MASTER’S THESIS

Minimizing Railway Rolling Stock
Influence of line planning on carriage-kilometers

by
H. Deckers

Supervisors:
Dr.ir. C.A.J. Hurkens
Prof.dr. L.G. Kroon
H. van Uden

Eindhoven, January 19, 2010
Preface

The thesis that you are about to read is the result of an eight-month internship at the Nederlandse Spoorwegen (NS) in Utrecht. This is also the conclusion of my master’s degree program Industrial and Applied Mathematics with specialization Statistics, Probability and Operations Research at Eindhoven University of Technology. Several people have supported and helped me during my studies and internship in particular therefore I would like to thank them.

Firstly, my thanks goes to my supervisors Cor Hurkens (TU/e), Leo Kroon (NS) and Hans van Uden (NS) for all their help and explanations, but also for the numerous discussions on the formulation of the problem and the interpretation of the results. I really enjoyed working with them. I would especially like to thank Leo and Hans for being my supervisors at NS and allowing me to do this internship.

Secondly, I am grateful to Gerard Woeginger and Ivo Adan for completing the assessment committee.

Thirdly, the work environment was inspiring thanks to my colleagues at NS Reizigers/Logistiek/Innovatie. They all have an exceptional knowledge of really everything that has to do with trains. I have learned a lot about NS simply by listening to them. They were always willing to help me if I had a problem whether it concerned getting access to data, working with unfamiliar software or explaining mathematical subjects.

Fourthly, I would also like to express my admiration for my family and boyfriend. They gave me unconditional support throughout my studies and encouraged me to persevere when I was struggling. Last but not least, I would like to thank my friends who showed an interest in my studies.
Abstract

The Nederlandse spoorwegen (NS) is the largest railway operator in the Netherlands. Among other services, it provides passenger transportation. Basically, NS wants to transport as many passengers as possible at low costs. One of the costs are the carriage-kilometers, i.e. the sum of the mileages of all the carriages. In this thesis, we look for ways to minimize the carriage-kilometers. Hereby, we are allowed to neglect constraints that have been developed over time due to tradition, government and even infrastructure. Actually, we want to know the minimum carriage-kilometers to get the passengers from origin to destination. To keep the solution at least a bit realistic, we define some restriction that we will not neglect even though it was allowed.

This thesis contains three different approaches to minimize the carriage-kilometers. Firstly, we will develop the combination model. This model includes the option to cut train lines into multiple parts. Therefore, we are able to adjust the lengths of the trains better to the number of passengers on the route. The results, based on our examples, are however disappointing. We save too little carriage-kilometers to make up for the extra work that comes along with the cuts.

Secondly, we will construct the cancellation model. Cutting train lines like before is not allowed now but deleting train rides is. The passengers that are supposed to take a deleted train are divided over the train ride before and the train ride after the deleted train. The results of our cases are rather positive concerning the carriage-kilometers but the question remains if it is a good thing to start canceling train rides or not.

Thirdly, we will generate the line model. This model selects a new set of train lines such that all the passengers can get from their origin to their destination and the carriage-kilometers, needed to do so, are minimal. We tested this model on one small case. The results indicate some differences with the current line planning.

To conclude this thesis, we will give some extensions for the three models. Especially the line model leaves room for improvements.
# Contents

**Preface** ............................................ i

**Abstract** ...................................... iii

1 Introduction ..................................... 1
   1.1 Rolling Stock Planning .......................... 2
   1.2 Problem Description and Approach .......... 3
   1.3 Related Literature ............................. 4
   1.4 Structure of the Thesis ..................... 5

2 Combination Model ............................... 7
   2.1 Planning Process .............................. 7
   2.2 Objective .................................... 7
   2.3 Constraints .................................. 8
   2.4 Variations on the Model .................... 11
   2.5 Solutions ..................................... 12
      2.5.1 Data and Settings ..................... 12
      2.5.2 PC and Software ........................ 14
      2.5.3 Results and Conclusions ............. 15

3 Cancellation Model .............................. 19
   3.1 Objective .................................... 19
   3.2 Constraints .................................. 20
   3.3 Cases ........................................ 25
      3.3.1 Data and Settings ..................... 25
      3.3.2 PC and Software ........................ 26
      3.3.3 Case I and II ........................... 26
      3.3.4 Case III - V ......................... 27
   3.4 Conclusions .................................. 28

4 Line Model ...................................... 29
   4.1 Objective .................................... 29
   4.2 Column Generation ............................ 30
   4.3 Constraints .................................. 31
      4.3.1 Restricted Master Problem ............ 31
      4.3.2 Subproblem .............................. 33
   4.4 Solutions ..................................... 37
      4.4.1 Data and Settings ..................... 37
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.2 PC and Software</td>
<td>39</td>
</tr>
<tr>
<td>4.4.3 Results and Conclusions</td>
<td>40</td>
</tr>
<tr>
<td><strong>5 Extensions</strong></td>
<td>43</td>
</tr>
<tr>
<td>5.1 Combination Model</td>
<td>43</td>
</tr>
<tr>
<td>5.2 Cancellation Model</td>
<td>43</td>
</tr>
<tr>
<td>5.3 Line Model</td>
<td>44</td>
</tr>
<tr>
<td><strong>A Extended Combination Model</strong></td>
<td>45</td>
</tr>
<tr>
<td><strong>B Schedules</strong></td>
<td>53</td>
</tr>
<tr>
<td>B.1 Current Schedules</td>
<td>54</td>
</tr>
<tr>
<td>B.2 Schedules of the Combination Model</td>
<td>56</td>
</tr>
<tr>
<td>B.3 Schedules of the Cancellation Model</td>
<td>68</td>
</tr>
<tr>
<td><strong>Bibliography</strong></td>
<td>77</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Inventions like automobiles and airplanes have affected the usage of railway transportation. In addition, the national railway market was opened by a directive of the European Union in 1995. As a result, there is more competition, so railway companies are forced to compete harder for the customers. Raising their service level and cutting their costs are measures to attract more customers. Railway companies are therefore optimizing their planning process since it influences both the service level and the costs.

Nowadays, the Nederlandse spoorwegen (NS) is the largest railway operator in the Netherlands. The Dutch state and NS signed a contract which gives NS the exclusive right to use the core of the railway system for passenger transportation until 2015. A premium is even granted if NS is able to increase its share of the morning peak traffic. On the other hand, NS has to pay a certain amount if the punctuality does not reach a particular percentage. Moreover raising the ticket prices depends on the punctuality. This punctuality is the percentage of trains that have at most 3 minutes delay on arrival at one of the larger stations.

NS can use the main part of the infrastructure but they do not control it. ProRail is responsible for infrastructure management. NS needs the approval of ProRail to apply a timetable. The reason for this is that NS is not the only user of the rail network. Passenger transportation is provided mainly by NS but also by Syntus, NoordNed and Connexxion. On the other hand, freight traffic also uses the Dutch railway network. Some operators are Railion, ACTS, Rail4Chem and ERS. The precise planning of rolling stock and crew is left to the operators.

The NS consists of three subdivisions, see Figure 1 for an overview. First, Reizigersvervoer contains NS Reizigers (NSR), NS Hispeed, NedRailways and NedTrain. NSR tries to realize qualitative transport capacity, good service and information, appealing travel products and a high standing safety profile. NS Hispeed is the high-speed company of NS and will be the
Chapter 1. Introduction

Dutch umbrella brand for high-speed trains such as Thalys, ICE International, TGV, Eurostar and the new high-speed train on the HSL-Zuid. NedRailways is responsible for the recruitment and exploitation of rail licenses outside the Netherlands. NedTrain gives maintenance, advice and service to companies that offer public transport by rails. Second, Knooppuntontwikkeling contains NS Poort which handles the exploitation and development of real estate and does retail. Third, Railinfra & Bouw contains Strukton which is specialized in developing rail infrastructure and information systems. In this thesis, we will concentrate on the use of rolling stock.

1.1 Rolling Stock Planning

Within NS, there are two subparts that are involved in rolling stock planning. NS Reizigers (NSR) plans the timetable and schedules the rolling stock and crew. It is also responsible for carrying out these plans. The quality of the rolling stock is checked and improved by NedTrain.

The rolling stock consists of electrical and diesel units. Each unit has a number of carriages, an engine and driver’s cabins on both ends. The only option to enlarge the capacity of a train is to add a unit. A single carriage can not be added. NS uses units of different types and lengths. A unit of a certain type can contain different numbers of carriages. For example, a koploper-type unit can consist of three or four carriages.

![Figure 1.2: Example of a train consisting of 2 units(VIRM4-type) and 8 carriages.](image)

There are some restrictions to the composition of trains. Units need to be of a compatible type in order to be able to combine them. So, a train consists of a compatible selection of types of units. It can contain multiple units and these units may or may not have the same number of carriages. Also, not every train length is allowed. This depends on safety regulations and the lengths of the platforms at which the train needs to stop.

Non-timetabled train movements inside the stations are called shunting. This contains the routing of trains through the station and the positioning of temporarily unused units to the shunting yards. At some stations, there is a possibility of changing the composition of the trains. Normally one can either couple or uncouple units. Doing both is not granted. It is even not allowed to couple or uncouple more than two units. When a train consists of different kinds of units, one should consider the order of these units. For example, if a train arrives at a station and it is not going to leave the station in the direction it came from, then it is not possible to uncouple units at the front of the train. If a train leaves the station in the direction it came from, then it is not possible to uncouple at the back of the train. The reason is that the uncoupled units will be blocking the exit route of the leaving train. This is just one example of these kind of practical restrictions.

