Delay Management with Re-Routing of Passengers and Capacity Constraints

Author: P. van Leeuwen -311127-

Supervisor: dr. D. Huisman
Co-reader: dr. W. van den Heuvel

Bachelor Thesis of Econometrics & Operations Research, Logistics,
Econometric Institute, Erasmus School of Economics,
Erasmus University Rotterdam, the Netherlands

ERASMUS UNIVERSITEIT ROTTERDAM

July 1, 2011

ABSTRACT

Delay management focuses on wait-depart decisions, in which the main question is whether trains should wait for a delayed train or should depart on time. In the traditional delay management models passengers are assumed to always take their originally planned route. In the latest literature however re-routing of passengers is incorporated in the delay management model. Passengers are now assumed to be rational: they always take the shortest route to their destination, also in case of a delay. The delay management model with re-routing (DMwRR) obtained significant improvements to the passengers' traveling times. However, in this model, capacity constraints are not taken into account. Therefore in this thesis, we will present three new models with different capacity constraints. The first model includes "platform constraints", which make sure that trains can only arrive at a station at a certain platform if that platform is empty. The second model includes "not overhaul" constraints, which make sure that trains can only overhaul each other when the next station is a 'big' station (a station with more than four tracks). The third and last model includes both "platform" constraints and "not overhaul" constraints.

At the end experiments based on real-world data from the NS (*Nederlandse Spoorwegen*, Dutch railways) show that adding capacity constraints to the DMwRR-model makes the model even more complex than it already was. For small real-world instances it turned out that the constraints do not influence the optimal solution. The large real-world instances turned out to be unsolvable due to the high solving times.

Keywords: Public Transportation, Delay Management, Re Routing, OD-pairs, Capacity Constraints, Overhaul, Platform

TABLE OF CONTENTS

1 INTRODUCTION

Every day more than one million people in the Netherlands travel by train. One of the reasons for this high number of people that travel by train is that train services in the Netherlands are very frequent. Next to this another reason is that the Dutch rail network is one of the densest in the world with approximately 3,000 kilometers of railway track and some 400 stations.

Like most European railway companies the NS (*Nederlandse Spoorwegen*, Dutch railways) has a cyclic timetable, which means that the timetable is repeated after a certain period of time. This ensures a high frequency and an easy to remember timetable for the passengers. A disadvantage of the timetable in the Netherlands is that passengers often have to transfer trains, since it is impossible to have routes without transfers between all stations. A passenger making a trip from one station to another is called an OD-pair (origin-destination pair). With transfer we mean that the passenger gets off his train at a certain station and gets on another train that departs at the same station some time later, in order to reach his destination as fast as possible. To reduce the inconvenience of transfers the timetable is constructed in such a way that transfer-related times are minimized. This sounds advantageous, but with an example you can easily see that this also has a big disadvantage.

For example a passenger wants to travel from Zwolle to Gouda, but due to the fact that there is no direct connection, the passenger has to transfer at Amersfoort. Now suppose that the train from Zwolle to Amersfoort is delayed, because the timetable is constructed in such a way that the train Amersfoort – Gouda departs shortly after the train Zwolle–Amersfoort, the passenger now has a high probability to miss his next train to Gouda. Passengers who have to transfer hope that the train Amersfoort – Gouda will wait, however, this causes unnecessary delays for other passengers. The main delay management question in this example is: *should the train Amersfoort – Gouda wait for the train Zwolle – Amersfoort, or should it depart on time*? These decisions are called wait-depart decisions.

In this thesis delay management is treated as an offline problem, this means that all delays are known a priori. We will only focus on delays in arrival events, which cause an increase of traveling time between two stations. In case of delays the timetable is not longer feasible and has to be updated. Dollevoet et al. (2010) incorporated re-routing within the delay management process (this model is called DMwRR). Re-routing means that if a passenger misses his connection because of a delay, he often can take another line that brings him faster to his destination than waiting for the next connection. By incorporating re-routing it is possible for passengers to behave in such a way in the model.

Due to the limited capacity of the Dutch rail network there are a lot of capacity constraints that could also be integrated in the DMwRR-model. In this thesis we will consider two kinds of capacity constraints. First we will integrate so called “platform” constraints, these ensure that trains cannot use the same platform at a station at the same time. Next we will integrate so called “not overhaul” constraints, these ensure that trains cannot overhaul always and everywhere. We say that trains can only overhaul between two stations if the next station is a ‘big’ station, a station with more than four tracks.

The data we used for this thesis is the same as in Dollevoet et al. (2010) and consists of parts of the Dutch rail network.

The obtained solutions cannot be compared in terms of the objective function (total passengers' delay), because this would be the same as comparing different models to each other. A better way to compare, is comparing the models in terms of delayed events, delayed OD-pairs and delayed passengers.

The remainder of this thesis is structured as follows. In Chapter 2, we present a more detailed problem description, we give an overview of the assumptions that are made for the model and we discuss the data that is used. A brief overview of the relevant literature on this subject is given in Chapter 3. In Chapter 4 we define mathematical formulations, which we used to solve all four problems. An overview of the cases we used for testing our formulations can be found in Chapter 5. Computational results will be discussed in Chapter 6, followed by Chapter 7 about the shortcomings of the model. We will end with a conclusion and options for further research in Chapter 8.

2 PROBLEM DESCRIPTION

In this chapter we will describe the problem we deal with. First we will describe delay management with re-routing (section 2.1). Second we will define the objective of our research (section 2.2): namely adding capacity constraint to the problem. After that we will discuss the available data (section 2.3), give an overview of the assumptions we made (section 2.4) and in the last section (section 2.5) we will introduce graph representations.

2.1 General problem description: delay management with re-routing

In classical delay management, in case of delays, passengers are assumed to stick to their predefined routes. Assuming rational behavior of the passengers, their predefined routes are always the fastest path to arrive at their destination. Because passengers often have to change trains, it is possible that they miss their connection in case of a delay. If a connection on such a route is dropped, in the original delay management problem, it is assumed that passengers wait until the same connection takes place one cycle time later. This assumption is usually not valid in practice, often there might be faster routes for passengers to arrive at their destination. Passengers choosing for the earlier alternative is called re-routing. In the delay management model with re-routing (DMwRR) the sum of all delays over all origin-destination pairs (OD-pairs) is minimized, with the assumption that passengers always take the shortest path, also in case of delays.

2.2 Objective: adding capacity constraints to the model

DMwRR is described in Dollevoet et al. (2010). However, in practice, the limited capacity of the Dutch rail network also has a large impact on the wait-depart decisions. Therefore, capacity constraints should also be integrated in DMwRR. There are a lot of capacity constraints that could be added to the model, but in this thesis we will consider two of them.

2.2.1 “Platform” constraints

For simplicity it is assumed that the Netherlands has a double track rail network, so that trains in both directions have their own track. Hereby, a delayed train in one direction does not delay the train in the opposite direction. We also assume that trains in the same direction and on the same route, share the same platform at a station, so if a train is delayed this can delay the next train that will arrive at the same platform. The first capacity constraint makes sure that trains can only arrive at a station, if its platform is empty. Therefore the time between a departing and arriving train, on the same station in the same direction and from the same platform, is at least three minutes. For simplicity it is assumed that trains will always arrive at the platform that is assigned beforehand, so if the platform is not empty the next train will be delayed.

2.2.2 "Not overhaul" constraints

In the Netherlands we have two types of trains: intercity trains and regional trains. Intercity trains drive with a faster speed than regional trains. In DMwRR intercity trains can overhaul regional trains everywhere; however in practice of course this is not possible. The second capacity constraint makes sure that intercity trains can only overhaul regional trains on their way to a 'big station', a station with more than four tracks. Therefore when a regional train (from A to B) departs from station A before a intercity train (also from A to B) departs from station A, the intercity train can only arrive earlier at station B than the regional train if station B is a 'big station'.

