

ONLINE DELAY MANAGEMENT IN RAILWAYS

- SIMULATION OF A TRAIN TIMETABLE
WITH DECISION RULES -

28-08-2013

N. VAN MEERTEN

333485

Econometrics & Operational Research
Erasmus University Rotterdam
Bachelor thesis

Supervision:
Dr. D. Huisman

ABSTRACT

Every day there are many people that travel by train. There are also many people that have a delay caused by this travelling. This delay can be a small delay of a few minutes or a big delay. A big delay can be caused when passengers miss their connecting train because of a small delay of the feeder train. To satisfy the passengers in the feeder train, the connecting train can wait for these passengers. But then the passengers in the connecting train are delayed and can miss their connecting train at another station. So the decision to be made for the connecting train is waiting for the delayed feeder train or departing on the scheduled time. This problem is known as delay management. In this thesis we will look at online delay management, this is useful in real life. Because in real life the delays are not known beforehand. For the online delay management we will use a simulation of a real life timetable and we will generate delays. With decision rules, we will use in the simulation, we can decide if a connecting train waits for a delayed feeder train or depart on the scheduled time. Finally, we evaluate the number of passengers that have a delay and the total delay of all passengers for the different decision rules.

TABLE OF CONTENTS

1	Introduction	1
2	Problem definition	2
3	Simulation	3
3.1	Assumptions	3
3.2	Activities	3
3.3	Steps of the simulation.....	4
3.4	Events	6
3.4.1	Train departure event.....	7
3.4.2	Train arrival event	8
3.5	Example of a model.....	8
3.5.1	Schedule with delays	9
4	Decision rules	11
4.1	Waiting Time Rule	11
4.2	Ratio of Transferring Passengers	11
5	Evaluation.....	13
5.1	Algorithm.....	13
5.1.1	Graph definition	13
5.1.2	Example of a graph.....	14
5.2	Dijkstra's algorithm.....	15
6	Computational results.....	16
6.1	The data.....	16
6.2	Implementation	16
6.3	Results Waiting Time Rule.....	17
6.3.1	Results for the 10 and 100 runs	17
6.3.2	Comparing the output for 10 and 100 runs	17
6.4	Results Ratio of Transferring Passengers	18
6.4.1	Results for the 10 and 100 runs	19
6.4.2	Comparing the output for 10 and 100 runs	19
6.5	Comparing the WTR and RTP decision rules.....	20
7	Conclusion.....	22
8	References.....	23

1 INTRODUCTION

About 1.2 million people travel by train in the Netherlands in a normal weekday. Especially in the Western part of the Netherlands, the Randstad, a lot of people make use of the railway system. There are several reasons for the fact that a lot of people travel by train, the main reasons are: people do not have another opportunity, people do not want to get stuck in the traffic jam with the car and the biggest reason (46%, according to the NS) is that people want to travel relax.

A problem about the railway system is that trains can have a delay because of an accident, disturbances, the weather or in the peak hours when there are a lot of people that want to travel by train to go to their job, school or going home. Most of the time the delays are small, the train arrives a few minutes later. This few minutes can be important for people that want to transfer to a connecting train on a station. They can miss their transfer to the connecting train and have to wait for the next train. To keep these people satisfied, the connecting train should wait for the delayed feeder train so the people of the feeder train can transfer. But then there are people who are already in the connecting train and will arrive later at another station where these people will miss their connection and so they will have a delay. This is an important issue, because 25% of the passengers transfer to another train. We want to minimize the total delay of all passengers and the number of passengers that have a delay. We focus on the small delays, the delays from 0 till 15 minutes. Because for the large delays most of the time people can wait for the next train, most trains run 2 or 4 times in one hour.

We will use data from Netherlands Railways (Nederlandse Spoorwegen, NS), this data is from the Randstad. In this thesis we will minimize the total delay of all passengers and the number of passengers with a delay with online delay management decisions. The decisions, we called decision rules, we will use are a decision rule that check if the delay is smaller than a given threshold and we will use a decision rule that takes into account the number of passengers.

This thesis is divided in a few chapters. In chapter 2 the problem will be defined. Then in chapter 3 we will discuss the simulation which will generate delays which will go throughout the whole network, so we get a real life situation. Within this simulation we will use wait and depart decisions to decide if a train better can wait for a delayed feeder train or better can run on the scheduled time. These decision rules will be discussed in chapter 4. The solutions of these decisions will be evaluated in chapter 5 with the shortest path algorithm. Then the results will be discussed in chapter 6. The overall conclusion will be given in chapter 7.

2 PROBLEM DEFINITION

This chapter describes the problem of online delay management. Delay management investigates if a connecting train can better wait for a delayed feeder train or it is better to depart on the scheduled time. When it is decided that a connecting train should wait for a delayed feeder train, this connecting train will probably arrive with a delay at the next station. On this station there are other passengers that want to transfer. This can be propagated through the entire railway network. Also the passengers that are already in the connecting train, will be delayed. But when it is decided that a connecting train departs on the scheduled time, the transferring passengers from the delayed feeder train will have a larger travel time because they have to wait for the next train to their destination. These decisions of waiting or departing are the aim of delay management. Also the overall delay for the passengers have to be minimized. Because the decisions have to be made for every train, these wait-depart decisions propagate throughout the entire railway network. So we have to consider the entire railway network at once, when we solving the delay management problem.

There are two types of delay management, namely online delay management and offline delay management. In the offline delay management problem, we assume that all delays in the system are known for a given period of time. This means that all primary delays in the system are known before the optimization process starts and we can predict the following delays with the primary delays. So offline delay management can be used when delays can be predicted and offline delay management can propose how to react to these following delays. In the online delay management problem there can be arise new delays. This is more like reality, because normally we know the delays when they occur and not beforehand. In this thesis we will use online delay management.

The research question for this thesis is as follows:

'How can you apply models for online delay management?'

In this thesis we will investigate the research question with use of different decision rules. With this rules we can decide if a connecting train will wait on a delayed feeder train or departs on the scheduled time.

3 SIMULATION

In this chapter we describe the structure of the simulation of delay management. A simulation is an imitation of reality. First we have to simulate the whole train system and generate delays. Then with the different decision rules we can decide if a connecting train departs or waits for a delayed feeder train. Finally, we can evaluate the consequences of every decision rule. So the model consists of three parts, namely the simulation of the train system, the decision rules and the evaluation.

Section 3.1 describes the assumptions that we need in the simulation. The activities will be described in section 3.2. Then the steps of the simulation will be described in section 3.3. Section 3.4 describes the events. Finally, an example of a model and a schedule are shown in section 3.5.