NS operates two types of train lines in the Netherlands. Regional trains stop at every station along their way and inter-city trains only stop at the major stations. Train types are designed especially for each of these types of train lines. A train line is a series of trains that directly connect given stations. All train lines have their own number, for example, the 3000 line is the inter-city train line connecting Nijmegen (Nm) to Den Helder (Hdr). More specifically, each ride along a train line has its own number, for example, the first ride from Den Helder to Nijmegen has number 3019. This number is always larger than the number of the train line and smaller than the number of the train line plus 100. If the train travels in southern direction, it has an
odd number otherwise it is even.

Finally, we need to mention the maintenance. When making a planning, we need to make sure that every unit gets a check-up in time, in order to maintain the quality of the rolling stock. This reduces the number of unexpected defects on train units and makes them more reliable.

In the next section, we will explain what the point of interest is concerning the rolling stock.

1.2 Problem Description and Approach

To remain competitive, NS has to optimize its timetables, rolling stock schedules and crew schedules constantly. Usually NS only adjusts the existing timetable slightly. A reason for this is that not all passengers will consider the change to be an improvement. Naturally NS wants to avoid passengers deciding to travel by car instead of by train. If the change is small, most people will accept it and continue to use the services of NS. Moreover, the Dutch government restricts NS' ability to adapt the timetable. NS is not allowed to make just any change. Besides, it takes less development time to adjust an existing timetable than to start all over and make a completely new table.

The problem that we are looking into concerns the line planning and we will focus mainly on the rolling stock planning. Eventually, we want to reorganize the usage of equipment such that the capacity of the trains better matches the expected number of customers. So, without having the intention of actually implementing these changes, NS is interested in finding some lower bound on the rolling stock needed to maintain the same service capacities. In order to achieve this goal, we are allowed to soften some constraints that have been hard constraints in the past. At this moment, trains stay almost the complete ride in the same composition even when the expected number of passengers varies a lot over the ride. More specifically, in some cases it is even possible to use a smaller train for some parts of the ride. The issue is that coupling and uncoupling options are limited and undesirable during rush hours. The potential delay is too large.

We want to find out if it would be cheaper, for NS, to work with shorter train lines. In most cases, the NS never tried to cut these train lines into smaller parts. As a result, it would be possible to deploy more appropriate compositions of train units. The major reason why NS does not apply this strategy is that passengers would have to change trains more often. What would happen if we ignore this fact? We also investigate if canceling rides would have a positive effect. For example, during the rush hours, people need to take a certain train to be in time for their job. Outside the rush hours, mainly people going on a recreational trip take the train. Since they are not bound by working hours, it is easier for them to take an earlier or later train. So, if it is possible to distribute the passengers of a certain train over the train before and the train after, then it is possible to cancel that train. We even search for a completely new line plan that could save money on rolling stock but still get all the passengers to their destination.

There is more than one way to adjust the line plan to influence the rolling stock plan. For the first approach, we consider the current timetable and rolling stock plan. The intention is to use the current train lines and cut them into smaller lines. The frequencies of the trains stay the same, as does the rest of the timetable. This means, arrival times and departure times will stay the same, but we are allowed to soften some constraints that have been hard constraints in the past. The rolling stock plan will change completely since a train unit that was assigned to a certain train line can get reassigned to only a part of that line.

In this part, the objective criteria are minimizing the carriage-kilometers and the amount of seat shortages. Carriage-kilometers can be reduced by using more appropriate train composi-
tions. Seat shortage is one of the factors that determine the service quality. We leave all other factors out of the model because we want to find a schedule where the train compositions have minimum capacity. For instance, we do not include the continuity requirement which states that at least one unit should follow the complete route of that service. This means that passengers who want to travel the complete route do not have to change seats. This gives us the opportunity to cut train lines into smaller lines.

First, we do not take order restrictions into account. So, at each station the composition of the train can be changed into any other order. We do respect the fact that we can only change the composition by two units. Also coupling and uncoupling at the same station is not allowed. Later, they can be added to the model.

Second, we neglect all shunting activities and we assume that stations have unlimited storage capacity and platforms. We only include some extra time after a unit has been uncoupled. This time should be enough to do all shunting activities that are necessary. This means that a unit can not be used for a trip during this period.

Third, we will not consider constraints for certain train types to be used for certain train lines. There will be no difference between regional, inter-regional and inter-city material. Any type of equipment can be used. Later this may also be adjusted in a more realistic, suitable constraint.

At first, we will only consider the 3000 line, which is the inter-city train line connecting Nijmegen (Nm) to Den Helder (Hdr). This line is suitable to start with since it contains parts where the train capacity is optimally used but also parts where there seems to be room for improvement. This reasoning is based more on experience than on pure knowledge. A second line is the 800 line. This is a combined train line since it connects Schagen (Sgn) not only with Maastricht (Mt) but also with Heerlen (Hrl) by splitting in Sittard (Std). When we have enough feeling for and insight into the problem, the goal is to consider the entire national train network of NS.

The second approach is canceling train rides. The simplifications mentioned above will also be used this time. Now, we are not cutting the train lines but canceling the train rides. We only cancel a train ride if its passengers can be distributed among the previous train and the next train. It is allowed to enlarge the previous train and the next train to be able to transport all the passengers. But the carriage-kilometers need to be less than before, otherwise it was not wise to cancel the train ride. For the tests, we use the same train lines as for the first approach.

The third approach is developing a better line plan. This means that the resulting line plan consists of train lines that are better adjusted to the number of passengers using a certain part of the rail network. Order restrictions and shunting activities are not taken into account. Minimizing only the carriage-kilometers is also in this case not enough. Moreover, we want to maximize the number of travelers that do not have to change trains during their journey.

We test this approach first on the most southern part of the Netherlands, this means every station and piece of rail from Roermond (Rm) downwards except for the parts used by other passenger transporters.

1.3 Related Literature

This thesis is related to literature concerning four main topics, namely mixed integer optimization problems, shortest path algorithms, column generation and practical railway problems. All our approaches relate to the first topic, only the last approach relates to the second and the third topic.
First of all, mixed integer optimization problems are well-studied. The books by Schrijver \cite{17} and Nemhauser and Wolsey \cite{14} are standard works. An overview of the development on this topic can be found in the article by Darby-Dowman and Wilson \cite{8}.

Second, shortest path algorithms have been developed extensively. Although, we only use a basic algorithm, faster algorithms can be found in \cite{10}, \cite{16}, \cite{12}. Some examples of shortest path algorithms are the A* algorithm, naive label correcting methods and bidirectional versions.

Third, we use column generation for the last approach. The book by Desaulniers, Desrosiers and Solomon \cite{9} contains basic information about column generation but is also an overview of the state-of-the-art in integer programming column generation and its many applications. Other papers are \cite{19}, \cite{18} and \cite{1}. The first paper is a review on the topic, while the other two include new insights on the matter.

Finally, there are related practical railway problems that were discussed earlier. The thesis by Lindner and Zimmermann \cite{20} analyses the Periodic Event Scheduling Problem (PESP) and a minimum cost scheduling problem (MCSP). The paper by Caprara, Fischetti and Toth \cite{3} handles the train timetabling problem of a single, one-way track linking two major stations, with a number of intermediate stations in between. Carey and Lockwood write in their paper \cite{6} about a train pathing and timetabling problem for rail lines having separate tracks for trains in each direction. On the other hand, Caprara et al. discuss in \cite{4} a variety of railway problems like line planning, train timetabling, train platforming and rolling stock circulation. Carey and Carville concentrate in their paper \cite{5} on the performance and reliability for trains. The thesis by Maróti \cite{13} focuses on all kinds of planning problems that arise at the major Dutch passenger railway operator NS. Some papers, like \cite{11}, \cite{2} and \cite{7}, only consider the line planning problem of the last approach.

\section{1.4 Structure of the Thesis}

In this section, we introduced the company NS and the problem. We also initiated three approaches of the problem.

In the continuation of this thesis, we will develop optimization models for all the approaches. We will study the content of these models and test them in multiple ways. Eventually, we are able to state the restrictions of these models. We will also analyze the results of the various test cases. These models are called combination model, cancellation model and line model.

In the end, we give some research questions that might be interesting to take a closer look at, in the future. Moreover, some options are formulated that might make the results more precise and realistic. These questions and options are considered extensions.
Chapter 2

Combination Model

In this thesis, we will consider three approaches to minimize the number of carriages-kilometers (CKM). The first approach starts with the current timetable and allows train lines to be cut at every desired station. This strategy will be explained in this chapter. First, the model is defined by setting the objective and all the constraints. Second, we will explain which data and settings are used and we will give an overview of the results. In the last part we will also give some conclusions.

2.1 Planning Process

The railway planning process operates on four levels: strategic, tactical, operational and short-term planning. The first steps towards a complete planning are taken a few years in advance. This is then called the strategic planning. Tactical planning is done two months up to a year in advance while operational planning is done three days up to two months before the actual use of the schedule. Finally, the short-term planning fills in all the remaining details only three days, or less, before the day of execution.

Our problem is a part of the tactical planning since our model will determine which type of material should make the trip but not which exact unit it should be. This means that if, some short time before the trip, the scheduled unit breaks down, our schedule is still applicable. One just needs to find another unit of the same type that is not already scheduled for another trip. Our schedule will be even more general at some points since we do not take the order of the train units, shunting activities and train types into account. We also assume that the shunting yard is always large enough and there are always enough platforms in the stations. The model that we use is a derivative of the composition model described by Maróti in [13].

2.2 Objective

As was mentioned before, the goal is to minimize the number of carriage-kilometers. These carriage-kilometers are an important cost for the NS. Cutting carriage-kilometers is not hard since we can just make every train shorter and save money on the carriage-kilometers. The result is less space inside the trains, so people have no seat or worse do not fit in the train at all. These people will not travel by train in the future and the NS will lose profit. Consequently, we need to maintain at least the same service levels when cutting carriage-kilometers. Therefore, also the seat-shortage-kilometers (SKM) are included in the objective function. The seat-shortage-kilometers are calculated by adding up every kilometer that a person has no seat. The third part
of the objective function is related to the possibility to cut train lines. If the model calculates that it is best to continue, after arrival at a station, with the same combination of units then there is no need to cut the train line at this station. The last component of the objective function is a penalty if there still is a cut in this station.