2.3 Data

In this thesis we will use the same data as used in Dollevoet et al. (2010). We will also use the same cases to do computational experiments, to compare the results of Dollevoet et al. (2010) with our results. In all six cases we consider a part of the railway network in the Netherlands during a period in the late evening. For every case the data consists of the following information:

- **The set of stations *S* where trains can arrive and depart from.**
  For every station an index, full name and size of the station (big or small) is given.

- **The set of all origin-destination pairs *P*.**
  For every OD-pair an index, origin station, destination station, planned arrival time at the destination and the expected number of passengers are given.

- **The set of events *E*: the departure and arrival events of the trains at stations.**
  For every event an index, train number, station, planned time and type (Arrival or Departure) are given.

- **The set of activities *A*: relations between the departures and arrivals of the trains**.
  For every activity an index, from, to, planned time to perform the activity and the type (Drive, Dwell, Transfer, Origin or Destination) are given.
  The first three types activities are: driving from one station to the next, waiting at a station to let the passengers get on and off the train and passengers transferring trains. All three connect two nodes. However, the origin (destination) activities make sure that passengers really depart (arrive) from (at) the origin (destination) nodes. Therefore origin activities connect nodes with OD-pairs.

- **The set of source delays d: simulated delay scenarios for arrival events.**
  For every arrival event there is a delay, given the different delay scenarios.

2.4 Overview of the assumptions

The following assumptions are made:

(1) Passengers are rational: they always take the shortest path, also in case of delays. From this it follows that re-routing of passengers is incorporated in the delay management process.

(2) All delays are known before the optimization process starts, so it is an offline problem.

(3) Trains can only get delayed during an arrival-event. When delays occur they cause an increase of the traveling time between two stations.

(4) The number of passengers who want to travel from a given origin to a destination at a certain time is known.

(5) There are two types of trains: ‘intercity’ and ‘regional’. All trains of the same type have the same speed and intercity trains have a higher speed than regional trains.

(6) The entire railway network consists of double track railway.

(7) There are two types of directions: ‘forward’ and ‘backward’. All trains in the same direction and on the same route share the same platform at a station. It is not possible for trains to arrive at another platform than assigned on forehand.

(8) The minimal time between an arrival and departure on a platform at a station has to be at least 3 minutes

(9) There are two types of stations: ‘big’ and ‘small’. Where big stations have more than four tracks and small stations have four or less tracks.

(10) Intercity trains can only overhaul regional trains on their way to ‘big’ stations.

2.5 Graph representation

In this thesis the used data can be represented as a graph G = (A,E), where A is a set of arcs/activities and E is a set of nodes/events. The arcs represent the time that is needed to go from one node to another node.

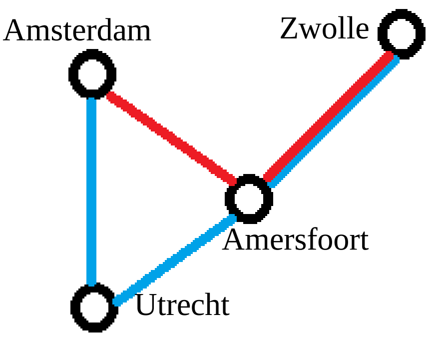

Figure 1 Graph representation with two possible routes from Zwolle to Amsterdam.

In Figure 1 we see two possible routes to go from Zwolle to Amsterdam CS. Passengers can follow the red line and travel with the train from Zwolle to Utrecht CS and transfer at Amersfoort to the intercity from Amersfoort to Amsterdam CS. Passengers can also follow the blue line and travel with the train from Zwolle to Utrecht CS and transfer at Utrecht CS to the intercity from Utrecht CS to Amsterdam CS.

In Figure 2 an example of an event-activity network is given for the route from Zwolle to Amersfoort. The solid-lined arcs represent the time that is needed to do one of the following three activities: drive, wait, transfer. The dashed-line arcs represent the origin and destination arcs, which makes sure that passengers depart from their origin and arrive at their destination. The squared nodes are the arrival and destination events, the oval nodes are the origin and destination events.

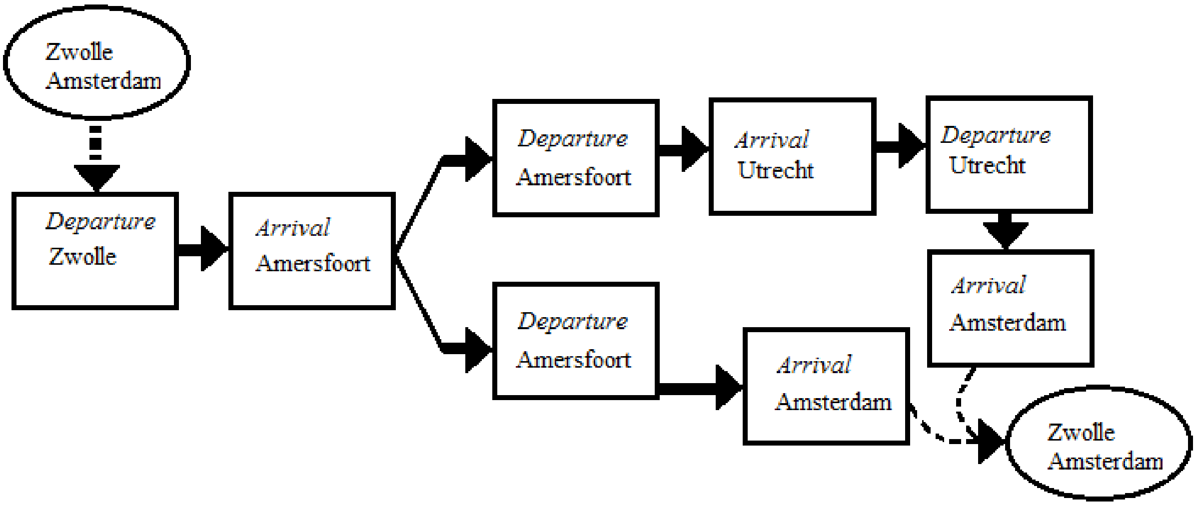

Figure 2 Example of an event-activity network for passengers travelling from Zwolle to Amsterdam.

Figure 2 shows all arcs and nodes for passengers travelling from Zwolle to Amsterdam. The arcs are measured in minutes. The shortest path can be calculated by summing up the lengths of the arcs for every route, and take the minimum. When trains get delayed some transfers would no longer be available, so that some routes are not longer available and the shortest path should be calculated again.

Note that in Figure 2 we did only consider one possible departure time, therefore we did not include the starting time in the origin and destination events. For every possible starting time such a network as in Figure 2 is available.

3 LITERATURE OVERVIEW

Although, to the best of our knowledge, the problem we consider in this thesis has never been handled before in the scientific literature, there are some articles that provide background information about delay management in general, capacity constraints and re-routing. In this chapter we will give a brief overview of some articles that are interesting for our own case.

In Nachtigall (1998) the event-activity network for timetabling problems is introduced. The author states that a railroad network can be modeled by a graph, with a set of nodes and a set of arcs. In which nodes represent stations and arcs represent railroad tracks connection these points.

Schöbel (2001) uses the concept described in Nachtigall (1998) and makes use of the following three activities or arcs: driving (a train driving between two consecutive stations), waiting (a train waiting to let passengers get on and off the train) and changing (passengers transferring from one train to another when there is no direct connection between origin and destination).

In Schöbel (2007) the integer programming formulation presented in Schöbel (2001) has been further developed. A never-meet property is added, so that cycles in the graph are not allowed. This property requires that the paths of delayed customers will not meet, so that in the objective functions delays are not counted twice.

In Schöbel (2009) capacity constraints are added to the delay management problem, because the limited capacity of the tracks has been neglected so far. To model these constraints headway activities are added to the network and algorithms are made to take these constraints into account.

