for the local train 5160. This train should run before train 616, but arrived with a delay at Rotterdam Centraal, where its departure was postponed until trains 2260 and 1962 had departed. As can be seen, the model takes into account the minimal headways at Rotterdam Centraal and Delft Aansluiting between trains 1962 and 5160 correctly, leading to a good prediction.

As explained in the statistical analysis of the previous chapter, some trains experience more peak-hour dependency than others. In order to test whether the model can be improved by including peak-hours, the arc weights for the running times of the local trains between Rotterdam and Den Haag are changed to include peak hours. The running times are now calculated separately for each train number. As an estimation for the minimum running time, the 20th percentile of all recorded unhindered running times of that train number, and its two neighbouring train numbers, is calculated (e.g. the minimum running time for train 5168 is estimated by the 20th percentile of the recorded unhindered running times of trains 5166, 5168 and 5170). Hence, a 'moving percentile' is obtained, which can be found in Figure 14, where the peak hours are clearly visible.

The same has been done for the ST 5000 trains, for which the moving 20th percentile roughly has the same shape (until 20:15 after which the ST 5000 no longer runs). The running times of the other trains were unchanged for the following reasons:

- For the S 2200 and IC 1900, almost no peak hour dependency on the section Rotterdam Centraal – Den haag HS was found.
- The IC 2100, IC 2400 and international trains have no scheduled stops between Rotterdam Centraal and Den Haag HS. Since the peak hours influence train operations mainly via dwell times almost no peak hour dependency has been found for these trains,.

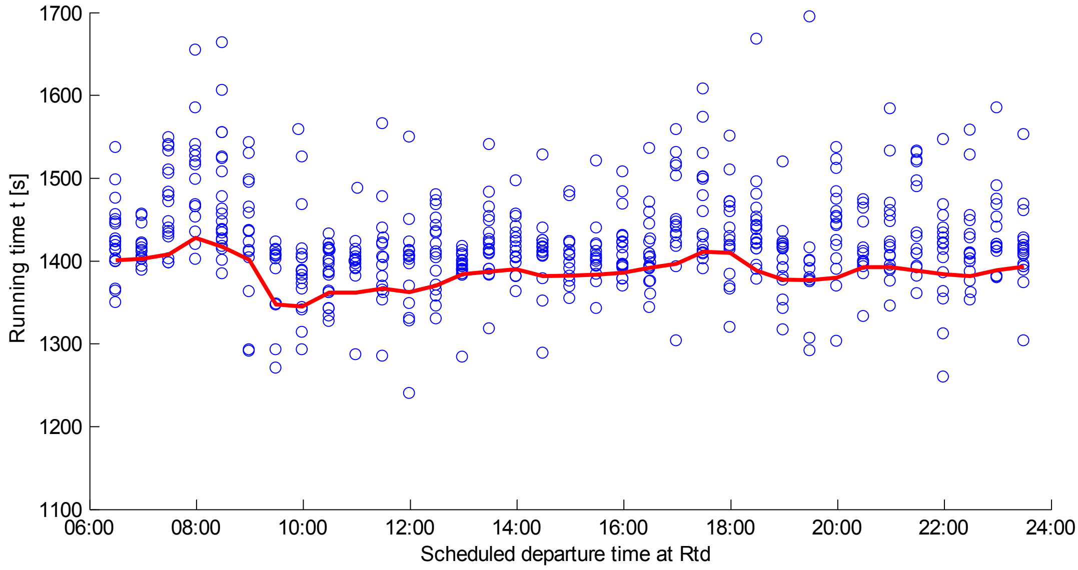


Figure 14 Running times of ST 5100 trains at Rtd - Gv. The red line is the moving 20th percentile.

Now, the same predictions as before have been tested for the ST 5000 trains. The results are shown in Figure 15. Comparison with Figure 11 (right) shows only limited difference. In order the compare the results, a mean squared error MSE has been calculated by

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{Actual\,delay}(i) - \mathrm{Predicted\,delay}(i))^2$$

For the prediction of the ST 5000 trains with fixed arc weights *MSE* = 8969, whereas for the analysis with peak-hour dependent running times for the local trains *MSE* = 8669. Hence, a small improvement has been found, although the individual predictions only show very small differences when compared to the model with the fixed arc weight. No case with a relatively large improvement could be found.

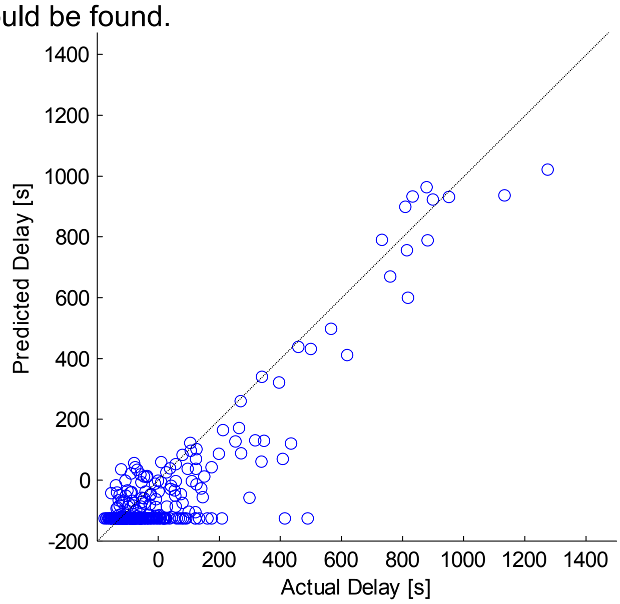


Figure 15 Prediction of the ST 5100 arrival times at Den Haag HS, with a time horizon of 25 minutes, with peak hour dependency

CONCLUSIONS

This paper presented a flexible delay propagation model in which train path conflicts and dispatching decisions are taken into account, and parameters are estimated by offline statistical analysis of historical train detection data. In dense railway networks, like the Dutch network, delay propagation has a great impact on train operations. The model can be used for predicting running times and arrival times of punctual and delayed trains, which can be used in decision support systems for dispatchers, to evaluate the effectiveness of certain dispatching decisions, or for dynamic passenger information systems.