Before we can actually state the objective function, we need to define some concepts and sets. First of all, a trip is a sequence of train movements that need to be carried out without combination changes. If \( t_1 \) is a trip from station \( a \) to station \( b \) and \( t_2 \) is a trip from station \( b \) to station \( c \) and \( t_1 \) is followed by \( t_2 \), then \( t_2 \) is the successor trip of \( t_1 \) and \( t_1 \) is the predecessor trip of \( t_2 \). A combination is a set of train units that can be used for a trip. \( T \) is the set of all trips, while \( T_0 \) is the set of trips with no predecessor trips and \( T_\infty \) is the set of trips with no successor trips. \( M \) is the set of all rolling stock types and \( B_t \) is the set of all combinations that are allowed for trip \( t \). The objective function eventually is stated as,

\[
\begin{align*}
&w_1 \sum_{t \in T} \sum_{m \in M} d_t c_m N_{t,m} + w_2 \sum_{t \in T} \sum_{b \in B_t} d_t s_{t,b} X_{t,b} + w_3 \sum_{t \in T \setminus T_\infty} D_t,
\end{align*}
\]

where \( d_t \) is the length of trip \( t \), \( c_m \) is the number of carriages in a unit of type \( m \) and the variable \( N_{t,m} \) is the number of units of type \( m \) used on trip \( t \). The constant \( s_{t,b} \) is the number of people that have no seat if combination \( b \) is used on trip \( t \) and the decision variable \( X_{t,b} \) indicates if combination \( b \) is used on trip \( t \). The decision variable \( D_t \) indicates if there is a useless cut after trip \( t \) at its station of arrival. We define the vector \( w = (w_1, w_2, w_3) \) where the constants \( w_1, w_2, \) and \( w_3 \) are the weight factors of respectively the CKM, the SKM and the sum of useless cuts. They reflect the relative importance of the elements of the objective function. We are looking for a schedule of minimum cost.

### 2.3 Constraints

All train lines can be divided into two types. The simple train lines, like the 3000 line, follow a simple path from start to finish, while the combined train lines follow a path that splits and combines at some stations. An example of such a train line is the 800 line from Alkmaar (Amr) to Heerlen (Hrl) and Maastricht (Mt). Each train coming from Amr is split in Sittard (Std). One part goes to Hrl and the other part goes to Mt. Each train coming from Hrl or Mt is combined in Std and goes as one train to Amr.

The constraints defined in this section only apply to simple train lines. The extended combination model for simple and combined train lines can be found in Appendix A since it has a more complicated notation and does not give new insights. Still, all calculations discussed later are done with the extended combination model. The reason for this is to be able to compare all the results with each other.

Before we can state all the constraints, we need some more notations. The sets \( T \), \( T_0 \), \( T_\infty \), \( M \) and \( B_t \) were defined earlier. Other sets are \( S \), for the set of all stations, and \( C \), for the set of all service classes. A list of all constants that we used in the model is given in Table 2.1. By re-allocation time, we mean the time needed to take a unit to the shunting yard. After a trip, units can be sent to the shunting yard because they are not needed for some time. It takes as long as the re-allocation time to get there. During this time, the unit can not be used for any kind of trip. After this time, the unit can be used again immediately. This means there is no waiting time for units to get back from the shunting yards.

The main decision variables are listed in Table 2.2.
2.3. Constraints

\[\sigma(t) : \text{successor of trip } t,\]
\[\nu(b)_m : \text{number of units of type } m \text{ in combination } b,\]
\[s_d(t) : \text{departure station of trip } t,\]
\[s_a(t) : \text{arrival station of trip } t,\]
\[\tau_d(t) : \text{departure time of trip } t,\]
\[\tau_a(t) : \text{arrival time of trip } t,\]
\[\rho(t) : \text{re-allocation time after trip } t,\]
\[n_m : \text{number of available units of type } m,\]
\[\kappa_{m,c} : \text{number of seats of class } c \text{ in a unit of type } m,\]
\[\mu_{t}^{\text{max}} : \text{maximum number of carriages, in total, trip } t \text{ can receive},\]
\[\mu_{t}^{\text{min}} : \text{minimum number of carriages, in total, trip } t \text{ must receive},\]
\[\delta_t : \text{passenger demand per service class},\]
\[\beta_{b_1,b_2} : \text{indicates if the change of combination } b_1 \text{ to } b_2 \text{ is an allowed change.}\]

Table 2.1: A list of constants used in the combination model.

\[
\forall t \in T, b \in B_t:
X_{t,b} = \begin{cases} 
1 & \text{if combination } b \text{ is used for trip } t, \\
0 & \text{otherwise},
\end{cases}
\]

\[
K_t = \begin{cases} 
1 & \text{if there is no cut after trip } t, \\
0 & \text{otherwise},
\end{cases}
\]

\[
\forall t \in T \setminus T_s, b_1 \in B_t, b_2 \in B_{\sigma(t)}:
Y_{t,b_1,b_2} = \begin{cases} 
1 & \text{if combination } b_1 \text{ is used for trip } t \text{ and } b_2 \text{ for } \sigma(t), \\
0 & \text{otherwise}.
\end{cases}
\]

Table 2.2: A list of decision variables used in the combination model.

The variable \(N_{t,m}\) was explained above, but we also need others. The variable \(C_{t,m}\) stands for the number of units of type \(m\) that are to be coupled to the train right before trip \(t\) and \(U_{t,m}\) stands for the number of units of type \(m\) that are to be uncoupled right after trip \(t\). Finally, there are three variables related with the inventory. The inventory of type \(m\) at station \(s_d(t)\) immediately after the departure of trip \(t\) is indicated by \(I_{t,m}\). The variables \(I_{0,s,m}\) and \(I_{\infty,s,m}\) denote the number of units of type \(m\) stored at station \(s\) at the beginning and the end of the day.

The basic combination model is the following.

Minimize

\[
\begin{align*}
& w_1 \sum_{t \in T} \sum_{m \in M} d_{t,m}N_{t,m} + w_2 \sum_{t \in T} \sum_{b \in B_t} d_{t}s_{t,b}X_{t,b} + w_3 \sum_{t \in T \setminus T_s} D_t,
\end{align*}
\]  

subject to

\[
\forall t \in T \setminus T_s, b \in B_t: 
D_t \geq (1 - K_t) + X_{t,b} + X_{\sigma(t),b} - 2, \tag{2.3.2}
\]

Cutting train lines

\[
\forall t \in T: 
\sum_{b \in B_t} X_{t,b} = 1, \tag{2.3.3}
\]
∀ \in T \setminus T_\infty:
\begin{align*}
K_t &= \sum_{b_1 \in B_t, b_2 \in B_{\sigma(t)}} Y_{t,b_1,b_2}, \\
X_{t,b_1} &\geq \sum_{b_2 \in B_{\sigma(t)}} Y_{t,b_1,b_2}, \\
X_{\sigma(t),b_2} &\geq \sum_{b_1 \in B_t} Y_{t,b_1,b_2},
\end{align*}
(2.3.4)

∀ \in T \setminus T_\infty, b_1 \in B_t:
\begin{align*}
X_{t,b_1} &\geq \sum_{b_2 \in B_{\sigma(t)}} Y_{t,b_1,b_2}, \\
X_{\sigma(t),b_2} &\geq \sum_{b_1 \in B_t} Y_{t,b_1,b_2},
\end{align*}
(2.3.5)

∀ \in T \setminus T_\infty, b_2 \in B_{\sigma(t)}:
\begin{align*}
X_{\sigma(t),b_2} &\geq \sum_{b_1 \in B_t} Y_{t,b_1,b_2},
\end{align*}
(2.3.6)

Allowed combinations
∀ \in T \setminus T_\infty, b_1 \in B_t, b_2 \in B_{\sigma(t)}:
\begin{align*}
Y_{t,b_1,b_2} &\leq \beta_{b_1,b_2},
\end{align*}
(2.3.7)

(Un)coupling
∀ \in T, m \in M:
\begin{align*}
N_{t,m} &= \sum_{b \in B_t} \nu(b)_m X_{t,b}, \\
C_{t,m} &= N_{t,m}, \\
U_{t,m} &= N_{t,m}, \\
C_{\sigma(t),m} &= \sum_{b_1 \in B_t, b_2 \in B_{\sigma(t)}} \nu(b)_2 \nu(b)_1 \left( X_{\sigma(t),b_2} - \sum_{b_1 \in B_t} Y_{t,b_1,b_2} \right),
\end{align*}
(2.3.8)

∀ \in T \setminus T_\infty, m \in M:
\begin{align*}
U_{t,m} &= \sum_{b_1 \in B_t, b_2 \in B_{\sigma(t)}} \nu(b)_1 \nu(b)_2 \left( X_{t,b_1} - \sum_{b_2 \in B_{\sigma(t)}} Y_{t,b_1,b_2} \right),
\end{align*}
(2.3.9)

Inventory
∀ \in T, m \in M:
\begin{align*}
I_{t,m} &= I^0_{s_{d(t)},m} - \sum_{s_{d(t_2)}=s_{d(t_1)}} C_{t_2,m} + \sum_{s_{d(t_2)}=s_{d(t_1)}} U_{t_2,m,}, \\
I^\infty_{s,m} &= I^0_{s,m} - \sum_{s_{d(t)}=s} C_{t,m} + \sum_{s_{d(t)}=s} U_{t,m}, \\
n_m &= \sum_{s \in S} I^0_{s,m},
\end{align*}
(2.3.13)
∀s∈S, m∈M:
\[ I_{s,m}^0 = I_{s,m}^\infty, \]  
(2.3.16)

Variable domains

∀t∈T, b∈B:
\[ X_{t,b} \in \{0, 1\}, \]  
(2.3.17)

∀t∈T\setminus T_\infty, b_1\in B_i, b_2\in B_{\sigma(t)}:
\[ Y_{t,b_1,b_2} \in \{0, 1\}, \]  
(2.3.18)

∀t∈T:
\[ K_t \in \{0, 1\}, \]  
(2.3.19)

∀t∈T, m∈M:
\[ N_{t,m}, C_{t,m}, U_{t,m}, I_{t,m} \in \mathbb{R}_+, \]  
(2.3.20)

∀s∈S, m∈M:
\[ I_{s,m}^0, I_{s,m}^\infty \in \mathbb{R}_+, \]  
(2.3.21)

∀t∈T\setminus T_\infty:
\[ D_t \in \mathbb{R}_+. \]  
(2.3.22)

Constraints (2.3.2) state that \( D_t \) equals one only if the cut after trip \( t \) is not necessary. This is if trip \( t \) and its successor, trip \( \sigma(t) \), use the same combination of units and there is still a cut between the trips.