In Schachtebeck (2009) capacity constraints are also added to the delay management problem by adding headway activities to the network. However in this article the capacity constraints are really added to the programming formulation.

All the literature aforementioned assumes that passengers will stick to their predefined routes. However, in practice passengers behave rational: they will always take the shortest route to their destination, what is called re-routing. Therefore, the most relevant article for our thesis is Dollevoet et al. (2010). They describe a delay management problem with re-routing, but without capacity constraints. To model the re-routing now five activities are needed: driving, waiting, changing, origin and destination. The last two activities, in combination with origin and destination nodes, make sure that passenger will always take the shortest route to their destination.

4 MODELS FOR DELAY MANAGEMENT WITH RE-ROUTING

In this chapter we will present four different models for delay management with re-routing. We will start with the basic model presented in Dollevoet et al. (2010) and then present three different extensions to this model. In the first extension the capacity constraints of platforms are added. In the second extension the capacity constraints of not overhauling when driving to a 'small' station are added. And in the third extension both the "platform" and "not overhaul" constraints are added. In section 4.1 we will describe all sets, parameters, variables and constraints that are used in the basic model. In section 4.2 we will describe the extra sets, parameters, variables and constraints that are needed to add the "platform" constraints to the basic model. In section 4.3 we will describe the extra sets, parameters, variables and constraints that are needed to add the "not overhaul" capacity constraints to the basic model. In section 4.4 an overview of the 4 models is given.

4.1 Basic model: sets, parameters, variables, objective function and constraints

Sets

The set of OD-pairs P, with $p \in P$ an OD-pair from the set P.

The set of events E, with $e \in E$ an event from the set $E = E_{arr} \cup E_{dep} \cup E_{origin} \cup E_{dest}$

The set of activities A, with $a \in A$ an activity from the set $A = A_{drive} \cup A_{wait} \cup A_{change} \cup A_{origin} \cup A_{dest}$

Parameters

For set P

SP_p: planned arrival time (in minutes after 00:00) for OD-pair p of the passengers if there are no delays, this is a lower bound for the arrival time of the passengers for OD-pair p

A_p: planned arrival time (in minutes after 00:00) for OD-pair p, where the route from origin to destination only contains 'safe transfers' (these are transfers with a transfer time of at least 2 + $\max_{e \in E} d_e$). For this route it is not possible to get delayed by missing a transfer.

w_p: number of passengers associated to an OD-pair $p \in P$.

For set E

π_e: planned time (in minutes after 00:00) for events $e \in E_{arr} \cup E_{dep}$
time at which a passenger of OD-pair p arrives at his departure station for events $e = org(p) \in E_{origin}$
time at which a passenger of OD-pair P arrives at his destination for events $e = dest(p) \in E_{dest}$

d_e: delay in minutes for event $e \in E_{arr}$

For set A

L_a: minimal time that is needed to perform activity $a \in A$, with $L_a = 0$ for $a \in A_{origin} \cup A_{dest}$

Variables

$z_a: \begin{cases} 1 & \text{if connection a is maintained} \\ 0 & \text{otherwise} \end{cases}$ with $a \in A_{change}$

$q_{ap}: \begin{cases} 1 & \text{if activity a is used by passengers in p} \\ 0 & \text{otherwise} \end{cases}$ with $a \in A, p \in P$

x_e: rescheduled time that event e takes place, with $e \in E_{arr} \cup E_{dep}, x_e \in \mathbb{N}$

t_p: arrival time for and OD-pair pair p (in minutes after 00:00) with $p \in P$

Constraints

(1) $x_e \geq \pi_e + d_e$ $\quad \forall\, e \in E_{arr} \cup E_{dep}$

(2) $x_e \geq x'_e + L_a$ $\quad \forall\, a = (e', e) \in A_{drive} \cup A_{wait}$

(3) $x_e \geq x'_e + L_a - M_1(1 - z_a)$ $\quad \forall\, a = (e', e) \in A_{change}$

(4) $q_{ap} \leq z_a$ $\quad \forall\, p \in P, a \in A_{change}$

(5) $\sum_{a \in \partial^{out}(e)} q_{ap} = 1$ $\quad \forall\, e = Org(p) \in E_{org}$

(6) $\sum_{a \in \partial^{out}(e)} q_{ap} = \sum_{a \in \partial^{in}(e)} q_{ap}$ $\quad \forall\, p \in P, e \in E_{arr} \cup E_{dep}$

(7) $\sum_{a \in \partial^{in}(e)} q_{ap} = 1$ $\quad \forall\, e = Dest(p) \in E_{dest}$

(8) $t_p \geq x_e - M_2\left(1 - q_{ap}\right)$ $\quad \forall\, e = Dest(p) \in E_{dest}, a \in \partial^{in}(e)$

(9) $z_a \in \{0,1\}$ $\quad \forall\, a \in A_{change}$

(10) $q_{ap} \in \{0,1\}$ $\quad \forall\, p \in P, a \in A$

(11) $x_e \in \mathbb{N}$ $\quad \forall\, e \in E_{arr} \cup E_{dep}$

(12) $t_p \in \mathbb{N}$ $\quad \forall\, p \in P$

Where $M_1 = \max_{e \in E} d_e$ and
$M_2 = A_p - SP_p + 2 \cdot \max_{e' \in E} d_{e'}$,
$a = (e', e)$: an arc from node e' to node e.
$\partial^{in}(e)$: set of arcs into node e
$\partial^{out}(e)$: set of arcs out of node e

Constraints (1) imply that for all nodes the rescheduled time has to be at least the original planned time increased with the delay for this node.

Constraints (2) and (3) make sure that for all arcs the delays are propagated through the network correctly. When a connection is maintained ($z_a = 1$) constraints (2) and (3) are the same, they both imply that the delay from the start of activity a is transferred to its end. In the other case, when $z_a = 0$, constraints (3) transfer delays from the feeder train to the connecting train.

Constraints (4) – (7) are the constraints of the shortest path problem for each OD-pair p. Constraints (4) imply that for all arcs and OD-pairs changing activities can only be used if the connection is maintained. Constraints (5) and (7) make sure that all OD-pairs have exactly one origin and one destination. Constraints (6) make sure that for all OD-pairs and nodes the number of arcs going into these nodes are the same as the number of arcs going out of these nodes.

Constraints (8) are, like constraints (4) – (7), needed to take the routing decisions into account. These make sure that the arrival time for an OD-pair is at least the rescheduled time of the destination event, when the destination-activity is used by passengers in the OD-pair. When the activity is not used by passengers in the OD-pair, the value of t_p is unrestricted.

Constraints (9) and (10) make sure that the decision variables z_a and q_{ap} are binary. Constraints (11) and (12) make sure that the decision variable x_e and t_p are positive integers.

4.2 New sets, parameters, variables, objective function and constraints for the "platform" constraints

Sets

The set of combinations E^{COMB}, with $(e_1, e_2) \in E^{COMB}$ an combination of two events from the set E that meets the following conditions:

- Event e_1 is an arrival event and e_2 is an departure event $e_1 \in E_{arr}$, $e_2 \in E_{dep}$
- The events e_1 and e_2 have the same direction: $D_{e_1} = D_{e_2}$
- The events e_1 and e_2 use the same route: $R_{e_1} = R_{e_2}$
- The events e_1 and e_2 happen on the same train station: $TS_{e_1} = TS_{e_2}$
- The events e_1 and e_2 have an unequal train number: $TN_{e_1} \neq TN_{e_2}$

Parameters

TN_e: train number of event e, with $e \in E_{arr} \cup E_{dep}$

Every train in the Netherlands has a unique train number, which gives information about the direction of the train and the stations that it will visit. It also gives information about the moment the train drives.