A statistical analysis showed the dependencies of running times and dwell times on current delays and period of the day. A simple but well performing estimate for minimum running times is obtained by taking a small percentile of the observed running times, both for non-stop train runs and train runs including intermediate short stops. A robust estimate for minimum dwell times at stations with large scheduled dwell times is obtained by the median dwell time of trains with a sufficiently large arrival delay. Waiting times due to train path conflicts are added to the running and dwell times by the precedence constraints in the delay propagation model.

The structure of directed graphs to model precedence relations between events is not limited to timed event graphs, as used in this paper and for example in Goverde (2005, 2010) and Van der Meer (2008), but is generally applicable. The results of this research may thus form a starting point for further investigation on how to make these graph-based models more accurate by thorough statistical analysis of track occupation data, while achieving fast calculation times by keeping the models simple. For instance, the results of this research are equally applicable to the alternative graph models for computing optimal dispatching decisions (D'Ariano, 2008).

This paper considered simple estimators for the minimum running times. However, the statistical analysis revealed a linear dependency between delays and running times for small delays up to some threshold, and in particular for trains with large running time supplement. More advanced minimum running time estimations may therefore be used as a piecewise linear function consisting of the maximum of the regression line for small delays and a small percentile for large delays. Also multiple regression models can be investigated with a combination of many factors. The network model in this paper used only a subset of arrival, departure and passing events on the train routes. This can be refined to the measurement locations of the train describer systems (at signals) so that the most up-to-date current train positions and their dependencies are available to the online model. This will result in a more accurate but still manageable model as opposed to modelling all section occupations and releases and their relations. Current research focuses on the right balance between complexity and accuracy.

ACKNOWLEDGEMENT

This paper is a result of the project 'Reliable Transport Chains' within the Dutch TRANSUMO (TRANsition to SUstainable MObility) research programme.

BIBLIOGRAPHY

D'Ariano, A. (2008). *Improving real-time train dispatching: models, algorithms and applications*. PhD thesis, Delft University of Technology, Delft.

Conte, C. (2007). *Identifying Dependencies Among Delays*. PhD thesis, Georg-August-Universität zu Göttingen, Göttingen.

Daamen, W., Goverde, R.M.P., Hansen, I.A. (2009). Non-discriminatory automatic registration of knock-on train delays. *Networks and Spatial Economics*, 9(1), 47-61.

Flier, H., Gelashvili, R., Graffagnino, Th., Nunkesser, M. (2009). Mining Railway Delay Dependencies in Large-Scale Real-World Delay Data. In: *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems* (R.K. Ahuja, R.H. Möhring, C.D. Zaroliagis, eds.), LNCS 5868, pp. 354-368. Springer, Berlin.

Goverde, R.M.P. (2005). *Punctuality of railway operations and timetable stability analysis*. PhD thesis, Delft University of Technology, Delft.

Goverde, R.M.P. (2010). A Delay Propagation Algorithm for Large-Scale Railway Traffic Networks. *Transportation Research Part C*, accepted.

Goverde, R.M.P., Hansen, I.A. (2000). TNV-Prepare: Analysis of Dutch Railway Operations Based on Train detection Data. In: *Computers in Railways VII* (Allan, J., Hill, R.J., Brebbia, C.A., Sciutto, G., Sone, S., eds.), pp. 779-788. WIT Press, Southampton.

Goverde, R.M.P., Hansen, I.A., Hooghiemstra, G., Lopuhaä, H.P. (2001a). Delay Distributions in Railway Stations. Proceedings of the 9th WCTR, July 22-27, 2001, Seoul.

Goverde, R.M.P., Hooghiemstra, G., Lopuhaä, H.P. (2001b). *Statistical Analysis of Train Traffic: The Eindhoven Case*. TRAIL Studies in Transportation Science, S2001/1, DUP Science, Delft.

Hansen, I.A., Pachl, J. (eds.) (2008). *Railway Timetable & Traffic: Analysis, Modelling, Simulation*. Eurailpress, Hamburg.

Rousseeuw, P.J., Driessen, K. van (2006). Computing LTS Regression for Large Data Sets. *Data Mining and Knowledge Discovery*, 12, 29-45.

Van der Meer, D.J. (2008). *Modelling railway dispatching actions in switching max-plus linear systems*. MSc thesis, Department of Transport and Planning, Delft University of Technology, Delft.

Van der Meer, D.J., Goverde, R.M.P., Hansen, I.A. (2009). *Prediction of train running times using historical track occupation data.* Report, Department of Transport and Planning, Delft University of Technology, Delft.

Yuan, J., Hansen I.A. (2007). Optimizing capacity utilization of stations by estimating knock-on train delays. *Transportation Research Part B*, 41(2), 202-217.

Yuan, J. (2006). Stochastic modelling of train delays and delay propagation in stations. PhD thesis, Delft University of Technology, Delft.