Constraints (2.3.3) guarantee that, for each trip, exactly one allowed combination is used. Constraints (2.3.4) state that if there is no cut after trip \( t \) then exactly one allowed combination is used for trip \( t \) and exactly one is used for the successor of trip \( t \). Constraints (2.3.5) and (2.3.6) make sure that if there is no cut after trip \( t \), then the decision variables indicate the same combinations for trip \( t \) and trip \( \sigma(t) \). If there is a cut after trip \( t \) then there are no further restrictions on the decision variables \( X_{t,b_1} \) and \( X_{\sigma(t),b_2} \).

Constraints (2.3.7) guarantee that consecutive combinations are allowed. This means that there are not more than two units added or removed from the initial combination. It is also not allowed to add and remove units at the same time.

Constraints (2.3.8) link the combinations of a certain trip to the number of units of each type used on this trip. Constraints (2.3.9) to (2.3.12) specify the number of coupled and uncoupled units.

Constraints (2.3.13) define the inventory after departure of each trip at the station of departure while constraints (2.3.14) set the inventory of each station at the end of the day. Constraints (2.3.15) make sure that the initial inventories add up to the number of available units. Constraints (2.3.16) state that the inventory at the beginning of the day is equal to the inventory at the end of the day. Finally, constraints (2.3.17) to (2.3.22) define the variable domains.

The model stated above is our basic version but other constraints can be added that might be useful in certain cases.

2.4 Variations on the Model

Throughout this thesis, we use multiple versions of the basic combination model to serve several purposes. These purposes are explained later but the additions to the combination model are given in this section.
The first addition replaces constraints (2.3.15) and (2.3.16). These constraints state that at the beginning and the end of each day the inventory should be the same at each station. The new constraints allow different inventories at the beginning and end of the day but they are constant. So, at each station we can indicate which inventory is wanted to start the day and indicate which inventory is wanted at the end of the day. The constraints are:

\[ \forall s \in S, m \in M : \quad I^0_{s,m} = i^0_{s,m}, \quad (2.4.1) \]
\[ \forall s \in S, m \in M : \quad I^\infty_{s,m} = i^\infty_{s,m}, \quad (2.4.2) \]

where \( i^0_{s,m} \) is the desired inventory of type \( m \) at the beginning of the day at station \( s \) and \( i^\infty_{s,m} \) is the desired inventory of type \( m \) at the end of the day at station \( s \).

For the second addition, we need to define a new set of stations. We call this set \( S_0 \) and it contains stations where we do not want to couple or uncouple units and cutting is not allowed. In this case, the constraints that need to be added are:

\[ \forall t \in T \setminus T_\infty, m \in M, (s \in S_0: s_d(t) = s) : \quad C_{t,m} = 0, \quad (2.4.3) \]
\[ \forall t \in T \setminus T_\infty, m \in M, (s \in S_0: s_a(t) = s) : \quad U_{t,m} = 0, \quad (2.4.4) \]
\[ \forall t \in T \setminus T_\infty, (s \in S_0: s_a(t) = s) : \quad K_t = 1. \quad (2.4.5) \]

The last addition concerns the rush hours. Usually rush hours are assumed to be between 7.00 and 9.00 a.m. and between 16.00 and 19.00 p.m.. If we would not allow cutting, coupling or uncoupling during these hours then the constraints would be,

\[ \forall m \in M, (t \in T \setminus T_0 : 700 \leq \tau_d(t) < 900 \lor 1600 \leq \tau_d(t) < 1900) : \quad C_{t,m} = 0, \quad (2.4.6) \]
\[ \forall m \in M, (t \in T \setminus T_\infty : 700 \leq \tau_a(t) < 900 \lor 1600 \leq \tau_a(t) < 1900) : \quad U_{t,m} = 0, \quad (2.4.7) \]
\[ \forall (t \in T \setminus T_\infty : 700 \leq \tau_a(t) < 900 \lor 1600 \leq \tau_a(t) < 1900) : \quad K_t = 1. \quad (2.4.8) \]

These additions are only three examples. Many more combinations of restrictions might be desirable for the model at some point.

2.5 Solutions

In this section, we will specify which data we use and explain why we work precisely with this selection of train lines. We will also state which general settings we use although these are not the actual settings used by NS. Eventually, we will give an overview of some relevant results. In the end, we will also formulate our conclusion with regard to these results.

2.5.1 Data and Settings

In Section 2.3, we considered the basic combination model applicable for simple train lines. However, for all our calculations, we used the extended combination model stated in Appendix A. The only difference between both models is that the extended combination model can handle combined train lines. It handles simple train lines in exactly the same way as the basic combination model. For a combined train line, we decide that it can be cut in multiple ways. If the train line has two legs, it is allowed to cut off both legs such that both parts become autonomous but it is also allowed to cut off just one leg such that only that leg becomes autonomous and the other one still depends on the previous part of the train line.
2.5. Solutions

We apply the extended combination model to the simple 3000 line and the combined 800 line for a generic Monday. In Table 2.3 we show the two lines together and give a list of the standard abbreviations of the names of the stations.

These train lines share the part between Amr and Ut. Currently, NS tries to define for each day a set of train units for each train line or group of train lines for exclusive use. This improves the robustness of the schedule. This means that if a train is delayed, it affects only that single train line or group of train lines instead of the whole network.

In the data, that we are using to compare our results with, we found that 10 units out of 65 units are used for both the 3000 line and the 800 line. The current types of materials used for these train lines are the VIRM4, also called AD, and the VIRM6, also called OA. An AD consists of four carriages and has 63 first class seats and 342 second class seats. An OA consists of six units and has 132 first class seats and 464 second class seats. The 3000 line uses 18 ADs and 17 OAs. The 800 line uses 26 ADs and 14 OAs. Together they use 36 ADs and 29 OAs. So, 8 ADs and 2 OAs are being used by both train lines. All the numbers of the current case are taken from the timetable and rolling stock schedule of February 2nd of 2009. Figure B.1 shows the current schedule for the 800 line while Figure B.2 shows the current schedule for the 3000 line. These schedules have a strict pattern and the combinations rarely change during a train ride in comparison with the results of the combination model.

The extended combination model does not try to minimize the number of common train units. It also does not take the order of the train units into account and there are no restrictions at the stations. Even the continuity requirement is not met. This requirement states that there should at least be one unit that travels the whole train line such that people do not need to change seats if they want to travel the complete line.

Table 2.3: The simple 3000 line and the combined 800 line.

We assume that the maximum length of the train is always 12 carriages, except between Hdr and Amr, where we assume the maximum is 10 carriages. We set the re- allocation time at 15 minutes. As an extra constraint we prohibit to cut at the stations of Ed and Zd. These stations are too small to handle the extra work that comes along with a cut.

<table>
<thead>
<tr>
<th>Hdr</th>
<th>Den Helder,</th>
<th>Ed</th>
<th>Ede-Wageningen,</th>
<th>Wt</th>
<th>Weert,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sgn</td>
<td>Schagen,</td>
<td>Ah</td>
<td>Arnhem,</td>
<td>Rm</td>
<td>Roermond,</td>
</tr>
<tr>
<td>Amr</td>
<td>Alkmaar,</td>
<td>Nm</td>
<td>Nijmegen,</td>
<td>Std</td>
<td>Sittard,</td>
</tr>
<tr>
<td>Zd</td>
<td>Zaandam,</td>
<td>Ht</td>
<td>'s Hertogenbosch,</td>
<td>Hrl</td>
<td>Heerlen,</td>
</tr>
<tr>
<td>Asd</td>
<td>Amsterdam,</td>
<td>Eh</td>
<td>Eindhoven,</td>
<td>Mt</td>
<td>Maastricht,</td>
</tr>
<tr>
<td>Ut</td>
<td>Utrecht,</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We assume that the maximum length of the train is always 12 carriages, except between Hdr and Amr, where we assume the maximum is 10 carriages. We set the re-allocation time at 15 minutes. As an extra constraint we prohibit to cut at the stations of Ed and Zd. These stations are too small to handle the extra work that comes along with a cut.
Since we do not have access to specified initial and final inventories, we try to find cyclic schedules. This is a schedule that can be used day after day. So, the inventory at the beginning and the end of the day is the same at each station. In the data, the 800 line starts two times at Hdr while it never ends there. This makes it impossible to find a cyclic schedule. To solve this, we add two rides from Sgn to Hdr. We leave it up to the model to decide whether to use the additional rides or not.

The current schedule used by NS is not cyclic. The planners solved this problem by adding empty train rides to the schedule to get the rolling stock at the right station. Empty train rides also add up to the carriage-kilometers but we have left them out. This means that the number we state to be the current carriage-kilometers needs to be incremented by the carriage-kilometers of the empty train rides.