D_e: direction of event e, forward(0) or backward(1), with $e \in E_{arr} \cup E_{dep}$

The direction of an event can be determined with the train number. We state that trains with even train numbers are going in 'forward direction' and trains with odd train numbers are going in 'backward direction'.

R_e: route of event e, we assume that trains which share the same platform on all joint stations follow the same route. We therefore grouped the train numbers in such a way that trains that drive more or less the same route share the same number of a route. We need this parameter in the model, because direction is not the only factor that determines the platform a train will use at a certain station. The grouping of the routes is based on the Dutch railway map. Note that intercity and regional trains, visiting the same train stations, follow the same route.

TS_e: train station where event e takes place, with $e \in E_{arr} \cup E_{dep}$
Every train station in the Netherlands has a unique train station number.

Variables
For every combination of an arrival event e_1 and departure event e_2 : $(e_1, e_2) \in E^{COMB}$ we introduce a decision variable $y_{e_1 e_2}$.

With $y_{e_1 e_2} = \begin{cases} 1 & \text{if event } e_1 \text{ takes place after event } e_2 \\ 0 & \text{otherwise} \end{cases}$ and $e_1 \in E_{arr}, e_2 \in E_{dep}$

Constraints

(13) $x_{e1} - x_{e2} \geq 3 + (y_{e_1 e_2} - 1) \cdot M$ $\quad \forall\, (e_1, e_2) \in E^{COMB}$

(14) $y_{e_1 e_2} + y_{e_2 e_1} = 1$ $\quad \forall\, (e_1, e_2) \in E^{COMB}$

(15) $y_{e_1\, e_2} \in \{0,1\}$ $\quad \forall\, (e_1, e_2) \in E^{COMB}$

Constraints (13) make sure that a train can only arrive at its assigned platform if the previous train has left this platform at least three minutes before. It is assumed that three minutes are reasonable for the time between an arrival and departure.

Constraints (14) make sure that either event e_1 takes place after event e_2 or event e_2 takes place after event e_1, but not both.

Constraints (15) make sure that the decision variables $y_{e_1\, e_2}$ are binary.

4.3 New sets, parameters, variables, objective function and constraints for the "not overhaul" constraints

Sets
The set of combinations A^{COMB}, with $a_1 = (e_1, e'_1)$, $a_2 = (e_2, e'_2)$ and $(a_1, a_2) \in A^{COMB}$ an combination of two arcs from the set A that meets the following conditions:

- The arcs a_1 and a_2 have the same departure station $\quad TS_{e_1} = TS_{e_2}$
- The arcs a_1 and a_2 have the same arrival station $\quad TS_{e'_1} = TS_{e'_2}$
- The arcs both arrive at a small arrival station $\quad TS_{e'_1} = 1$
  $TS_{e'_2} = 1$ is implicitly stated because a_1 and a_2 have the same arrival station.
- The events e_1 and e_2 have the same direction: $\quad D_{e_1} = D_{e_2}$
- The events e_1 and e_2 use the same route: $\quad R_{e_1} = R_{e_2}$

Because all regional trains drive with the same speed, it is not possible for regional trains to overhaul each other or an intercity. Therefore it is not necessary to take the constraint 'It is only possible for intercity trains to overhaul' into account.

Variables

For every combination of two driving arcs $a_1 = (e_1, e_1')$ and $a_2 = (e_2, e_2')$: $(a_1, a_2) \in A^{COMB}$ there exist two decision variables $v_{e_1e_2}$ and $u_{e_1'e_2'}$.

$$v_{e_1e_2} = \begin{cases} 1 & \text{if event } e_1 \text{ takes place after event } e_2 \\ 0 & \text{otherwise} \end{cases} \quad \text{and } e_1, e_2 \in E_{dep}$$

$$a_1 = (e_1, e_1'), a_2 = (e_2, e_2') \in A_{drive}$$

$$u_{e_1'e_2'} = \begin{cases} 1 & \text{if event } e_1' \text{ takes place after event } e_2' \\ 0 & \text{otherwise} \end{cases} \quad \text{and } e_1', e_2' \in E_{arr}$$

$$a_1 = (e_1, e_1'), a_2 = (e_2, e_2') \in A_{drive}$$

Constraints

(16) $x_{e_1} - x_{e_2} \geq \left(v_{e_1e_2} - 1\right) \cdot M \qquad \forall\ (a_1, a_2) \in A^{COMB}$

(17) $x_{e_1'} - x_{e_2'} \geq \left(u_{e_1'e_2'} - 1\right) \cdot M \qquad \forall\ (a_1, a_2) \in A^{COMB}$

(18) $v_{e_1e_2} + v_{e_2e_1} = 1 \qquad \forall\ (a_1, a_2) \in A^{COMB}$

(19) $u_{e_1'e_2'} + u_{e_2'e_1'} = 1 \qquad \forall\ (a_1, a_2) \in A^{COMB}$

(20) $u_{e_1'e_2'} = v_{e_1e_2} \qquad \forall\ (a_1, a_2) \in A^{COMB}$

(21) $v_{e_1e_2} \in \{0,1\} \qquad \forall\ (a_1, a_2) \in A^{COMB}$

(22) $u_{e_1'e_2'} \in \{0,1\} \qquad \forall\ (a_1, a_2) \in A^{COMB}$

Constraints (16) and (17) make sure that when the departure event e_1 takes place after departure event e_2, the rescheduled time of event e_1 is not smaller than the rescheduled time of event e_2.Then the arrival event e_1' takes place after arrival event e_2', the rescheduled time of event e_1' is not smaller than the rescheduled time of event e_2'. When the departure (arrival) event $e_1(e_1')$ does not take place after departure (arrival) event $e_2(e_2')$, the time between these events has no restrictions.

Constraints (18) make sure that either event e_1 takes place after event e_2 or event e_2 takes place after event e_1 (or at the same time), but not both. Constraints (19) make sure that either event e_1' takes place after event e_2' or event e_2' takes place after event e_1' (or at the same time), but not both.

Constraints (20) make sure that if a train leaves a station earlier than another train, this train will arrive earlier at the next station than that other train. This only applies when the next station is a small station. When the next station is a 'big' station it is possible for trains to overhaul between the current station and the next (big) station.

Constraints (21) and (22) make sure that the decision variables $v_{e_1e_2}$ and $u_{e_1'e_2'}$ are binary.

4.4 Overview of the four models

Model 1: Delay management with re-routing

$$\min \sum_{p \in P} w_p (t_p - SP_p)$$

Subject to constraints (1) – (12)

Model 2: Delay management with re-routing and "platform" constraints

$$\min \sum_{p \in P} w_p (t_p - SP_p)$$

Subject to constraints (1) – (15)

Model 3: Delay management with re-routing and "not overhaul" constraints

$$\min \sum_{p \in P} w_p (t_p - SP_p)$$

Subject to constraints (1) – (12) and (16 - 22)

Model 4: Delay management with re-routing, "platform" constraints and "not overhaul" constraints

$$\min \sum_{p \in P} w_p (t_p - SP_p)$$

Subject to constraints (1) – (22)

5 CASES

In this chapter we will describe six different cases that we consider to compare the four integer programming models from Chapter 4. These cases are the same as in Dollevoet et al. (2010). The cases have a logical order: from small to big. After describing the different cases an overview of the characteristics of the cases is given in section 5.6.

In all six cases we consider a part of the railway network in the Netherlands during a period in the late evening. For all cases a graphical representation, of the area that is investigated in the case, is shown. In the figures the big dots represent the big stations which have more than four tracks and the small dots represent the small stations which have four or less tracks.

The train numbers are grouped in such a way that trains that drive more or less the same route share the same route.