3.1 ASSUMPTIONS

First we make a few assumptions, namely:

- Passengers always choose for the fastest route to get to their final destination. It is not necessary that a passenger choose for the train that will depart first, because this train do not have to arrive first on the final destination. It can be that a passenger can better choose for a train that departs later and arrives earlier on the final destination then when the passenger choose the first train that departs earlier and arrives later on the final destination. This also holds when a delay occurs. So this means that passengers know which trains will have delays and can plan their route to their final destination.
- Delays can only occur while driving from station to station. So delay cannot occur while stopping and waiting on a station.
- According to the NS about 5% of the trains have a delay. So we will take about 5% of the events as the number of delays that occur in the train system. In reality this is of course not always the case, but we take this as approximate for a general day.
- A delay varies from 1 to 15 minutes per trip. These numbers are chosen by Dollevoet (2013). This can be explained because we want to use small delays.
- Trains will not wait unnecessary. So if a train has a delay that is greater than the time that the train will stop and wait on a station (dwell time) plus the slack time while driving, then the train will have a smaller stopping time at a station if it is possible and departs as soon as possible.
- A connecting train that waits for passengers that want to transfer from a delayed feeder train to the connecting train cannot cause problems within a station and for the trains that drive on a track. In reality there can be problems, because often there are more trains that arrives in a station at the same platform. When the connecting train waits it can block another train that arrives at the station. The capacities of stations can be studied in Dollevoet (2013) and Dollevoet et al. (2012c).

3.2 ACTIVITIES

A train will have two activities, namely:

- Driving (A_{drive}): The set of activities where a train drives from station to station.
- Stopping (A_{stop}): The set of activities where a train stops and waits at a station.

This means that $A = A_{drive} \cup A_{stop}$.

For both activities there are some issues that apply. The following issues apply for driving:

- There is a next station known and the time to get there is also known.
- The slack time while driving from station to station is known, this can influence the actual arrival time.

The following issues apply for stopping:

- The slack time while waiting at a station (dwell time) is known, this can influence the actual departure time.
- For every trip it is known that there is a certain waiting time at a station, from which the train then left. This is known as the dwell time.
- For every trip, except the last one, there can be trains arriving at the same station with people who might transfer to the connecting train.
- For every trip, except the first one, there could be passengers from the connecting train who wants to transfer to another train (and so the connecting train have to be on time).

3.3 STEPS OF THE SIMULATION

Before we run the simulation, we will order the departure times on time of occurrence beginning with the smallest departure time in the timetable.

Also we will check per activity if an activity is a drive and if this activity will have a delay. So we will simulate delays for about 5% of the events, see section 3.1. We will do this by drawing a random value from the uniform distribution on the open interval (0,1) for every driving activity. When this value is larger than 0.95, this means that the drawn number is within the 5% of the trains that have a delay, we will allocate a delay to that activity. When a delay occurs while driving from station to station, the train it's delay is a uniformly distributed number between 1 and 15 minutes (see section 3.1).

Then we will make a column vector for every simulation run with the delay numbers at the same place as the activities that have a delay. The activities that do not have a delay, will have a zero instead of a delay number. So the delay numbers are fixed per trip, but are different in every simulation run. Thus, we have a matrix in which the columns representing the simulation runs and the rows are the activities. We have fixed the random delay numbers per trip, so we can compare the results of the several thresholds of a decision rule and we can compare the decision rules with each other.

Before we will start with the simulation we will calculate the shortest path of the passengers, this will be described in section 5. After the simulation we will again calculate the shortest path of the passengers. With the shortest path of the passengers before and after the simulation we can calculate the total delay of all passengers and the number of passengers that have a delay.

After the ordering of the activities and the added delays, we will describe the next steps in the simulation. The simulation will go throughout the ordered timetable activity by activity.

Step 1

Check if the current activity is a drive. If it is a drive, then go to step 2.

Step 2

Check if the delay at the current activity is not zero, so that there is a delay. Then go to step 3, otherwise go to step 5.

Step 3

Calculate the actual arrival of the current activity with formula (1) which will be described in section 3.4. Update the scheduled arrival time with the actual arrival time in the timetable.

Step 4

We will check for every activity (drive and dwell) after the current activity if the activity has the same train number as the current activity and if there is a positive delay, this means that the generated delay minus the slack time up to now is larger than zero. If both conditions are true, then update the arrival time of this next activity and the slack time with the slack time in this next activity. With this total slack time we update the departure time of this next activity for this train. This actual arrival and departure times will be the new scheduled arrival and departure times, because we have updated the times in the timetable.

When there is no longer a positive delay, then the arrival and departure times of the next activities do not have to be updated anymore.

Step 5

Now we can look at connections of feeder trains to the connecting train at the station from which the current activity will depart. We will look which trains have planned to make use of the connection with the connecting train.

Step 5a

First we will look if there are previous trips with the same departure and arrival stations as the current trip. If a trip or more trips exist, then we go to step 5b. Otherwise there is no previous trip with the same departure and arrival stations and we go to step 5c.

Step 5b

There is a previous trip or there are more previous trips with the same departure and arrival stations as the current trip. We select the last previous trip. Then we will look which trains have planned to arrive at the station before the connecting train will depart and after the departure | time of the previous train with the same next station. So this trains have planned to make use of the connection with the connecting train. Go to step 6.

Step 5c

There are no previous trips with the same departure and arrival stations as the current trip. So we will look which trains have planned to arrive at the station before the connecting train will depart . So this trains have planned to make use of the connection with the connecting train. Go to step 6.

Step 6

From each of these incoming trains which have planned to make use of the connection with the connecting train we will find the actual arrival times which are in the updated timetable. Then we will check if the actual arrival time of the incoming train is larger than the departure time of the connecting train. If this is true, then we can calculate the number of minutes that the incoming train is too late to make use of the connection with the connecting train.

For each incoming train that is too late to make use of the connection with the connecting train, we make use of a decision rule. With a decision rule we can decide if a connecting train better can wait for an incoming train or better can depart.

Step 7

In a decision rule we check if there are incoming trains that are too late that fall within the threshold. From these incoming trains we take the maximum of the number of minutes that these incoming trains are too late to make use of the connection with the connecting train. This maximum is the number of minutes that the connecting train will wait.

Then we will repeat step 4. After the updating of the arrival and departure times of the next activities of this connecting train, go to step 8.

Step 8

If there is a next activity in the timetable, go to step 1 and repeat the steps. If there is not a next activity in the timetable, we have run through the whole timetable. So the simulation is done.

3.4 EVENTS

The simulation consists of the events: train arrival and train departure. This arrival and departure events will take place at each station for every train, so for every activity. Except at the first station and the last station of a trip. At the first station there is only a departure event and at the last station of a trip there is only an arrival event.

How we get the arrivals and departures, is shown below in formulas. In these formulas we assume that a connecting train t will travel from station $k-1$ to station k to station $k+1$, etc. So the train travels in numerical order. Also we assume that a connecting train t is now in the current activity i which is a drive. It is a drive, because we only look at activities which are drives (see Section 3.3 Step 1). And this connecting train t will have a delay, that is generated in this activity. An example with the formulas is shown in section 3.5. First we define the variables that we will use in the formulas:

$DS_{i,t}$	= Scheduled departure time of activity i train t , this is the updated departure time
$DA_{i,t}$	= Actual departure time of activity i for train t
$D_{i,t}$	= Generated delay of train t of activity i
$SS_{i,t}$	= Scheduled slack time while driving or stopping at a station of activity i for train t (given from the data)
$AS_{i,t}$	= Scheduled arrival time of activity i for train t , this is the updated arrival time
$AA_{i,t}$	= Actual arrival time of activity i for train t
$MinutesLate_{i,s}$	= Number of minutes that train s will be too late for the connection with train t at activity i

In the last variable we will define a train s as an incoming train.