For cutting, we apply three different strategies. The first one allows cutting in every station except for the ones where it was prohibited. So, for every train ride, the stations where we cut may be different. We will refer to this strategy as the random strategy. The second one allows cutting only per station and per train line and not at stations where it was prohibited. This means a train line is cut at a station always or never. So, for every train ride of a certain train line, the stations where we cut are the same. We will refer to this strategy as the limited strategy. The last strategy never allows cutting. This strategy will be referred to as the never strategy and produces results that are closest to the current case.

Finally we need to decide which weight factors we are going to consider. We choose weight vector \( w = (1, 1, 1) \) to start with. The second weight vector is \( (1, 0, 1) \). To prevent us from getting unrealistic results we also add an extra constraint to indicate that the seat-shortage-kilometers have to be smaller than the current seat-shortage-kilometers. This way the model does not focus too much on saving seat-shortage kilometers but just enough to not make it worse than before. In the end, we are still trying to minimize the carriage-kilometers and not the seat-shortage-kilometers. Consequently, we are not considering other weight factors in this thesis but every weight factor might be interesting for NS at some point.

2.5.2 PC and Software

To do the computations, we use a PC with an Intel Pentium III Xeon processor with a speed of 2.92 GHz and a memory (RAM) of 4096 Mb. We use the modelling software IBM ILOG OPL IDE 6.3 and the mixed integer programming solver IBM ILOG CPLEX 12.1.0.

CPLEX is able to calculate on four cores but CPLEX is faster using one core. So, we set the global default thread count to one and the memory available for working storage to 2048.0 MB. The MIP starting algorithm defines which continuous optimizer should be used to solve the initial relaxation of a MIP. The dual simplex optimizer is the default setting of CPLEX but we select the Barrier optimizer. We also change the relative objective difference cutoff from 0 to 3% to make the solution time shorter. We set the priority order of some of the decision variables as well. For the random strategy, the order is \( N_{t,m}, X_{t,b}, Y_{t,b_1,b_2} \) and then \( K_t \). For the other two strategies, the order is \( K_t, N_{t,m}, X_{t,b} \) and then \( Y_{t,b_1,b_2} \). The cutting variable \( K_t \) is more important for the limited strategy and the never strategy since there are more restrictions for this variable. For the random strategy, it does not matter where the cuts are because they are allowed at every station and therefore \( K_t \) is at the end of the list.
2.5. Solutions

2.5.3 Results and Conclusions

First, we run the model for both train lines separately and compare the strategies with each other. Second, we run the model for both train lines together. This time we do not only compare the strategies with each other but we also decide whether it is better to do the calculations for the train lines together or separately. Last, we discuss our conclusions and our next step.

We start by calculating the current carriage-kilometers (CKM) and the current seat-shortage-kilometers (SKM) to compare later. They are given in Table 2.4.

<table>
<thead>
<tr>
<th></th>
<th>3000 line</th>
<th>800 line</th>
<th>both lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>18</td>
<td>26</td>
<td>36</td>
</tr>
<tr>
<td>OA</td>
<td>17</td>
<td>14</td>
<td>29</td>
</tr>
<tr>
<td>CKM</td>
<td>107722</td>
<td>178562</td>
<td>286284</td>
</tr>
<tr>
<td>SKM</td>
<td>73856</td>
<td>34601</td>
<td>108457</td>
</tr>
</tbody>
</table>

Table 2.4: Numbers of the current schedule.

We analyze the 3000 line first because it is a simple train line and likely to have shorter solution times. This makes it easier and faster to test the implementation of the model for bugs. After all bugs are found and solved, we can start running the cases. The solution may use the same number of units of each type as the NS is using in the current schedule, namely 18 ADs and 17 OAs. Each strategy is run twice. Once for minimizing the objective described in Section 2.2 with weight vector $w = (1, 1, 1)$ and once with weight vector $w = (1, 0, 1)$ and the additional restriction such that the seat-shortage-kilometers are smaller than the current number. The results can be found in Table 2.5.

<table>
<thead>
<tr>
<th>Weight</th>
<th>Strategy</th>
<th>CKM</th>
<th>SKM</th>
<th>Deviation</th>
<th>Load</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(1,1,1)$</td>
<td>Random</td>
<td>87580</td>
<td>8518</td>
<td>20142</td>
<td>65338</td>
<td>2.7031 s</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>90364</td>
<td>9313</td>
<td>17358</td>
<td>64543</td>
<td>2.7344 s</td>
</tr>
<tr>
<td></td>
<td>Never</td>
<td>92396</td>
<td>8749</td>
<td>15326</td>
<td>65107</td>
<td>2.7188 s</td>
</tr>
<tr>
<td>$(1,0,1)^*$</td>
<td>Random</td>
<td>79320</td>
<td>73460</td>
<td>28402</td>
<td>≈ 0</td>
<td>2.6719 s</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>80900</td>
<td>73575</td>
<td>26822</td>
<td>≈ 0</td>
<td>2.6875 s</td>
</tr>
<tr>
<td></td>
<td>Never</td>
<td>82744</td>
<td>73643</td>
<td>24978</td>
<td>≈ 0</td>
<td>2.6875 s</td>
</tr>
</tbody>
</table>

Table 2.5: Numbers of the 3000 line by using the extended combination model.

Table 2.5 contains the weight vector, the carriage-kilometers, the seat-shortage-kilometers and the deviation from the current case for both measures, for all the cases. In the last two columns, an indication is given for the time to load the model and the optimization time of the model. This last factor varies a lot for the different strategies. For the carriage-kilometers, we can conclude that they are significantly less than for the current case.

Although the results might seem positive, the advice for NS is still not to cut the train lines. Firstly, the random strategy has the most positive result but it is not realistic to actually use in practice. It would become more difficult for the passengers to find out at which stations they have to switch trains since this varies per train ride. Secondly, the limited strategy only saves an extra 2000 carriage-kilometers compared to the never strategy. This is not worth the trouble of the extra planning work. This pattern is the same for both weight factors.

*Plus the additional restriction
Only minimizing the carriage-kilometers, i.e. \( w = (1, 0, 1) \), has a significant negative effect on the optimization time and the carriage-kilometers. The optimization time is almost doubled and the carriage-kilometers drop by at least 8000 kilometers compared to the weight factor \( w = (1, 1, 1) \). This is a reduction of carriage-kilometers of at least 7%. As a consequence, the seat-shortage-kilometers increase but remain a few kilometers less than the current case because of the extra restriction.

### Table 2.6: Numbers of the 800 line by using the extended combination model.

<table>
<thead>
<tr>
<th>Weight</th>
<th>Strategy</th>
<th>CKM</th>
<th>SKM</th>
<th>Deviation</th>
<th>Load</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1,1)</td>
<td>Random</td>
<td>133160</td>
<td>19674</td>
<td>45402</td>
<td>4927</td>
<td>6.6250 s</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>135660</td>
<td>22894</td>
<td>42902</td>
<td>11907</td>
<td>6.6094 s</td>
</tr>
<tr>
<td></td>
<td>Never</td>
<td>138420</td>
<td>20978</td>
<td>40142</td>
<td>13623</td>
<td>6.6406 s</td>
</tr>
<tr>
<td>(1,0,1)*</td>
<td>Random</td>
<td>126980</td>
<td>34490</td>
<td>51582</td>
<td>( \approx 0 )</td>
<td>6.5469 s</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>131840</td>
<td>34576</td>
<td>46722</td>
<td>( \approx 0 )</td>
<td>6.6250 s</td>
</tr>
<tr>
<td></td>
<td>Never</td>
<td>133330</td>
<td>34130</td>
<td>45232</td>
<td>( \approx 0 )</td>
<td>6.6563 s</td>
</tr>
</tbody>
</table>

We replace the data of the 3000 line in the model by the data of the 800 line and do the same calculations as before. The results can be found in Table 2.6. The same conclusions can be drawn as for the 3000 line i.e. cutting saves carriage-kilometers but not enough to compensate for the disadvantages. In comparison with the 3000 line, the optimization times are higher. This is a consequence of the 800 line being a combined train line. We can conclude that it is harder to find a solution when combined train lines are involved. The carriage-kilometers are also higher but this is not surprising since the 800 line is longer than the 3000 line. The difference is about 78 kilometers.

Finally, we do the calculation again for both train lines at the same time, in order to find out if it is possible to save even more carriage-kilometers. Table 2.7 gives an overview of all the numbers.

### Table 2.7: Numbers of both lines by using the extended combination model.

<table>
<thead>
<tr>
<th>Weight</th>
<th>Strategy</th>
<th>Line</th>
<th>CKM</th>
<th>SKM</th>
<th>Deviation</th>
<th>Load</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1,1)</td>
<td>Random</td>
<td>3000</td>
<td>88488</td>
<td>10226</td>
<td>19234</td>
<td>63630</td>
<td>10.8280 s</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>3000</td>
<td>92210</td>
<td>10276</td>
<td>15512</td>
<td>63580</td>
<td>10.6880 s</td>
</tr>
<tr>
<td></td>
<td>Never</td>
<td>3000</td>
<td>92656</td>
<td>9738</td>
<td>15066</td>
<td>64118</td>
<td>10.8280 s</td>
</tr>
<tr>
<td>(1,0,1)*</td>
<td>Random</td>
<td>3000</td>
<td>80462</td>
<td>73841</td>
<td>27260</td>
<td>( \approx 0 )</td>
<td>10.5000 s</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>3000</td>
<td>85750</td>
<td>73591</td>
<td>21972</td>
<td>( \approx 0 )</td>
<td>10.5310 s</td>
</tr>
<tr>
<td></td>
<td>Never</td>
<td>3000</td>
<td>86120</td>
<td>73521</td>
<td>21602</td>
<td>( \approx 0 )</td>
<td>10.5160 s</td>
</tr>
</tbody>
</table>

*Plus the additional restriction*
When we compare the deviation from the current case for the carriage-kilometers in Table 2.7 with the numbers in Tables 2.5 and 2.6, we might be tempted to conclude that it is better to do the calculations separately for every train line. But this is not true. The inventories are not the same because the inventory of both lines is not equal to the sum of the inventories of the lines separately. There are units that serve both lines. These units are counted twice for the inventories of the individual lines and counted only once for the inventory of both lines. Again, the same conclusion applies i.e. this approach is not the right one to save carriage-kilometers. To give an indication of the size of the model for each of the cases so far, we included Table A.1. In Appendix B.2, the schedules are included for all the cases in Table 2.7 with the weight factor \( w = (1,1,1) \). These schedules show that the random strategy causes a complete chaos as to where the train line should be cut. The limited strategy provides more structure but needs more carriage-kilometers. It is also clear that the resulting schedules of the never strategy are different from the current schedules due to the constraints that we relaxed or deleted like the order of the units in a combination.