5.1 Case I

In the first case, a short time period of 2 hours is taken into account. In this case we focus on a delay for the train from Zwolle to Amersfoort (and Utrecht CS), because this is the only delay that can violate connections. Delays of more than 30 minutes are not interesting, because the next train will be driving 30 minutes later, therefore we will only look at 31 different scenarios: delays from 0 until 30 minutes. In the first case, we consider the average number of passengers, therefore for each OD-pair p we set $w_p = \overline{w}_p$.

The first case consists of both intercity and regional trains in one direction: a regional and intercity train from Amersfoort – Hilversum – A'dam CS. Where the regional train also stops at the smaller stations between Amersfoort, Hilversum and A'dam CS. And two intercity trains from Zwolle – Amersfoort – Utrecht CS and Utrecht CS – A'dam Amstel – A'dam CS.

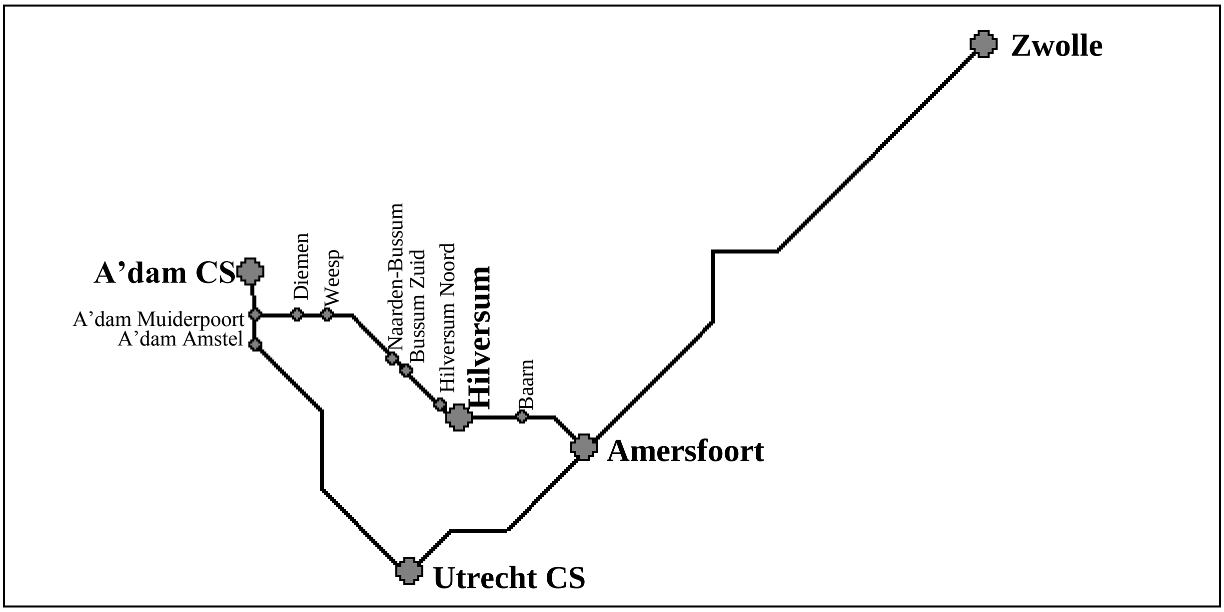

Figure 3 Graph representation of the railway network used in case I.

5.2 Case II

For all other cases than case I, a time period of about 4 hours is taken into account. It does not make sense to consider a longer time period than 4 hours, because all trips of passengers and trains take less than these 4 hours. Also for all other cases than case I, we take a look at 100 simulated delay scenarios where each arrival activity has a probability of 10% to be delayed. When a train is delayed, the delay will be a uniformly distributed integer number between 1 and 15 minutes. Note that for all cases the delay scenarios are different, because they are obtained by simulation.

In the second case, all intercity trains (in both directions) between 8 'big' stations and 2 'small' stations are considered. In Figure 2 you can find a graph representation of the railway network that is used in case II.

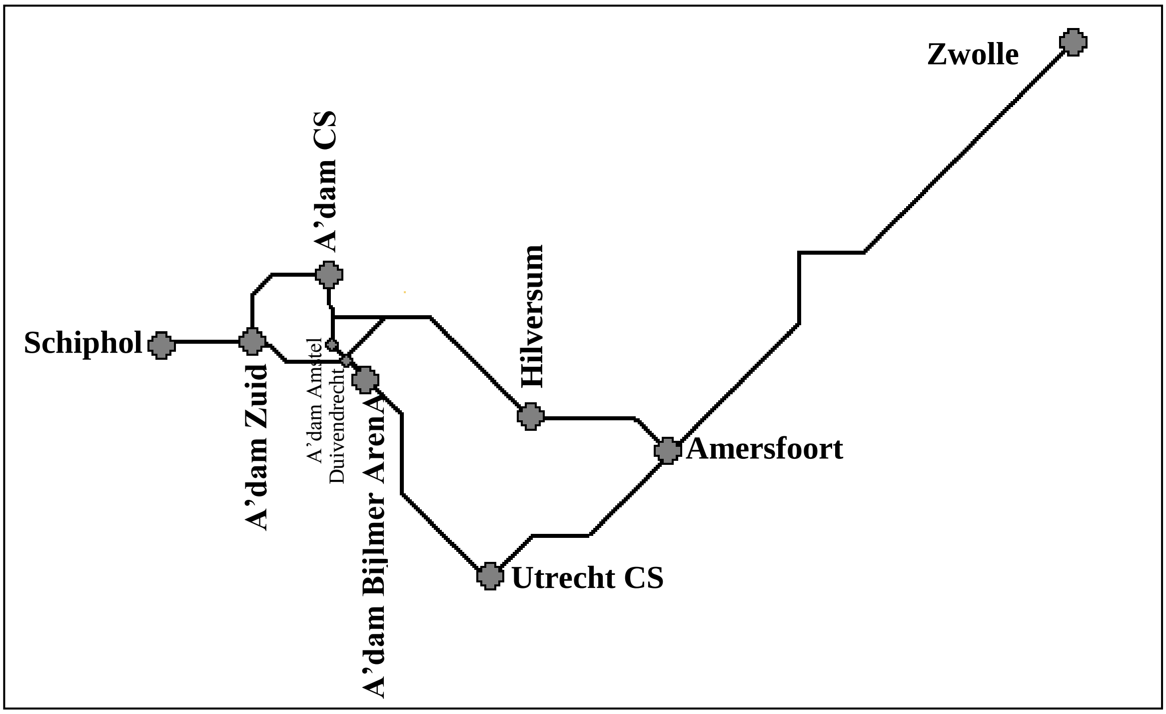

Figure 4 Graph representation of the railway network used in case II.

5.3 Case III

In the third case, some stations are added to the stations of the second case: Den Haag CS, Den Haag HS, Gouda, Leiden CS, R'dam Alexander, R'dam CS are added. The area covered by the stations in the third case is the Randstad, the most densely populated area of the Netherlands. In this case all intercity trains (in both directions) between 13 'big' stations and 3 'small' stations are considered. In Figure 3, on the next page, you can find a graph representation of the railway network that is used in case III.