With the variables described above we can formulate the formulas for the arrival and departure events. These variables, named $AA_{i,t}$ and $DA_{i,t}$, will be updated throughout the simulation. For the actual arrival and departure times we make use of a maximum of the scheduled arrival and departure times, respectively, and this scheduled times plus a delay and minus the slack times of the activities from the activity with a delay till now. Thus, train t arrives and departs at the latest time that we get from the maximum. The scheduled times that are used are updated every activity with the actual times obtained from the previous activities.

We will use the pseudo code in Figure 1 for updating the departure and arrival times of the next activities.

```

Slackt = SSi,t;
AAi,t = max (ASi,t; ASi,t + Di,t - Slackt);           (1)
ASi,t = AAi,t; // Update this actual arrival time in the timetable
for every activity j = i+1:T: // T is the length of the timetable, this is 3741
    if the activity is of train t and Dt - Slackt > 0:
        DAj,t = max (DSj,t; DSj,t + Di,t - Slackt);           (2)
        DSj,t = DAj,t; // Update this actual departure time in the timetable
        NextSlackt = SSj,t;
        Slackt = Slackt + NextSlackt;
        AAi+1,t = max (ASi+1,t; ASi+1,t + Di,t - Slackt);           (3)
        ASi+1,t = AAi+1,t; // Update this actual departure time in the timetable
    end if
end for

```

Figure 1: Pseudo code for updating the departure and arrival times of next activities of train t .

It is possible that a train s connects with train t in the scheduled timetable and that train s have a delay, so that train s connects no longer with train t . Except if train t waits for train s . But then train t is delayed. This can be propagated through the whole railway network. With a decision rule there can be decided if train t better can wait for train s or better can depart. If train t better can wait, we get the following formulas for train t :

$$MinutesLate_{i,s} = AA_{i,s} - DA_{i,t} \quad \forall s \quad (4)$$

Formula (4) will be done for every incoming train s . Then a decision rule will be used to determine which incoming trains fall within the threshold. Select from these incoming trains the $MinutesLate_{i,s}$ and take the maximum of this, we called $MaxMinutesLate_i$, this is the number of minutes of the latest incoming train that will fall within the threshold. We will use the pseudo code in Figure 2 for updating the departure and arrival times of the next activities.

```

Slackt = 0;
for every activity j = i:T // T is the length of the timetable, this is 3741
    if the activity is of train t and MaxMinutesLatei - Slackt > 0:
        DAj,t = max (DSj,t; DSj,t + MaxMinutesLatei - Slackt);           (5)
        DSj,t = DAj,t; // Update this actual departure time in the timetable
        NextSlackt = SSj,t;
        Slackt = Slackt + NextSlackt;
        AAi+1,t = max (ASi+1,t; ASi+1,t + MaxMinutesLatei - Slackt);           (6)
        ASi+1,t = AAi+1,t; // Update this actual departure time in the timetable
    end if
end for

```

Figure 2: Pseudo code for updating the departure and arrival times of next activities of train t when this train waits for delayed feeder trains.

3.4.1 TRAIN DEPARTURE EVENT

When a train t is not delayed ($D_{i,t} = 0$) and the incoming trains at activity i arrived at their scheduled time ($AA_{i,t} = AS_{i,t}$) or there are no incoming trains, the actual departure time of train t will be the scheduled departure time ($DA_{i,t} = DS_{i,t}$) of activity i .

When a train t is delayed ($D_{i,t}$), the scheduled arrival time of activity i is no longer correct if $D_{i,t} > SS_{i,t}$. Because of this it can be that the scheduled departure time of activity $i + 1$ is also no longer correct. When an activity is a drive and there are incoming trains that will be too late, we will use the different decision rules. With these rules we can decide if the train departs at the scheduled time or will depart later because of the delay of a feeder train s or more delayed feeder trains.

To reduce the delay we can make use of the slack time of activity i ($SS_{i,t}$) and if it is necessary of the slack times of the next activities of train t . In the pseudo codes in the beginning of section 3.4 is shown how we make use of the slack times. It can be seen that the slack times are used in the formulas for the actual arrival and departure times.

3.4.2 TRAIN ARRIVAL EVENT

When a train t arrives at a station k , the train can stop to pick up passengers. When the train is not delayed or the delay is smaller than the slack time while driving ($D_{i,t} < SS_{i,t}$), the train will arrive on the scheduled time. Then the actual arrival time is the same as the scheduled arrival time ($AA_{i,t} = AS_{i,t}$). But when the train will be delayed, the actual arrival time is not the same as the scheduled arrival time anymore if the delay is larger than the slack time while driving ($D_{i,t} > SS_{i,t}$). The delay will be included in next activities of train t . In the pseudo codes in the beginning of section 3.4 is shown how we make use of the slack times and how we updated the arrival times.

3.5 EXAMPLE OF A MODEL

For every train route, present in the data, we can make a model like the one in Figure 3. This model shows the train route of train number 2174 from beginning to the end with in between the stops at stations.

Each trip can have a certain delay, this is given by the “Delay Generator”. This delay can influence the arrival time at the next station, this depends on the slack time while driving. If this slack time is larger than the delay, everything is going well. At a station there is a certain time needed that people can get off and on the train. If a train is arriving with a delay then the slack time at the station (dwell time) can be used, so that the train can depart on the scheduled departure time.

Also we can see from which stations trains arrive at the stations on train route with train number 2174 and to which stations trains depart from stations on train route with train number 2174. So that the incoming trains must connect to the connecting train and the connecting train must connect to the departing trains. The other trains can be delayed as well, so that the actual arrival and departure times of this trains will no longer be the same as the scheduled times. With decision rules we can decide if the connecting train have to wait or depart.