Since cutting train lines is not a success, we will focus now on the never strategy. This strategy does not allow cutting but it is not the same as the current case either based on the results so far. There are multiple factors responsible for the differences with the current case. One of them is the possibility to couple or uncouple at a station. The never strategy allows coupling and uncoupling at every station while the current case works with a selected group of stations. Also, in the current case, units are dedicated to one train line or a set of train lines. The never strategy does not specify such sets. The simplifications stated in section 2.5.1 also make a difference.

We want to turn the never strategy into a realistic method to save carriage-kilometers. Therefore, some extra constraints are necessary. The first restriction is no coupling or uncoupling during rush hours at any station. The second restriction is no coupling or uncoupling during rush hours at any station except the station of Amr. The third restriction is no coupling or uncoupling ever at the stations of Ed and Zd, during rush hours at any other station except the station of Amr. This last constraint is the most severe of the three. We calculate the never strategy together with one of the new constraints and focusing on the weight factor \( w = (1,1,1) \). The results can be found in Table 2.8.

<table>
<thead>
<tr>
<th>Weight</th>
<th>Constraint</th>
<th>Line</th>
<th>CKM</th>
<th>SKM</th>
<th>Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>CKM</td>
<td>SKM</td>
<td></td>
</tr>
<tr>
<td>(1,1,1)</td>
<td>1</td>
<td>3000</td>
<td>93500</td>
<td>58845</td>
<td>14222</td>
</tr>
<tr>
<td></td>
<td></td>
<td>800</td>
<td>145860</td>
<td>63531</td>
<td>32702</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3000</td>
<td>94236</td>
<td>9768</td>
<td>13486</td>
</tr>
<tr>
<td></td>
<td></td>
<td>800</td>
<td>145980</td>
<td>21646</td>
<td>32582</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3000</td>
<td>95058</td>
<td>10240</td>
<td>12664</td>
</tr>
<tr>
<td></td>
<td></td>
<td>800</td>
<td>146940</td>
<td>21646</td>
<td>31622</td>
</tr>
</tbody>
</table>

Table 2.8: The never strategy and one of the extra constraints.

The case with the first constraint saves the most carriage-kilometers but the seat-shortage-kilometers increase. Therefore, this case is not good. Then, the second case is the best case but the third case is not much worse and more realistic. NS has to decide which case is more positive because not all factors are known to us. Maybe it is better to invest in the stations at Ed and Zd such that coupling and uncoupling is possible if the savings thanks to less carriage-kilometers are large enough. Maybe it is the other way around. Anyway, these two cases are the most
realistic ones that can save carriage-kilometers. In Appendix B.2, the schedules can be found of all the never cases with the extra constraint. They show that there is no coupling or uncoupling during the rush hours except at Amr. Outside the rush hours there are still many combination changes.

To end, we want to check if it is possible to use fewer units in the current schedule. By entering all the data about the combinations into the model, only one solution remains. We can start lowering the available units until no solution is left. The smallest inventory that is still feasible is 35 ADs and 27 OAs instead of 36 and 29. If the reallocation time is shorter, like 10 minutes instead of 15 minutes, even fewer units are necessary, i.e. 34 ADs and 26 OAs. Saving carriage-kilometers is good but saving this number of units is even better.

The reason why NS is using too many units for the 3000 and 800 line is the following. Some units do not only serve the 3000 and the 800 line but also do some trips for other train lines. Therefore, we do not know exactly how many units can be saved.

The overall conclusion is that cutting train lines does not have a large enough effect while the never strategy extended by the second or third restriction is more realistic and saves carriage-kilometers. It is also possible to work with the same schedule as currently is being used and use fewer units. To know the concrete number of units that can be put out of use, we should look at all the train lines that have a unit in common. We will not do further research on this topic here.
Chapter 3

Cancellation Model

In the previous chapter, we investigated the opportunity to cut the current train lines into smaller pieces. This is one way to save carriage-kilometers but not the only way. In this chapter, we will contemplate a second approach which is canceling train rides. We will first consider the objective and the constraints of the cancellation model. Then we will discuss the cases that we selected and their solutions. We will finish this chapter by giving conclusions about the approach.

3.1 Objective

In essence, the objective remains the same as for the combination model. We still want to minimize the carriage-kilometers (CKM), so that part of the objective function is not changed. For the same reason as was mentioned in Chapter 2, we also need to include the seat-shortage-kilometers (SKM). Only minimizing the CKM would result in trains of length zero because the CKM would be zero in that case and therefore optimal. This is not a realistic solution, hence we need to eliminate this option. Wanting to accommodate as many seats as possible for the expected passengers, is the required addition to the objective.

The SKM cannot be calculated the same way as before. We need to define what happens with the passengers of a train ride that is canceled. In reality, some of the passengers will use other transportation means, others will take the previous or next train ride. We assume, for the model, that all the passengers continue traveling by train and they are distributed equally over the previous and the next trip. This assumption can easily be changed if desired. Consequently, the number of seat shortages for trip $t$ of service class $c$ is no longer a constant but it depends on the decision variables. The objective function for the cancellation model becomes the following function:

$$w_1 \sum_{t \in T} \sum_{m \in M} d_{t,c} m N_{t,m} + w_2 \sum_{t \in T} \sum_{c \in C} d_{t,c} S_{t,c}$$

(3.1.1)

where $w = (w_1, w_2)$ is the weight factor of this function. The definitions of all the constants and (decision) variables used in the cancellation model can be found in Table 3.1. For the combination model, we used the product $s_{t,b} X_{t,b}$ to indicate that only if combination $b$ is used for trip $t$ ($X_{t,b} = 1$) the number of seats that are short, defined by $s_{t,b}$, are counted. This product is replaced by the variable $S_{t,c}$. The value of this variable is defined in a constraint and will be explained more thoroughly in the next section.
Chapter 3. Cancellation Model

Table 3.1: A list of constants and variables used in the cancellation model.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_t$</td>
<td>length of trip $t$,</td>
</tr>
<tr>
<td>$c_m$</td>
<td>number of carriages that a type $m$ unit contains,</td>
</tr>
<tr>
<td>$\delta_{t,c}$</td>
<td>passenger demand of trip $t$ for class $c$,</td>
</tr>
<tr>
<td>$A(t)$</td>
<td>trip at the beginning of the train ride that contains trip $t$,</td>
</tr>
<tr>
<td>$\alpha_t$</td>
<td>previous trip with the same arrival station and departure station as trip $t$,</td>
</tr>
<tr>
<td>$\omega_t$</td>
<td>next trip with the same arrival station and departure station as trip $t$,</td>
</tr>
<tr>
<td>$\nu(b)_m$</td>
<td>number of units of type $m$ in combination $b$,</td>
</tr>
<tr>
<td>$\kappa_{m,c}$</td>
<td>number of seats of class $c$ in a unit of type $m$,</td>
</tr>
<tr>
<td>$\sigma(t)$</td>
<td>successor of trip $t$,</td>
</tr>
<tr>
<td>$\sigma_1(t)$</td>
<td>first successor of trip $t$ if trip $t$ has more than one successor,</td>
</tr>
<tr>
<td>$\sigma_2(t)$</td>
<td>second successor of trip $t$ if trip $t$ has more than one successor,</td>
</tr>
<tr>
<td>$\beta_{b_1,b_2}$</td>
<td>indicates if the change of combination $b_1$ to $b_2$ is an allowed change,</td>
</tr>
<tr>
<td>$s_d(t)$</td>
<td>departure station of trip $t$,</td>
</tr>
<tr>
<td>$s_a(t)$</td>
<td>arrival station of trip $t$,</td>
</tr>
<tr>
<td>$\tau_d(t)$</td>
<td>departure time of trip $t$,</td>
</tr>
<tr>
<td>$\tau_a(t)$</td>
<td>arrival time of trip $t$,</td>
</tr>
<tr>
<td>$\rho(t)$</td>
<td>re-allocation time after trip $t$,</td>
</tr>
<tr>
<td>$n_m$</td>
<td>number of available units of type $m$,</td>
</tr>
<tr>
<td>$\mu_{\text{max}}^t$</td>
<td>maximum number of carriages, in total, trip $t$ can receive,</td>
</tr>
<tr>
<td>$\mu_{\text{min}}^t$</td>
<td>minimum number of carriages, in total, trip $t$ must receive,</td>
</tr>
<tr>
<td>$I_{t,m}$</td>
<td>inventory of type $m$ at station $s_d(t)$ immediately after the departure of trip $t$,</td>
</tr>
<tr>
<td>$I_{s,m}^0$</td>
<td>number of units of type $m$ stored at station $s$ at the beginning of the day,</td>
</tr>
<tr>
<td>$I_{s,m}^\infty$</td>
<td>number of units of type $m$ stored at station $s$ at the end of the day,</td>
</tr>
<tr>
<td>$C_{t,m}$</td>
<td>number of units of type $m$ that are to be coupled to right before trip $t$,</td>
</tr>
<tr>
<td>$U_{t,m}$</td>
<td>number of units of type $m$ that are to be uncoupled right after trip $t$,</td>
</tr>
<tr>
<td>$N_{t,m}$</td>
<td>number of units of type $m$ that are used for trip $t$,</td>
</tr>
<tr>
<td>$S_{t,c}$</td>
<td>number of seat shortages for trip $t$ of service class $c$.</td>
</tr>
</tbody>
</table>

3.2 Constraints

In this section, we will give an overview of all the constraints of the cancellation model. For each constraint, we will state its purpose. We will also explain why certain sets and decision variables are needed. Note that, eventhough it might seem as if all the constants and variables are the same as in Chapter 2, but not all the definitions are.