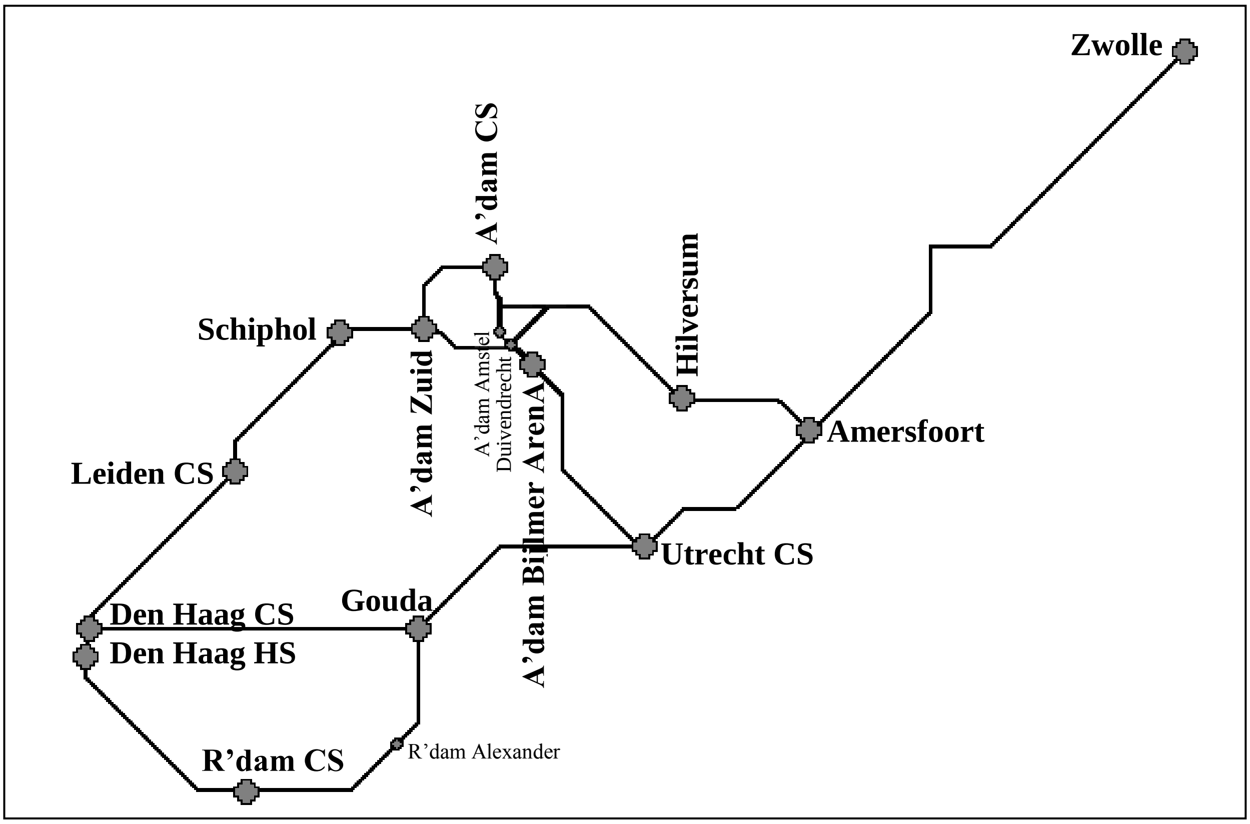

Figure 5Graph representation of the railway network used in case III.

5.4 Case IV & V

In the fourth case, some 'small' stations (and one 'big' station A'dam Sloterdijk) are added to the stations of the second case. In this case all intercity trains and regional trains (in both directions) between 9 'big' stations and 24 'small' stations are considered.

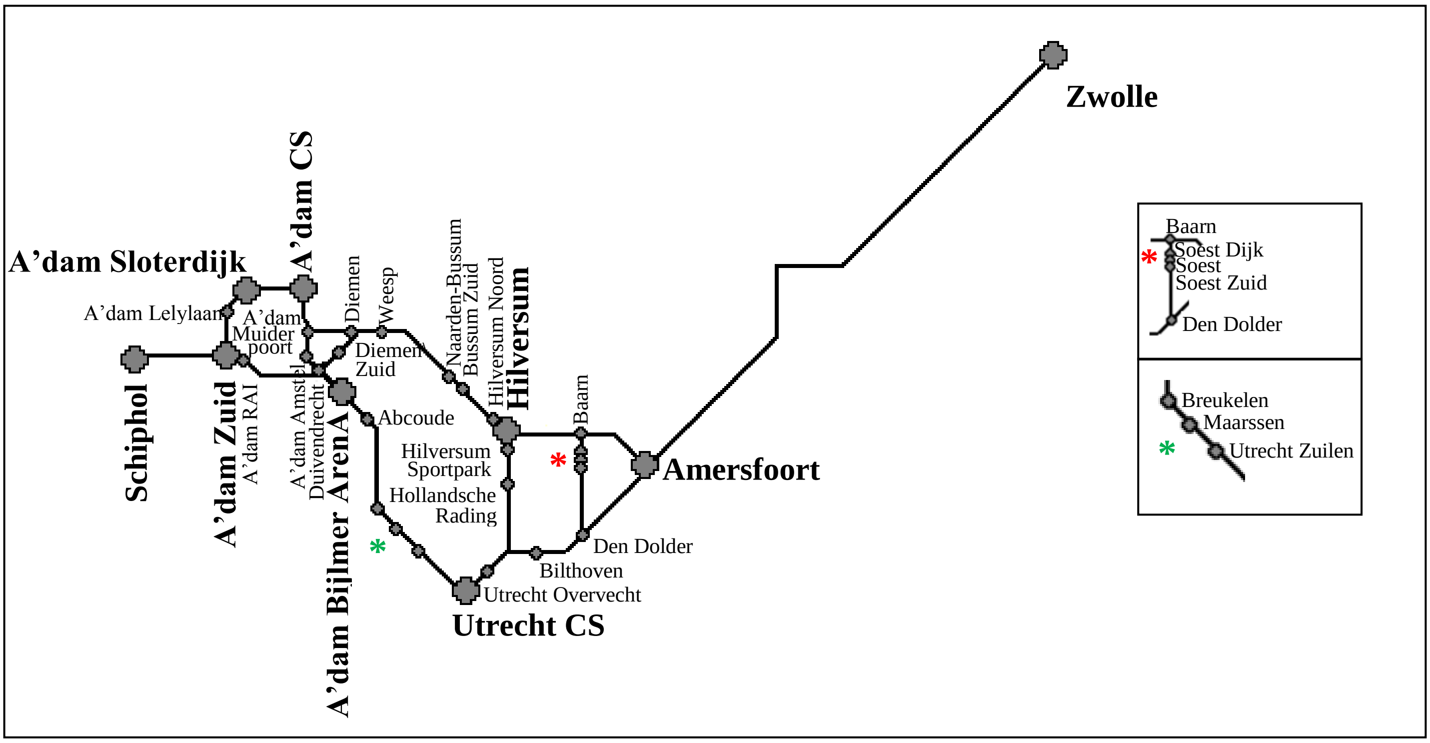

Figure 6 Graph representation of the railway network used in cases IV and V.

By including the regional trains, the number of changing activities and OD-pairs grow. As the regional trains also stop at smaller stations, which are the destinations of few passengers, there are many OD-pairs for which the average number of passengers w_p is small. We therefore create two cases with the same stations and trains, but with different amount of passenger. In case IV we will use the average passenger figures, so $w_p = \overline{w}_p$ and in case V a possible realization of the passenger figures will be used.

$$\text{Where } w_p = \begin{cases} \lfloor \overline{w}_p \rfloor + 1 & \text{with probability } \overline{w}_p - \lfloor \overline{w}_p \rfloor \;\; (\text{fractional part of } \overline{w}_p) \\ \lfloor \overline{w}_p \rfloor & \text{otherwise} \end{cases}$$

5.5 Case VI

In the sixth case, some ‘small’ stations are added to the stations of the third case. In this case all intercity trains and regional trains (in both directions) between 13 ‘big’ stations and 78 ‘small’ stations are considered. Because of the enormous amount of OD-pairs we consider, just like in case V, a realization of the passenger figures.