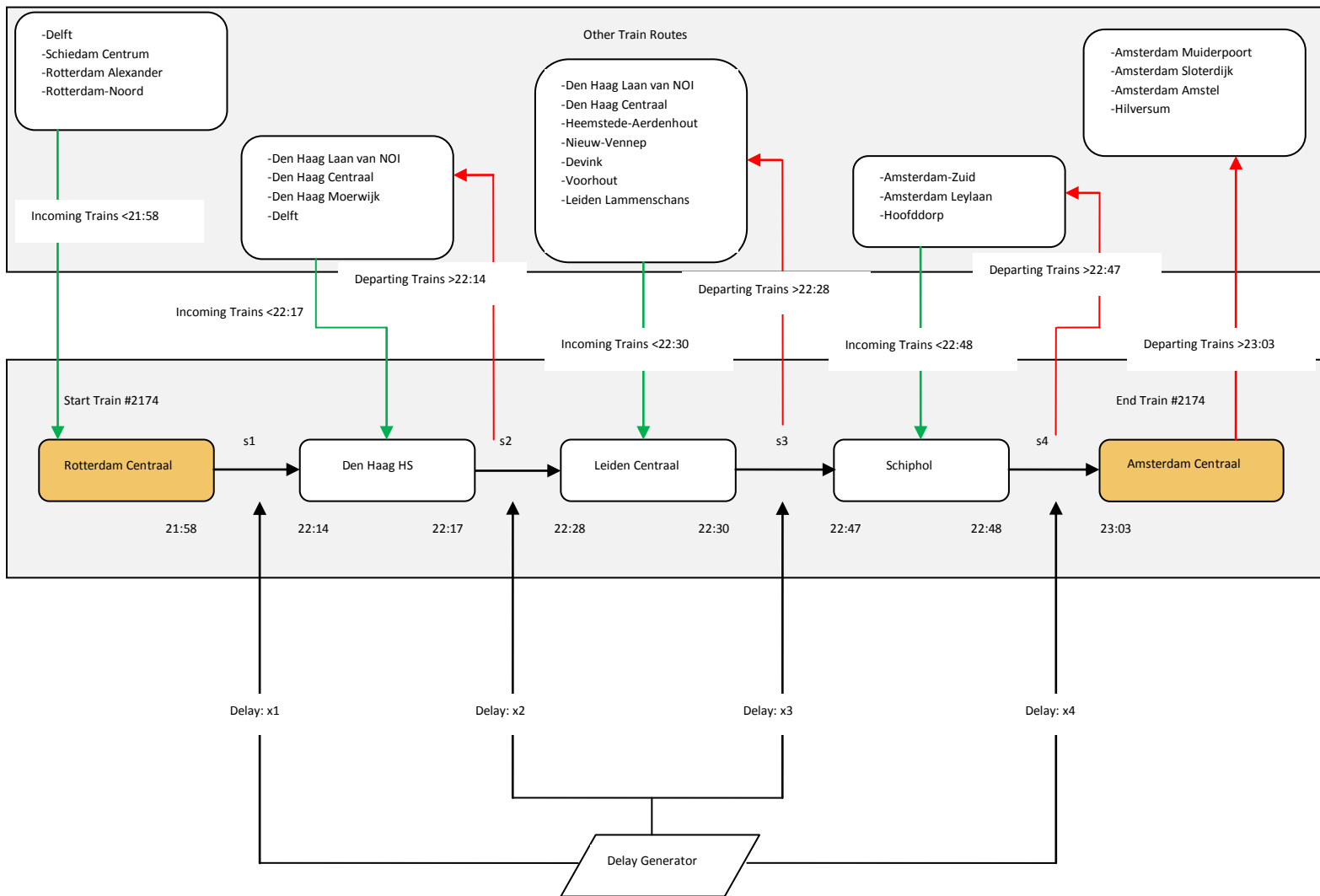


Figure 3: Train route 2174.

3.5.1 SCHEDULE WITH DELAYS

Table 1 shows a schedule (with the formulas (1), (2) and (3) of section 3.4) of the simulation of train route 2174. This schedule consist of $j = i, i + 1, \dots, i + 6$ activities, this activities are drives and dwells. We assume that in the trip from Rotterdam Centraal to Den Haag HS, activity i , that there is a delay of 3 minutes. This delay ensures that the sequel model can look different from the scheduled model, because it can be that the actual arrival times and departure times are different from the scheduled times by these delays. Also we assume that there are no incoming trains.

This schedule can be done for every train route.

	Scheduled departure ($DS_{j,2174}$)	Actual departure ($DA_{j,2174}$)	Delay ($D_{i,2174}$)	Slack time ($SS_{j,2174}$)	$SS_{j,2174}$ needed? ($D_{i,2174} > \sum_{j=t}^{i+6} SS_{j,2174}$?)	Scheduled arrival (AS_{ij2174})	Actual arrival ($AA_{j,2174}$)
R'damC – DH HS (i)	21:58	21:58	00:03	00:01	00:03>00:01	22:14	Formula (1) =max(22:14; 22:14+00:03-00:01) =22:16
DH HS – DH HS (i+1)	22:14	Formula (2) =max(22:14; 22:14+00:03-00:01) =22:16	00:03	00:01	00:03>00:02	22:17	Formula (3) =max(22:17; 22:17+00:03-00:02) =22:18
DH HS – LeidenC (i+2)	22:17	Formula (2) =max(22:17; 22:17+00:03-00:02) =22:18	00:03	00:00	00:03>00:02	22:28	Formula (3) =max(22:28; 22:28+00:03-00:02) =22:29
LeidenC – LeidenC (i+3)	22:28	Formula (2) =max(22:28; 22:28+00:03-00:02) =22:29	00:03	00:00	00:03>00:02	22:30	Formula (3) =max(22:30; 22:30+00:03-00:02) =22:31
LeidenC – Schiphol (i+4)	22:30	Formula (2) =max(22:30; 22:30+00:03-00:02) =22:31	00:03	00:01	00:03=00:03	22:47	Formula (3) =max(22:47; 22:47+00:03-00:03) =22:47
Schiphol – Schiphol (i+5)	22:47	Formula (2) =max(22:47; 22:47+00:03-00:03) =22:47	00:03	00:01	00:03<00:04	22:48	Formula (3) =max(22:48; 22:48+00:03-00:04) =22:48
Schiphol – A'damC (i+6)	22:48	Formula (2) =max(22:48; 22:48+00:03-00:04) =22:48	00:03	00:00	00:03<00:04	23:03	Formula (3) =max(23:03; 23:03+00:03-00:04) =23:03

Table 1: Schedule with a delay, changing arrival times and departure times of train route 2174.

4 DECISION RULES

Dollevoet (2013) describes different models for the delay management problem. In this thesis we will look at different decision rules and compare the solutions. We will look at the following decision rules of Dollevoet (2013): the Waiting Time Rule (WTR) and the Ratio of Transferring Passengers (RTP) in respectively section 4.1 and section 4.2.

4.1 WAITING TIME RULE

Netherlands Railways (Nederlandse Spoorwegen, NS) applies a so-called Waiting Time Rule (WTR). For each connection in the railway network, a threshold is determined. The threshold, or maximal waiting time at the station, is determined for each connection. We will assume a constant threshold for all connections, this is easier to simulate.

If the delay for connection a (D_a) is smaller than the threshold (d_{max}), the connecting train waits. Otherwise, the train departs on the scheduled time. Dollevoet (2013) describes how to find the best values for this threshold values. He has evaluated the heuristics for values ranging from 0 to 5 minutes. Note that $d_{max}=0$ corresponds to a no-wait policy. We will also range the threshold values from 0 to 5 minutes and find the optimal value.

If the train waits, the train will wait the maximum number of minutes from the incoming train(s) that is(are) too late and are lower than the threshold. Suppose the threshold is 4 minutes and there are two incoming trains that arrives 1 and 3 minutes later than the departure time of the connecting train, then the connecting train will wait 3 minutes instead of the 4 minutes of the threshold. Then the arrival and departure times of the next activities of this connecting train can be updated in the timetable. This is shown in section 3.4.

This decision rule has two important aspects. The first aspect has to do with peak and off-peak hours. In our data we only have off-peak hours, because we look at the data from 09.30 PM. So we do not have to take this aspect into account. The second aspect has to do with the number of passengers that transfer to the train. The WTR decision rule does not consider this. It seems to be obvious that when there are many passengers that transfer to the connection train, the connecting train should wait longer for these transferring passengers.