In Chapter 2, the difference between a simple train line and a combined train line is clarified. The combination model, as stated in Chapter 2, is only applicable for simple train lines. The cancellation model, described in this chapter, is operable for simple train lines as well as combined train lines.

The constants and variables that we use are listed in Table 3.1. We still need to define the sets and decision variables of the model. These can be found in Table 3.2. The decision variable that is typical for the cancellation model is $K_t$. It determines whether a train ride is deleted or not. Making the model accessible for combined train lines involves extra notation. The sets $T_S$ and $T_C$ indicate where a combined train line is spit or combined, respectively. The constants $\sigma_1(t)$ and $\sigma_2(t)$ are needed since some trips have two successors instead of one. This also means that such a trip needs two decision variables, like $Y_{t,b_1,b_2}^1$ and $Y_{t,b_1,b_2}^2$ for $Y_{t,b_1,b_2}$, to determine
3.2. Constraints

\(T\) : set of all trips,
\(T_0\) : set of all trips with no predecessor trips,
\(T_\infty\) : set of all trips with no successor trips,
\(T_S\) : set of all trips that have two successor trips,
\(T_C\) : set of all trips that will be combined at the end of the trip,
\(S\) : set of all stations,
\(C\) : set of all service classes,
\(M\) : set of all rolling stock types,
\(B\) : set of all combinations,
\(B_t\) : set of all combinations that are allowed for trip \(t\),
\(X_t,b\) : \(\begin{cases} 1 & \text{if combination } b \text{ is used for trip } t, \\ 0 & \text{otherwise,} \end{cases}\)
\(K_t\) : \(\begin{cases} 1 & \text{if the train ride starting at trip } t \text{ is deleted,} \\ 0 & \text{otherwise,} \end{cases}\)
\(Y_{t,b_1,b_2}\) : \(\begin{cases} 1 & \text{if combination } b_1 \text{ is used for trip } t \text{ and } b_2 \text{ for } \sigma(t), \\ 0 & \text{otherwise,} \end{cases}\)
\(R_{t,b_1,b_2}^S\) : \(\begin{cases} 1 & \text{if } Y_{t,b_1,b_2}^1 \text{ and } Y_{t,b_2,b_2}^2 \text{ are equal to } 1, \\ 0 & \text{otherwise,} \end{cases}\)
\(R_{\sigma(t),b_1,b_2}^C\) : \(\begin{cases} 1 & \text{if } Y_{t_1,b_1,b} \text{ and } Y_{t_2,b_2,b} \text{ are equal to } 1, \\ 0 & \text{otherwise.} \end{cases}\)

Table 3.2: A list of sets and decision variables used in the cancellation model.

which combinations are used for its two successor trips. The reason why we include \(R_{t,b_1,b_2}^S\) and \(R_{\sigma(t),b_1,b_2}^C\) in the list of decision variable is explained later in this section.

First, we state the whole cancellation model and then clarify all the constraints.

Minimize

\[
\begin{aligned}
& w_1 \sum_{t \in T} \sum_{m \in M} d_{t,c} m N_{t,m} + w_2 \sum_{t \in T} \sum_{c \in C} d_{t,c} S_{t,c}, \\
& \text{subject to}
\end{aligned}
\]

\[
\forall t \in T, c \in C: \quad S_{t,c} \geq (1 - K_{A(t)})(\delta_{t,c} + \frac{1}{2} K_{A(\alpha_t)})(\delta_{t,c}) + \frac{1}{2} K_{A(\omega_t)})(\delta_{t,c}) - \sum_{b \in B, m \in M} (\nu(m b, m, c, X_{t,b},), (3.2.2)
\]
Canceling trips

\( \forall t \in T: \)

\[ \sum_{b \in B} X_{t,b} = 1, \quad (3.2.3) \]

\( \forall t \in T_{\infty}, (b \in B; b = 0): \)

\[ X_{t,b} = K_A(t), \quad (3.2.4) \]

\( \forall t \in T \backslash (T_{\infty} \cup T_S), (b \in B; b = 0): \)

\[ Y_{t,b,b} = K_A(t), \quad (3.2.5) \]

\( \forall t \in T_S \backslash T_{\infty}, i \in \{1, 2\}, (b \in B; b = 0): \)

\[ Y_{t,b,b}^i = K_A(t), \quad (3.2.6) \]

\[ \forall t \in T \backslash (T_{\infty} \cup T_S): \]

\[ \sum_{b_1 \in B} \sum_{b_2 \in B_{\sigma(t)}} Y_{t,b_1,b_2} = 1 - K_A(t), \quad (3.2.7) \]

\[ \forall t \in T_S \backslash T_{\infty}, i \in \{1, 2\}: \]

\[ \sum_{b_1 \in B} \sum_{b_2 \in B_{\sigma(t)}} Y_{t,b_1,b_2}^i = 1 - K_A(t), \quad (3.2.8) \]

\( \forall t \in T_0: \)

\[ K_t + K_{A(\alpha_t)} \leq 1, \quad (3.2.9) \]

\( \forall t \in T_0: \)

\[ K_t + K_{A(\omega_t)} \leq 1, \quad (3.2.10) \]

\( \forall t \in T \backslash (T_{\infty} \cup T_S), b_1 \in B: \)

\[ X_{t,b_1} = \sum_{b_2 \in B} Y_{t,b_1,b_2}, \quad (3.2.11) \]

\( \forall t \in T \backslash (T_{\infty} \cup T_S), b_1 \in B: \)

\[ X_{t,b_1} = \sum_{b_2 \in B} Y_{t,b_1,b_2}^i, \quad (3.2.12) \]

\( \forall t \in T \backslash (T_{\infty} \cup T_S), b_2 \in B: \)

\[ X_{\sigma(t),b_2} = \sum_{b_1 \in B} Y_{t,b_1,b_2}^i, \quad (3.2.13) \]

\( \forall t \in T \backslash (T_{\infty} \cup T_S), b_2 \in B: \)

\[ X_{\sigma(t),b_2} = \sum_{b_1 \in B} Y_{t,b_1,b_2}, \quad (3.2.14) \]

Allowed combinations

\( \forall t \in T \backslash (T_{\infty} \cup T_S) \cup T_C, b_1 \in B_1, b_2 \in B_{\sigma(t)}: \)

\[ Y_{t,b_1,b_2} \leq \beta_{b_1,b_2}, \quad (3.2.15) \]

\( \forall t \in T_S \backslash T_{\infty}, m_1, m_2 \in M, b_1 \in B_1, b_2 \in B_{\sigma(t)}, b_3 \in B_{\sigma_2(t)}; m_1 < \nu(b_2)m_1 + \nu(b_3)m_2 < \nu(b_1)m_2 + \nu(b_3)m_2 > \nu(b_1)m_2: \)

\[ R_{t,b_1,b_2,b_3}^S = 0, \quad (3.2.16) \]

\( \forall (t_1, t_2) \in T_C \backslash T_{\infty}, m_1, m_2 \in M, b_1 \in B_1, b_2 \in B_2, b_3 \in B_{\sigma(t_1)}; \sigma(t_1) = \sigma(t_2) \land \nu(b_1)m_1 + \nu(b_2)m_2 < \nu(b_3)m_1 \land \nu(b_1)m_2 + \nu(b_2)m_2 > \nu(b_3)m_2: \)