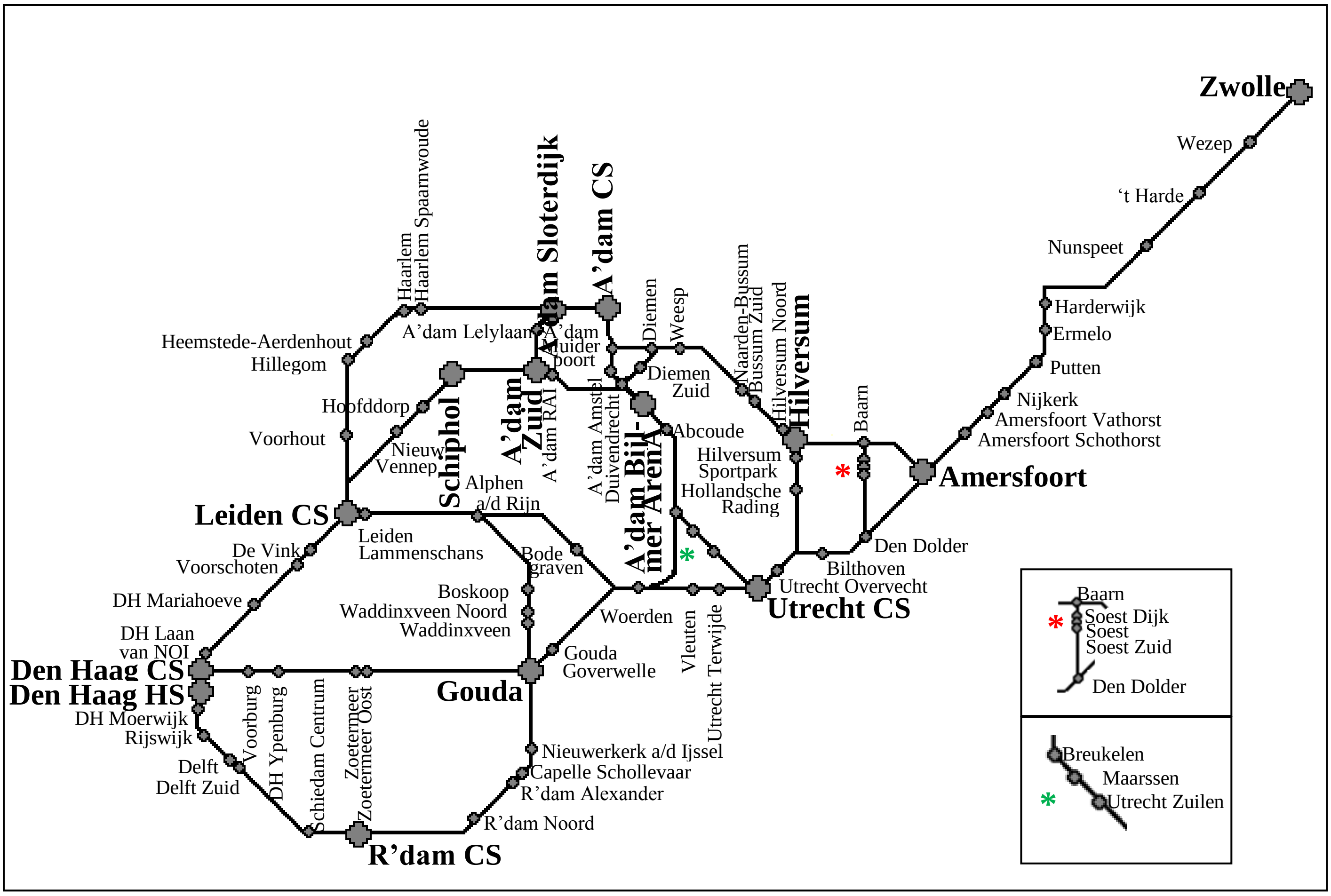

Figure 7 Graph representation of a part of the railway network for case VI

5.6 Overview of the cases

In Table 1, an overview of the six cases is given. The second column gives the number of OD-pairs that are included. Note that if it is possible to travel from one station to another at several times, these possibilities correspond to multiple OD-pairs. We have reported the total number of OD-pairs and between brackets the percentage of OD-pairs that need a transfer. The third column gives the total number of passengers that are considered; between brackets the percentage of passengers that need a transfer is given. The fourth column gives the number of trains and the percentage of intercity trains. Finally, the last column indicates the number of stations and the percentage of 'big' stations for each case.

Case	**# OD-pairs (% that need a transfer)**		**# Passengers (% that need a transfer)**		**# Trains (% intercity)**		**# Stations (% 'big')**	
I	111	*(15%)*	23	*(2.3%)*	7	*(71.4%)*	13	*(38.5%)*
II	355	*(36%)*	147	*(12%)*	117	*(100%)*	10	*(80.0%)*
III	914	*(55%)*	345	*(21%)*	168	*(100%)*	16	*(81.3%)*
IV	3940	*(65%)*	261	*(12%)*	284	*(54.3%)*	33	*(27.3%)*
V	908	*(28%)*	289	*(12%)*	284	*(54.3%)*	33	*(27.3%)*
VI	2875	*(39%)*	775	*(17%)*	404	*(50.9%)*	81	*(16.1%)*

Table 1: Overview of the cases: for each case, the number of OD-pairs, passengers, trains and stations are presented. Furthermore the percentage of OD-pairs that need a transfer, passengers that need a transfer, intercity trains and big stations are given.

In Table 2, an overview of the size of the event-activity network is given. The second column gives the number of departure events, which equals the number of arrival events. The third column gives the number of destination events, which equals the number of origin events and also the number of OD-pairs, because each OD-pair needs a destination and origin event. The fourth column gives the total number of events for each case. The fifth, sixth, seventh, eighth and ninth column respectively give the number of driving, dwelling, transferring, origin and destination activities. Finally, the last column indicates the total number of activities for each case.

Case	$\lvert E_{dep} \rvert$	$\lvert E_{dest} \rvert$	$\lvert \mathbf{E} \rvert$	$\lvert A_{drive} \rvert$	$\lvert A_{dwell} \rvert$	$\lvert A_{change} \rvert$	$\lvert A_{origin} \rvert$	$\lvert A_{dest} \rvert$	$\lvert \mathbf{A} \rvert$
I	28	111	**278**	28	21	10	180	146	**385**
II	219	355	**1148**	219	102	1074	2439	1164	**4998**
III	349	914	**2526**	349	181	1723	7975	3354	**13582**
IV	1022	3940	**9924**	1022	738	8068	41296	21574	**72698**
V	1022	908	**3860**	1022	738	8068	9300	4828	**23956**
VI	2053	2875	**9856**	2053	1649	13812	35376	15568	**68458**

Table 2: Overview of the size of the event-activity network: for each case the number of events and activities are given.

Comparing case III and VI, the number of trains has more than doubled, while the number of changing activities is seven times as large. This is caused by the fact that an regional train increases the number of changing activities a lot, because more 'small' stations can be reached now, wherefore passengers often have to change trains. We also observe that there are way more departure and arrival events, as the regional trains stop more often than intercity trains. Similar effects can be observed when cases II and IV are compared.

6 COMPUTATIONAL RESULTS

In this Chapter we will represent the computational results in the different cases for the four models we presented in Chapter 4. We used CPLEX 11.1 in AIMMS 3.10 on **an Intel® Core™ 2 CPU(@3.00 GHz) with 4GB of memory**. For every model and for every case we will show, in a table, the objective value, number of delayed events, percentage of delayed OD-pairs and percentage of delayed passengers. Note that each entry in the table is the average value over all delay scenarios.

	Objective value (s)				**Delayed Events**			
Model / Case	**(1)**	**(2)**	**(3)**	**(4)**	**(1)**	**(2)**	**(3)**	**(4)**
I	30,462	30,462	30,462	30,462	2.7	2.7	2.7	2.7
II - VI	-	-	-	-	-	-	-	-
	Delayed OD-pairs (%)				**Delayed Passengers (%)**			
Model / Case	**(1)**	**(2)**	**(3)**	**(4)**	**(1)**	**(2)**	**(3)**	**(4)**
I	11.4	11.4	11.4	11.4	9.0	9.0	9.0	9.0
II - VI	-	-	-	-	-	-	-	-
	Solving time (s)							
Model / Case	**(1)**	**(2)**	**(3)**	**(4)**				
I	0.23	0.25	2.28	2.42				
II - VI	-	-	-	-				

Table 3: Characteristics of the optimal solutions for four different models: the objective value, number of delays, percentage of delayed trips, percentage of delayed passengers and average solving times are given.