4.2 RATIO OF TRANSFERRING PASSENGERS

Dollevoet (2013) describes another decision rule, namely the Ratio of Transferring Passengers (RTP). The RTP heuristic computes the ratio of the number of passengers who plan to use connection a and the number of passengers who plan to use the connecting train, this ratio we called p_a . In formula it looks as follows:

$$p_a = \frac{\text{Number of passengers that planned to use connection } a}{\text{Number of passengers that planned to use driving activity } (e, f)}$$

Note that for every departure event e , there is exactly one driving activity $(e, f) \in A_{drive}$. If the ratio p_a exceeds a given threshold p_{min} , so that $p_a \geq p_{min}$, connection a is maintained. Otherwise it is dropped. Dollevoet (2013) experimented with values of p_{min} between 0 and 100 percent. Note that we obtain a

no-wait policy if $p_{min} > 100\%$. We will also range the threshold values from 0 to 100 percent and find the optimal value.

Suppose the connecting train with train number 2174 will travel from Rotterdam Centraal to Amsterdam Centraal and have stops in Den Haag HS, Leiden Centraal and Schiphol. And suppose there is an incoming train in Leiden from Voorhout. In this example the numerator is the number of passengers that want to transfer in Leiden from the train from Voorhout to the connecting train from Rotterdam Centraal to Amsterdam Centraal. And the denominator is the number of passengers that want to travel from Leiden Centraal to Schiphol.

If the train waits, the train will wait the maximum number of minutes from the incoming train(s) that is(are) too late and are fall within the threshold. Then the arrival and departure times of the next activities of this connecting train can be updated in the timetable. This is shown in section 3.4.

5 EVALUATION

In this chapter we will describe the method that evaluates the solutions of the different decision rules, which are discussed in chapter 4. The decision rules decide if a connecting train better can wait for a delayed feeder train or it is better to depart on the scheduled time. To evaluate the solutions of these decision rules we will calculate the routes of the passengers. With these routes we can calculate the following aspects per decision rule: total delay of the passengers and the number of passengers that have a delay. Then we can compare the results of these calculations of the different decision rules. It is obvious that the smaller the total delay, the better the decision rule works.

To calculate the total delay of the passengers we will use the fastest routes of the passengers before and after the simulation. With the fastest routes after the simulation the delay is taken into account. From these situations we will have the difference of the arrival times and so we can calculate the total delay of all passengers.

The method we will use to compute the fastest routes is a shortest path algorithm. We will use Dijkstra's algorithm, but any other algorithm can be used. Dijkstra's algorithm solves the single-source shortest paths problems.

In section 5.1 we will define a graph and show an example of a graph. In section 5.2 we will describe Dijkstra's algorithm.

5.1 ALGORITHM

The main idea of the algorithm is to find the path with the smallest time (i.e. the shortest path) between the given source node and every other node in the graph. A graph consists of nodes and arcs. To use the algorithm we will first define this graph.

5.1.1 GRAPH DEFINITION

First we will define the nodes. A node, we will see it as a station, can be of two types. The first type is an ODT-pair (origin, destination, time in minutes after 0:00 AM). This is a pair of stations which are the origin and destination of a passenger with a time when the passenger begins on his trip (starting time). These ODT-pairs contain the starting and ending stations of the shortest path. The second type is an event, this can be a departure or an arrival event.

Next we will define the arcs. An arc is a link that connects two stations. The weight of an arc will be defined as the time that is needed to go from station to another station. A starting station only has outgoing arcs to other stations. These stations have the same station label and a time label that is equal to or larger than the time label of the ODT-pair. An ending station only has incoming arcs from other stations. These stations have the same station label and a time label that is equal or smaller than the time label of the ODT-pair. An arc between stations is possible if the time label of the station from which the arc departs plus the weight of the arc is equal or smaller than the time label of the station at which the arc arrives. Further, an arc is possible if there is a connection between station labels of the stations or the station labels of the stations are the same at a given time of the timetable.

5.1.2 EXAMPLE OF A GRAPH

We will give a small example of a graph with a shortest path in Figure 4. The circles represent an ODT-pair. We see 3 ODT-pairs, namely: Gv-Vh (Den Haag HS – Voorhout) at 22:17, Gv-Asd (Den Haag HS – Amsterdam Centraal) at 22:17 and 22:47. The numbers at an arc are the minutes till the following event. So we can see the route of ODT-pair Gv-Asd at 22:17: the route start at Gv (Den Haag HS) at 22:17 with a departure, then train 1 arrives 11 minutes later at Ledn (Leiden Centraal) where the train will stop for 2 minutes, next the train arrives 17 minutes later at Shl (Schiphol) where the train will stop for 1 minute and then the train will arrive 15 minutes later (23:03) at Asd (Amsterdam Centraal). With the ODT-pair Gv-Vh at 22:17 the passenger(s) will transfer to train 2 at Ledn, so the passenger(s) can get on their destination Vh. From the starting station Gv-Asd at 22:17 there can start another possible route, namely the route with train 4. This is the route which is also possible with ODT-pair Gv-Asd at 22:47. So train 4 drives the same route as train 1, but then the arrival and departure times are half an hour later.

When we look at ODT-pair Gv-Asd at 22:17 there are three possible routes to go from Gv to Asd, namely: the direct route with train 1, the direct route with train 4 and the indirect route with train 1, train 2 and train 3. This indirect route can be useful when the passenger(s) know beforehand that train 1 has a delay between Ledn and Asd of more than 14 minutes. At Ledn the passenger(s) can transfer from train 1 to train 2, then the passenger(s) can transfer at Had (Heemstede-Aerdenhout) to train 3 which arrives at Asd at 23:17.

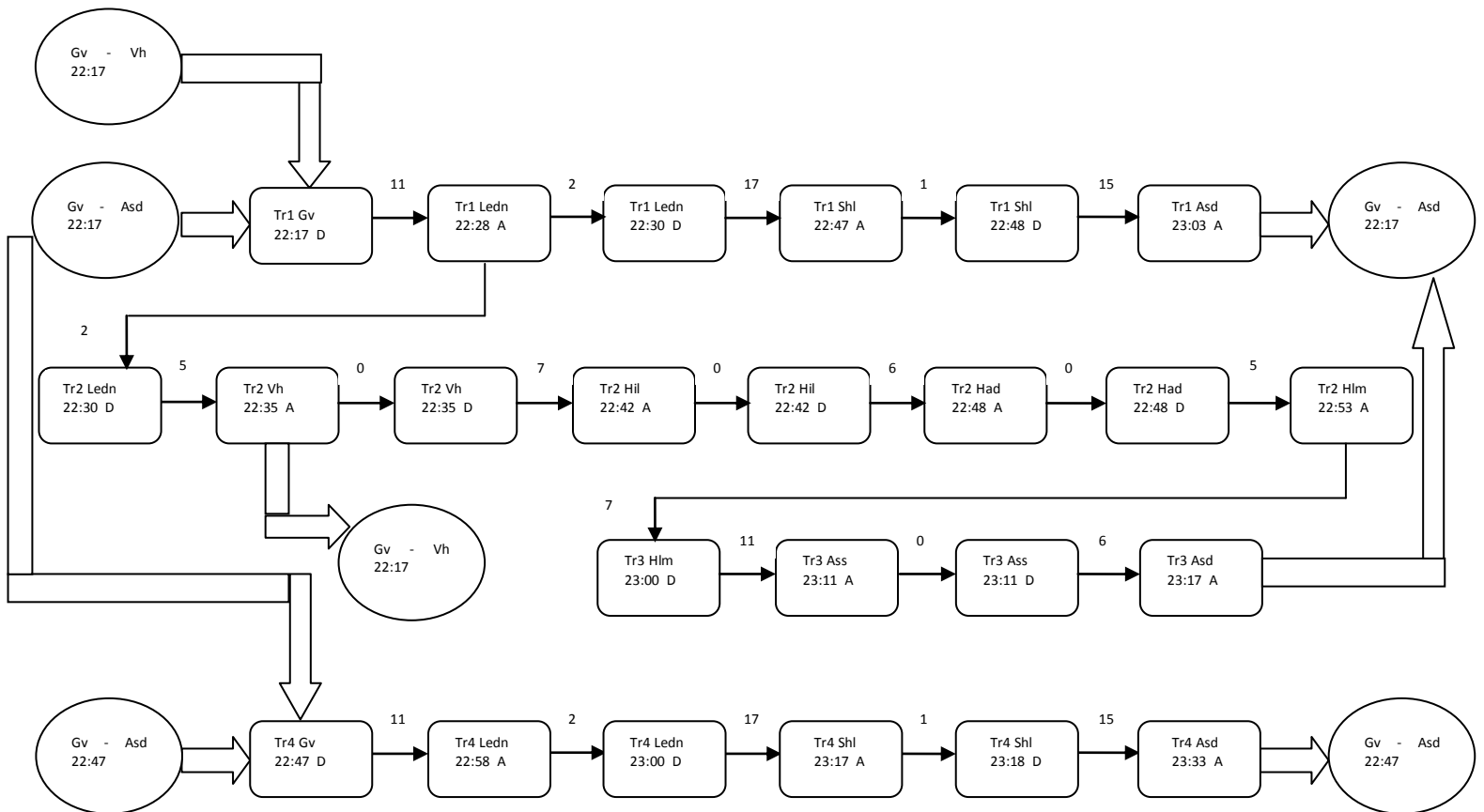


Figure 4: Example of a graph representation.

5.2 DIJKSTRA'S ALGORITHM

Now we have defined the graph, we can use Dijkstra's algorithm. The pseudo code of this algorithm adjusted to our situation can be found in Figure 5. To find the path with the smallest time (i.e. the shortest path) between the given start station and every other station in the graph we have to start with a column Q with numbers from 1 till 82, this are the number of stations, and two separate columns of infinity numbers named dist and previous. Where dist stands for the distance, in time, between two stations and previous stands for the previous stations. Then we can find for the station with the minimum arrival time in dist(Q) (the station at which you are) the stations that are available from this station and we delete this station in Q. Next, we can find for every available station the arrival time and put this arrival time for the available station in dist. Also put the number of the previous station (the station at which you are) in previous at the place of the available next station.

To calculate the shortest path of the passengers, we run Dijkstra's algorithm for every ODT-pair. Then we select the arrival time at the station of destination, this is the last arrival time. The travel time can easily be calculated by the last arrival time minus the start time of the passenger.

```

v = 82 ; // Number of stations
dist[length v] = infinity ; // Distance in time between two stations
previous[length v] = infinity // Previous stations
previous(StartPosition) = 0 ; // StartPosition is the origin station
dist(StartPosition) = StartTime ; // StartTime is the time at which the passenger starts with
// travelling
Q = 1:82 ; // The set of all stations in the graph
while Q is not empty :
    u = arc in Q with smallest distance in dist[] ;
    remove u from Q ;
    if dist[u] = infinity :
        break ;
    end if
    for each station v that is attainable from u : // Where v has not yet been removed from Q
        arrt = min(arrival times whereby:
            station from departure = station at which we are
            departure time >= dist[u]
            station from arrival = v ) ;
        if arrt < dist[v] :
            dist[v] = arrt ;
            previous[v] = u ;
        end if
    end for
end while
return dist ;

```

Figure 5: Pseudo code of Dijkstra's algorithm (Reference: Syllabus Wiskundige Methoden, 2007).

```

EndArrivalTimes = [] ;
TravelTime = [] ;
for each ODT-pair i :
    [dist, previous] = Dijkstra (Timetable, StartTime, StartPosition) ; // Run Dijkstra's algorithm
    EndArrivalTimes(i) = dist(ArrivalStation of the ODT-pair(i)) ; // Get the arrival time at the destination of an
                                                                    // ODT-pair
    TravelTime(i) = EndArrivalTimes(i) - (StartTime of the ODT-pair(i)) ;
end for

```

Figure 6: Pseudo code of the shortest path.

6 COMPUTATIONAL RESULTS

In this chapter we will discuss the results. First we discuss the data in section 6.1. In section 6.2 we will discuss the implementation of the simulation and other programs. Then we will discuss the results of the Waiting Time Rule and the Ratio of Transferring Passengers in section 6.3 and 6.4, respectively. Finally, in section 6.5 we compare the two decision rules with each other.

6.1 THE DATA

We will use data from Netherlands Railways (Nederlandse Spoorwegen, NS), this data is from the Randstad. In this part a lot of people make use of the railway system. The data contains timetables from 09:30 PM till 02:22 AM the next day, a list of ODT-pairs (origin, destination, time in minutes after 0:00 AM) and a list of stations. The timetables consists of train numbers, the station from which the train depart with associated departure time, the station at which the train drives to with associated arrival time, the slack time of the activities and the activities 'DRIVE' or 'DWELL'. A 'DRIVE' means that a train drives from a station to another station and a 'DWELL' means that a train stops at a station. The slack times are mainly between the 0 and 7 minutes. But there are exceptions, like the slack time of 21 minutes at Delft on 00:54 AM, the slack time of 10 minutes at Leiden Centraal on 01:02 AM, the slack time of 21 minutes at Gouda on 00:52 AM and the slack time of 9 minutes at Hilversum on 00:40 AM. The slack times consists of driving time and dwell time. This means that there is slack time when a train drives from station to station and there is slack time when a train stops at a station.

There are 3741 activities in this timetable. From this are 2066 a drive activity and 1675 activities are a dwell activity. The number of trains in this timetable is 391.

The list of ODT-pairs consists of an origin station, a destination station, a starting time of the passengers and the number of passengers of an ODT-pair. There are 3849 ODT-pairs with 9822 passengers.

6.2 IMPLEMENTATION

We have used Matlab 2009b to program the simulation. We chose Matlab because we have experience with programming in Matlab and Matlab supports vector and matrix calculations very well. Also we used the built-in math functions in Matlab and Matlab can handle large datasets, which we have.

The data we have contains all the input for the several simulations and programs. Before we run the simulation we calculate the shortest path per ODT-pair. In the simulation we make use of the programs of the different decision rules with the several thresholds. For every decision rule with a threshold we get a updated timetable. With this updated timetable we can again calculate the shortest path per ODT-pair. Then we can compare the shortest path per ODT-pair before and after the simulation, so we can calculate the total delay of all passengers and the number of passengers that have a delay. Finally, we can compare this for every decision rule with every threshold.

The comparison can be done because we have fixed the random delay numbers per trip in the timetable. So for every run of the simulation we use the same delay numbers at the same trip.