For the smallest real-world instance, case I, it turned out that that the optimal solution of the DMwRR-model does not violence the capacity constraints. Therefore all four models have the same outcomes. This can be explained by the fact that this case is really small: only 7 trains are subject to the capacity constraints.

Unfortunately it turned out that the larger instances (case II - VI) cannot be solved with the integer programming formulations. Therefore Table 3 shows no outcomes for these larger instances.

From Table 3 it follows that adding the “platform” constraints to the model does not significantly increase the solving time, however adding the “not overhaul” constraints to the model does increase the solving time significantly.

As mentioned before, the larger instances were unsolvable. For case II there were 5 (out of 100) scenarios that could be solved, but the solving times were very high. It took about 55 minutes (3300 seconds) to solve these. The other 95 out of 100 scenarios for the second case, make the used optimization software AIMMS, run out of memory. For all larger cases, case III – VI it also turned out that AIMMS runs out of memory because of the complexity of the problem.

7 SHORTCOMINGS

In this chapter we will describe some shortcomings and points of discussion of our models.

One assumption that can be doubted is “Intercity trains can only overhaul regional trains on their way to ‘big’ stations”. As told before, the used data can be represented as an undirected graph, where the arcs connect the nodes. For the “not overhaul” constraints, we take a look at the driving-arcs: these arcs represent trains driving from one station to another. To implement this constraint we state that trains cannot overhaul each other when they are driving on the same arc to a ‘small’ station. Two arcs are the same when they have the same start and end station. This can cause problems as the arcs of intercity trains do not include stations they pass by but do not stop at. We will show possible problems that can arise with two examples.

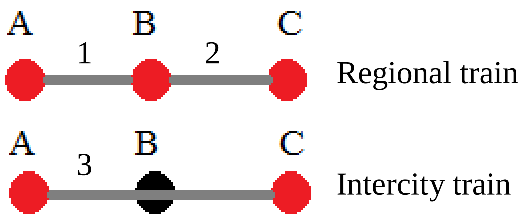

Figure 8 Example 1: A regional and an intercity train driving from A to C

Suppose there is one intercity train driving from station A to ‘small’ station C and passing by station B. And there is one regional train driving from station A to C, via station B. As can be seen in Figure 7 the graph for the regional train consists of two arcs: one from A to B and one from B to C. The graph for the intercity train from A to C is one arc: from A to C. Because these trains do not drive at the same arc (there is no arc in this figure with the same start and end station) the “not overhaul” constraint will not be taken into account for these two trains. Therefore the intercity train from A to C can still overhaul the regional train from A to C.

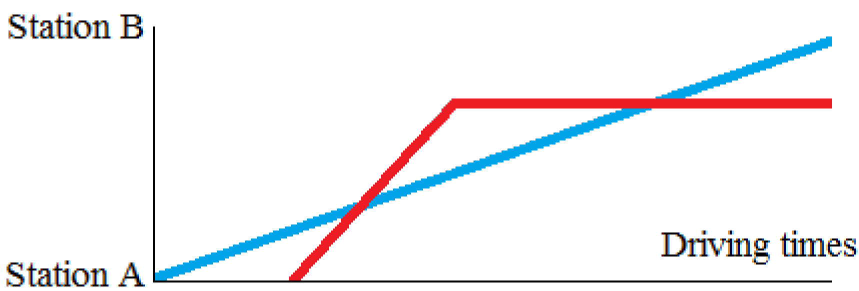

Figure 9 Example 2: The driving times for a regional and an intercity train driving from station A to C

In Figure 9 another problem is shown, to implement “trains cannot overhaul each other when they are driving on the same arc to a ‘small’ station”, we state that they do not overhaul if the train that departs first also arrives first at the next station. In Figure 9 can be seen that this does not make sure that the trains do not overhaul in between the stations.

Like in the original model, two trains driving on the same arc can still overhaul each other, if they are on their way to a ‘big’ station. When that is the situation, the trains can overhaul each other at any piece of the track between station A and ‘big’ station B. Of course in reality this is not possible, because trains can only overhaul each other at ‘big’ stations or when they are very close to the ‘big’ station at a sidetrack.

The described problems in this chapter all had to do with the fact that the current model has a macroscopic scale. To be able to implement the capacity constraints in such a way that the above described problems can be solved, the model has to be of microscopic scale. With microscopic scale we mean that not only stations (nodes) are in the model, but also platforms and tracks. For example when we divide an arc between two stations in a lot of small arcs, they constraint of “not overhaul” can be implemented in a better way.

Another point of improvement is the route. We grouped the train numbers in such a way that trains that drive more or less the same route share the same platform at all joint stations. WIt would be better to add a parameter which contains the platform where a train arrives at the station. Note that this parameter then would also have a microscopic scale.

8 CONCLUSION AND FURTHER RESEARCH

In this chapter we will summarize this thesis and derive some conclusions and possibilities for further research.

Because the original DMwRR-model does not take the limited capacity of the Dutch rail network into account, we considered two kinds of capacity constraints in this thesis. First we made a model with “platform” constraints, which ensure that trains cannot use the same platform at a station at the same time. Second we made a model with “not overhaul” constraints, these ensure that intercity trains can only overhaul between two stations if the next station is a station with more than four tracks. And the last model we made is a model with both “platform” and “not overhaul” constraints.

To evaluate the impact of the different capacity constraints, we have compared the performance of the DMwRR-model with the three models we made. Performance cannot be measured in terms of the objective function (total passengers’ delay), because adding capacity constraints can never result in lower delay. Actually when comparing the models in terms of the total passengers’ delay is like comparing different models with each other. We therefore chose to compare the models in terms of delayed events, delayed OD-pairs and delayed passengers.

For case I, we found out that the optimal solution is the same for all the four models. Even when the train from Zwolle to Amersfoort is delayed for 30 minutes, the capacity constraints were not violated in the DMwRR-model. This can be explained by the fact that this is a very small case: there are 7 trains driving 3 different routes. Because both the “platform” and “not overhaul” constraints only exist for trains driving the same route, it is likely that the capacity constraints are coincidentally taken into account.

Unfortunately it turned out that the larger instances (case II - VI) have really high solving times (over 3300 seconds) and are therefore unsolvable. The used optimization software AIMMS, runs out of memory when trying to solve these cases.

A first possibility for further research is improving and adding more capacity constraints, however because the problem is already complex, this will only lead to a more complex problem, which will not be solvable as well. A second possibility for further research is making heuristics to be able to also solve the larger instances for delay management with re-routing and capacity constraints.

References

T. Dollevoet, D. Huisman, M. Schmidt, A. Schöbel. Delay management with re-routing of passengers. *Econometric Institute Report EI 2010-31, 2010*.

K. Nachtigall. Periodic Network Optimizatin and Fixed Interval Timetables. *Deutsches Zentrum fűr Luf- und Raumfart, Institut fűr Flugfűhrung, Braunschweig, 1998, Habilitationsschrift*.

M. Schachtebeck, A. Schöbel. To wait or not to wait an who goes first? Delay management with priority decisions. *Transport Science, DOI 10.1287/trsc.1100.0318, 2010*.

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

A.Schöbel. Integer programming approaches for solving the delay management problem. *Lecture notes in computer science, 4359: 145 – 170, 2007*.

A.Schöbel. Capacity constraints in delay management. *Public transport. Planning and operations, 1(2): 135 – 154, 2009*.