All the runs of the simulation and other programs in this thesis have been done on a Genuine Intel(R) CPU T2080 processor (1.73 GHz).

6.3 RESULTS WAITING TIME RULE

In this section we will discuss the results when we use the Waiting Time Rule. We can compare the results between the different thresholds (TH), so we can see what the effect is when we use a larger threshold. We have done 10 runs and 100 runs of the simulation. The duration of the simulation was about 4 minutes for the 10 runs and for the 100 runs it was about 13.5 to 14 minutes. The output of the simulation is the mean of the several runs. So for 10 runs the output is the mean of the 10 runs.

The delays are generated beforehand, so we use for a simulation run the same delays per activity for every threshold. This means that we have fixed the random delay numbers per trip, so we can compare the results of the several thresholds of the Waiting Time Rule. Every simulation run does have different delays. So the number of delays and the average delays are different between the simulation runs. But are the same for every threshold within a simulation run.

The average number of delays is for the 10 runs of the simulation 103.30 with an average delay of 7.96 minutes for the 10 runs of the simulation. For the 100 runs of the simulation we have an average number of delays of 103.43 and an average delay of 7.98 minutes.

6.3.1 RESULTS FOR THE 10 AND 100 RUNS

In Table 2 and Table 3 all the output is shown for 10 runs and 100 runs, respectively, of the simulation for different thresholds. We can see that the total delay of all passengers follows a slightly upward trend when the threshold will be larger. Only when we use 10 runs of the simulation and the threshold is 0 minutes, the total delay of all passengers will be somewhat larger than when the threshold is between 1 and 3 minutes. This can make sense, because when a connecting train will never wait for delayed feeder trains, a lot of passengers will miss their connection to the connecting train and have to wait for the next train. A small delay of a delayed feeder train can cause a larger delay for passengers that will miss their connection and have to wait for the next train. The number of passengers that have a delay is larger when the threshold is larger. This makes sense, because a connecting train can wait longer for a delayed feeder train when the threshold will be larger. So that this connecting train will have a delay because of the waiting and the passengers in the connecting train can miss their connections at other stations. So the longer a connecting train can wait because of the threshold, the more passengers will have a delay.

When we look at the number of trains that will wait for delayed feeder trains, we see that this number will be larger when the threshold will be larger. Also here it makes sense, because a connecting train can wait longer for a delayed feeder train when the threshold will be larger. Note that the number of trains that will wait for delayed feeder trains is zero when we use a threshold of zero, this corresponds to a no-wait policy. On the other hand the number of trains that will not wait for delayed feeder trains will be smaller when the threshold will be larger. This number is just the opposite of the number of trains that will wait for delayed feeder trains. Also the number of trains that will run on time will become smaller when the threshold will be larger.

6.3.2 COMPARING THE OUTPUT FOR 10 AND 100 RUNS

Now we can compare the output for 10 runs and 100 runs of the simulation. The number of delays and the average delay are a little bit different for the 10 runs and 100 runs of the simulation. The total delay of all passengers and the number of passengers that have a delay differs slightly between the 10 runs and 100 runs of the simulation. This difference can be caused by the difference in delays between the 10

runs and 100 runs of the simulation. Because the delays are random and thus different when we use 10 runs or 100 runs of the simulation.

When we compare the number of trains that will wait for delayed feeder trains, we can see that the average numbers of the number of trains that will wait for delayed feeder trains, the number of trains that will not wait for delayed feeder trains and the number of trains that will run on time do not differ that much from each other for every threshold. This means that the output of all runs of the simulation are about the same.

10 runs of the simulation	Total delay of all passengers	Number of passengers that have a delay	Number of delays	Average delay	Number of trains that will wait for delayed feeder trains	Number of trains that will not wait for delayed feeder trains	Number of trains that will run on time
TH = 0	14200	1666	103.30	7.96	0.00	276.60	1789.40
TH = 1	13649	1833	103.30	7.96	45.60	237.00	1783.40
TH = 2	13444	2261	103.30	7.96	93.80	202.20	1770.00
TH = 3	13853	2547	103.30	7.96	150.30	171.40	1744.30
TH = 4	16138	3378	103.30	7.96	213.60	143.40	1709.00
TH = 5	18688	4276	103.30	7.96	299.60	121.50	1644.90

Table 2: The output for different thresholds (TH) of the Waiting Time Rule, the mean of 10 runs of the simulation.

100 runs of the simulation	Total delay of all passengers	Number of passengers that have a delay	Number of delays	Average delay	Number of trains that will wait for delayed feeder trains	Number of trains that will not wait for delayed feeder trains	Number of trains that will run on time
TH = 0	13710	1831	103.43	7.98	0.00	276.57	1789.43
TH = 1	13780	2064	103.43	7.98	45.68	235.02	1785.30
TH = 2	13650	2374	103.43	7.98	93.75	200.11	1772.14
TH = 3	14246	2704	103.43	7.98	148.86	168.80	1748.34
TH = 4	17065	3319	103.43	7.98	215.30	142.28	1708.40
TH = 5	20486	4568	103.43	7.98	296.59	120.27	1649.14

Table 3: The output for different thresholds (TH) of the Waiting Time Rule, the mean of 100 runs of the simulation.

6.4 RESULTS RATIO OF TRANSFERRING PASSENGERS

In this section we will discuss the results when we use the decision rule: Ratio of Transferring Passengers. We can compare the results between the different thresholds (TH), so we can see what the effect is when we use a larger threshold.

We have done 10 runs and 100 runs of the simulation. The duration of the simulation was about 4 minutes for the 10 runs and for the 100 runs it was about 13.5 to 14 minutes. The output of the simulation is the mean of the several runs. So for 10 runs the output is the mean of the 10 runs.

The delays are generated beforehand, so we use for a simulation run the same delays per activity for every threshold. This mean that we have fixed the random delay numbers per trip, so we can compare

the results of the several thresholds of the Ratio of Transferring Passengers decision rule. Every simulation run do have different delays. So the number of delays and the average delays are different between the simulation runs. But are the same for every threshold within a simulation run.

The average number of delays is for the 10 runs of the simulation 103.30 with an average delay of 7.96 minutes for the 10 runs of the simulation. For the 100 runs of the simulation we have an average number of delays of 103.43 and an average delay of 7.98 minutes.

6.4.1 RESULTS FOR THE 10 AND 100 RUNS

In Table 4 and Table 5 all the output is shown for 10 runs and 100 runs, respectively, of the simulation for different thresholds. We can see that the total delay of all passengers and the number of passengers that have a delay follows a downward trend when the threshold will be larger. So the smaller the chance that a connecting train will wait, the smaller the total delay of all passengers and the number of passengers that have a delay. Both make sense, because when there is a no-wait policy, threshold is 100%, there is less total delay of all passengers and the number of passengers that have a delay will be smaller than when a connecting train wait for a delayed feeder train.

When we look at the number of trains that will wait for delayed feeder trains, we see that this number will be smaller when the threshold will be larger. Note that the number of trains that will wait for delayed feeder trains is zero when we use a threshold of 100%, this corresponds to a no-wait policy. On the other hand the number of trains that will not wait for delayed feeder trains will be following a slightly downward trend when the threshold will be larger. The number of trains that will run on time will be larger when the threshold will be larger.

6.4.2 COMPARING THE OUTPUT FOR 10 AND 100 RUNS

Now we can compare the output for 10 runs and 100 runs of the simulation. The number of delays and the average delay are a little bit different for the 10 runs and 100 runs of the simulation. The total delay of all passengers for the several thresholds are about the same for the 10 runs and 100 runs of the simulation. The number of passengers that have a delay differs slightly between the 10 runs and 100 runs of the simulation. This difference can be caused by the difference in delays between the 10 runs and 100 runs of the simulation. Because the delays are random and thus different when we use 10 runs or 100 runs of the simulation.

When we compare the number of trains that will wait for delayed feeder trains, we can see that the average numbers of the number of trains that will wait for delayed feeder trains, the number of trains that will not wait for delayed feeder trains and the number of trains that will run on time do differ slightly from each other. Especially for the thresholds of 10%. This is a little bit strange, but it can be the case that the delays for 10 runs of the simulation are on average higher for the delayed feeder trains that fall within the threshold than for the 100 runs of the simulation.

10 runs of the simulation	Total delay of all passengers	Number of passengers that have a delay	Number of delays	Average delay	Number of trains that will wait for delayed feeder trains	Number of trains that will not wait for delayed feeder trains	Number of trains that will run on time
TH = 0%	21266	2598	103.30	7.96	96.90	306.80	1662.30
TH = 10%	19705	2259	103.30	7.96	59.80	307.30	1698.90
TH = 20%	16316	1905	103.30	7.96	29.20	287.50	1749.30
TH = 30%	14966	1728	103.30	7.96	21.90	283.10	1761.00
TH = 40%	14918	1718	103.30	7.96	16.40	281.50	1768.10
TH = 50%	14837	1700	103.30	7.96	11.30	275.70	1779.00
TH = 60%	14200	1666	103.30	7.96	7.30	277.10	1781.60
TH = 70%	14200	1666	103.30	7.96	5.50	276.90	1783.60
TH = 80%	14200	1666	103.30	7.96	5.40	277.00	1783.60
TH = 90%	14200	1666	103.30	7.96	4.50	277.60	1783.90
TH = 100%	14200	1666	103.30	7.96	0.00	276.60	1789.40

Table 4: The output for different thresholds (TH) of the Ratio of Transferring Passengers, the mean of 10 runs of the simulation.

100 runs of the simulation	Total delay of all passengers	Number of passengers that have a delay	Number of delays	Average delay	Number of trains that will wait for delayed feeder trains	Number of trains that will not wait for delayed feeder trains	Number of trains that will run on time
TH = 0%	20390	2540	103.43	7.98	94.42	302.38	1669.20
TH = 10%	18640	2351	103.43	7.98	48.99	296.79	1720.22
TH = 20%	16075	2146	103.43	7.98	28.84	287.72	1749.44
TH = 30%	14624	2001	103.43	7.98	19.61	281.62	1764.77
TH = 40%	14621	2000	103.43	7.98	12.84	278.83	1774.33
TH = 50%	14257	1982	103.43	7.98	9.22	277.34	1779.44
TH = 60%	14080	1885	103.43	7.98	6.66	278.06	1781.28
TH = 70%	14080	1885	103.43	7.98	5.05	277.80	1783.15
TH = 80%	14070	1884	103.43	7.98	4.33	277.88	1783.79
TH = 90%	14070	1884	103.43	7.98	3.36	277.86	1784.78
TH = 100%	13710	1831	103.43	7.98	0.00	276.57	1789.43

Table 5: The output for different thresholds (TH) of the Ratio of Transferring Passengers, the mean of 100 runs of the simulation.

6.5 COMPARING THE WTR AND RTP DECISION RULES

Now we have the results of the Waiting Time Rule and the Ratio of Transferring Passengers decision rule, we can compare these decision rules. Note in the first place that all results of the Waiting Time Rule for a threshold of 0 are exactly the same as all results of the Ratio of Transferring Passengers decision rule for a threshold of 100%. This results have to be the same, because in both cases we have a no-wait policy. When we look at the total delay of all passengers for 10 runs of the simulation, we can see that the values for the Ratio of Transferring Passengers decision rule are somewhat larger than the values for the Waiting Time Rule. For 100 runs of the simulation, the values of the total delay of all passengers are

about the same for both decision rules. The values for the number of passengers that have a delay and for the number of trains that will wait for delayed feeder trains are larger for the Waiting Time Rule than for the Ratio of Transferring Passengers decision rule when we look at all thresholds. This can be the case, because with the Waiting Time Rule we will look at the number of minutes that a connecting train can wait and with the Ratio of Transferring Passengers we will look at the passengers that will transfer to a train in comparison with the passengers that are in the connecting train. When we look at the number of trains that will not wait for delayed feeder trains, we can see that with the Waiting Time Rule the values become smaller when the threshold will be larger, so when a connecting train will wait longer, and for the Ratio of Transferring Passengers decision rule the values will be larger when the chance that a connecting train waits will be larger. The values for the number of trains that will run on time are for both decision rules about the same.

7 CONCLUSION

To imitate the real world when trains can have a delay, we have made a simulation with the timetable from NS. First we had to make some assumptions to make the simulation somewhat easier. Also we have generated random delays for 5% of the trips, the delay numbers are between 1 and 15 minutes. The average number of delays we have used for the 10 runs of the simulation is 103.30 and the average delay we have used is 8.13 minutes. To make decisions if a connecting train better can wait for a delayed feeder train or depart, we have used the following decision rules: Waiting Time Rule and Ratio of Transferring Passengers. For this decision rules we have used several thresholds. With the simulation and a decision rule we have obtained a new updated timetable. With this new timetable we have calculate the shortest path of the passengers with a shortest path algorithm. We compared this shortest path with the shortest path of the passengers when there are no delays and every train will run on schedule. With this result we have obtained the total delay of all passengers and the number of passengers that have a delay.

The Waiting Time Rule have shown that the larger the threshold, so the longer a connecting train will wait for delayed feeder trains, the slightly larger the total delay of all passengers. Also the number of passengers that have a delay and the number of trains that will wait for delayed feeder trains will be larger when the threshold will be larger. This all makes sense, because a connecting train will wait longer with a larger threshold. Because this connecting train will have a delay because of the waiting and passengers can miss their connections at other stations.

The Ratio of Transferring Passengers decision rule have shown that when the threshold will be larger, so the chance of waiting of a connecting train will be smaller, the smaller the total delay of all passengers, the number of passengers that have a delay and the number of trains that will wait for delayed feeder trains.

Note that for both decision rules the number of trains that will wait for delayed feeder trains will be zero when there is a no-wait policy. The result of this no-wait policy is generally a smaller total delay of all passengers and a smaller number of passengers that have a delay.

More research can be done, like using other and more decision rules and compare these with each other. Also we can make some less assumptions to come closer to the real world setting. Now we have not taken into account the number of stations and if a station is free or not, this will make the simulation somewhat more difficult.

8 REFERENCES

T. Dollevoet. Delay Management and Dispatching in Railways. 2013.

Syllabus Wiskundige Methoden, 2007.