\[ R_{t_1,b_1,b_2}^C = 0, \quad (3.2.17) \]
∀t ∈ T \ T_∞, (b_1 ∈ B_1, b_2 ∈ B_{σ_1(t)}; b_3 ∈ B_{σ_2(t)}; \sum_{m \in M} (μ(b_2)_m + μ(b_3)_m) > \sum_{m \in M} (μ(b_1)_m) + 2 ∨ \\
∀t ∈ T \ T_∞, \sum_{m \in M} (μ(b_2)_m + μ(b_3)_m) < \sum_{m \in M} (μ(b_1)_m) - 2): \\
R_{t,b_1,b_2,b_3}^S = 0, \quad (3.2.18) \\
∀t ∈ T \ (T_∞ \cup T_1 \cup T_2), m ∈ M: \\
C_{σ(t),m} = \sum_{b_1 ∈ B_1, b_2 ∈ B_{σ_1(t)}; \nu(b_2)_m > μ(b_1)_m} (μ(b_2)_m - μ(b_1)_m) Y_{t,b_1,b_2}, \quad (3.2.21) \\
∀t ∈ T \ (T_∞ \cup T_2 \cup T_3), m ∈ M: \\
U_{t,m} = \sum_{b_1 ∈ B_1, b_2 ∈ B_{σ_1(t)}; \nu(b_1)_m > μ(b_2)_m} (μ(b_1)_m - μ(b_2)_m) Y_{t,b_1,b_2}, \quad (3.2.22) \\
∀t ∈ T_0, m ∈ M: \\
C_{t,m} = N_{t,m}, \quad (3.2.23) \\
∀t ∈ T_∞, m ∈ M: \\
U_{t,m} = N_{t,m}, \quad (3.2.24) \\
∀t ∈ T_0 \ T_∞, m ∈ M: \\
C_{σ_1(t),m} = 0, \quad (3.2.25) \\
∀t ∈ T_0 \ T_∞, m ∈ M: \\
C_{σ_2(t),m} = \sum_{b_1 ∈ B_1, b_2 ∈ B_{σ_1(t)}; \nu(b_2)_m < μ(b_1)_m + μ(b_2)_m} (μ(b_1)_m + μ(b_2)_m) - μ(b_2)_m) R_{t,b_1,b_2,b_3}^S, \quad (3.2.26) \\
∀t ∈ T_0 \ T_∞, m ∈ M: \\
U_{t,m} = \sum_{b_1 ∈ B_1, b_2 ∈ B_{σ_1(t)}; \nu(b_1)_m > μ(b_1)_m + μ(b_2)_m} (μ(b_1)_m - μ(b_2)_m) R_{t,b_1,b_2,b_3}^S, \quad (3.2.27) \\
∀m ∈ M, (t_1, t_2 ∈ T_∞; σ(t_1) = σ(t_2)): \\
C_{σ(t),m} = \sum_{b_1 ∈ B_{σ_1(t)}, b_2 ∈ B_{σ_2(t)}; \nu(b_1)_m > μ(b_1)_m + μ(b_2)_m} (μ(b_1)_m - μ(b_2)_m) R_{t_1,b_1,b_2,b_2}^C, \quad (3.2.28) \\
∀m ∈ M, (t_1, t_2 ∈ T_∞; σ(t_1) = σ(t_2)): \\
U_{t_1,m} = 0, \quad (3.2.29) \\
∀m ∈ M, (t_1, t_2 ∈ T_∞; σ(t_1) = σ(t_2)): \\
(\text{Un})coupling
\[
U_{t_2,m} = \sum_{b \in B_{s(t_1)}, b_1 \in B_{t_1}, b_2 \in B_{s(t_2)}, \nu(b)_m < \nu(b_1)_m + \nu(b_2)_m} \left( \nu(b_1)_m + \nu(b_2)_m - \nu(b)_m \right) R^C_{t_1,b_1,b_2},
\]

(3.2.30)

**Inventory**

\[
I_{t_1,m} = I^{0}_s d(t_1)_m - \sum_{t_2 \in T: \sigma(t_2) = \sigma(t_1), \tau_d(t_2) \leq \tau_d(t_1)} C_{t_2,m} + \sum_{t_2 \in T: \sigma_d(t_2) = \sigma_d(t_1), \tau_d(t_2) \leq \tau_d(t_1) - \rho(t_2)} U_{t_2,m},
\]

(3.2.31)

\[
I^{\infty}_s,m = I^{0}_s d(t)_m - \sum_{t \in T: \sigma_d(t) = \sigma_s(t)} C_{t,m} + \sum_{t \in T: \sigma_a(t) = \sigma_s(t)} U_{t,m},
\]

(3.2.32)

\[
n_m = \sum_{s \in S} I^{0}_s,m,
\]

(3.2.33)

\[
I^{0}_s,m = I^{\infty}_s,m,
\]

(3.2.34)

**Variable domains**

\[
X_{t,b} \in \{0,1\},
\]

(3.2.35)

\[
Y_{t,b_1,b_2} \in \{0,1\},
\]

(3.2.36)

\[
Y^1_{t,b_1,b_2} \in \{0,1\},
\]

(3.2.37)

\[
Y^2_{t,b_1,b_2} \in \{0,1\},
\]

(3.2.38)

\[
K_t \in \{0,1\},
\]

(3.2.39)

\[
R^S_{t,b_1,b_2,b_3} \in \{0,1\},
\]

(3.2.40)

\[
R^C_{\sigma(t_1),b_1,b_2,b_3} \in \{0,1\},
\]

(3.2.41)

\[
N_{t,m}, C_{t,m}, U_{t,m}, I_{t,m} \in \mathbb{R}^+_0,
\]

(3.2.42)

\[
I^{0}_s,m, I^{\infty}_s,m \in \mathbb{R}^+_0,
\]

(3.2.43)

\[
S_{t,c}.
\]

(3.2.44)

Constraints (3.2.2) define the seat-shortage for each trip and service class. Once the combination of the trip is known the capacity of the train is also known. If a trip is canceled, then the expected passengers are divided over the previous and the next trip. We define the previous trips of the
first train ride of the day to be that same trip. Equivalently for the next trips of the last train ride.

The purpose of constraints (3.2.3) is assigning exactly one combination to a trip. Constraints (3.2.4), (3.2.5) and (3.2.6) state that if a train ride is canceled then all the involved trips get combination zero. This means that no units are used for those trips. Constraints (3.2.7) and (3.2.8) guarantee that if the train ride of the trip is not canceled then exactly one allowed combination is used for the trip and exactly one allowed combination is used for a successor of the trip.

If a train ride is deleted, the passengers are distributed over the previous and the next trips. Therefore, we need to make sure that the previous and next trips are not canceled as well. Constraints (3.2.9) and (3.2.10) accomplish this. By defining the previous trips of the first train ride of the day as being the same trips, we make sure that the first train ride of the day is never canceled. Therefore we avoid having the problem of not knowing how to distribute the passengers. The same counts for the next trips of the last train ride of the day. The constraints (3.2.11) to (3.2.14) make sure that the decision variables used in all the constraints indicate the same combination for each trip.

A change of combination between a trip and its successor is only allowed if the length of the combinations does not differ more than two units and it is prohibited to add and remove units at the same time. Constraints (3.2.15) indicates this for a trip with only one successor. Constraints (3.2.16) to (3.2.19) guarantee this for trips with two successors (splitting) and trips that have the same successor (combining). We use the decision variables $R^S$ and $R^C$ since we need to know which combinations are used for the three involved trips.

Constraints (3.2.20) define for each trip the number of units used of each type of material. To be able to calculate the inventories in the end, we need to register where and when units are coupled or uncoupled. Constraints (3.2.21) to (3.2.30) serve this purpose. Ultimately, the inventories are defined by constraints (3.2.31) and (3.2.32). Constraints (3.2.33) state that the inventory at the beginning of the day may not exceed the total number of available units while constraints (3.2.34) guarantees that, for each type of material, the inventory at the beginning of the day is equal to the inventory at the end of the day at each station. Finally, constraints (3.2.35) to (3.2.44) define the variable domains.

This cancellation model is a basic version. Extra constraints can be added like prohibiting to cancel a train ride during rush hours or specifying an inventory at the beginning and end of the day instead of assuming them being equal. We leave it up to NS to determine any further restrictions.

3.3 Cases

After explaining the cancellation model’s objective and constraints, it is now possible to test a few cases. We will first state which data, settings, PC and software we use. Thereafter, we will define the cases and the results of our calculations.

3.3.1 Data and Settings

We use the same datafile as for the combinations model which is the timetable and rolling stock schedule of February 2nd of 2009. Our focus is again on the simple 3000 line and the combined 800 line. More information about these lines is shown in Table 2.3. We provide the same number of units of each type of material as in the original schedule. The re-allocation time is set to 15
minutes and the maximum number of carriages for a combination is 12 except between Hdr and Amr where it is 10.

For calculations only concerning the 800 line, we need to add trips to get feasible solutions. We add the same trips as we did for the combination model.

The last setting is the weight factor $w = (w_1, w_2)$. We set this factor equal to $(1, 1)$ such that we can compare the results of the cancellation model to the results of the combination model. We do not consider $(1, 0)$ as a weight factor for all the cases. The reason becomes clear when we state the conclusions of the cancellation model. If we use $(1, 0)$ as the weight factor, then we also include the same extra restriction as we did for the combination model namely that the SKM have to remain less than the current SKM for each train line individually. Again, there are many more options for the weight factor but we leave it up to NS to select the appropriate weight factor for each situation.

3.3.2 PC and Software

We use the same PC and the cancellation model is also implemented in IBM ILOG OPL IDE 6.3 and solved by IBM ILOG CPLEX 12.1.0. CPLEX is allowed to calculate on only one core because it is faster. The memory available for working storage is set to 2048.0 MB. The relative objective difference cutoff and the MIP starting algorithm are set according to the settings for the combination model i.e. 3% and the Barrier optimizer, respectively.

The main difference with the combination model in terms of software-settings is the priority order of the decision variables. We do not define such an order of the cancellation model even though it could be defined.

3.3.3 Case I and II

The first case only concerns the 3000 line. We foresee 18 ADs and 17 OAs to be used for the solution. The second case uses the 800 line and we provide 26 ADs and 14 OAs. For both cases, the only restrictions as to the cancellations are the ones already in the model. Not only did we observe these cases, our intention was to compare them with a case that considered the 3000 line and the 800 line together. The problem is that there is not enough memory to do this calculation. Therefore, we are forced to leave out this case.

There is an advantage to considering the two train lines separately. We assumed that if a train ride is canceled then the passengers are divided equally over the previous and the next trips. Therefore, since the 800 line and 3000 line have a common route, it is possible that the previous or next trip is not of the same train line. By considering one train line at a time, we are sure that passengers do not have to switch trains more often than before.

A disadvantage lies in the provided units. Some units are used for multiple train lines in the original schedule while in our solutions they are assigned exclusively to one train line. These units are therefore counted twice but it is not sure that eventually they can be replaced by the original number of units. It is possible that the new schedule is so tight that no unit can be eliminated. The worst case scenario is that, for each unit serving multiple train lines in the original schedule, we need an additional unit in the new schedule.

The results for case I and case II can be found in Table 3.3. This table contains almost the same columns as the tables in Subsection 2.5.3. We added the last column to indicate how many train rides were canceled, compared to the current schedule. Nearly all canceled train rides were scheduled outside the rush hours. Figure B.15 and Figure B.16 show the schedules of case I and
3.3 Cases

Table 3.3: Numbers of case I and II by using the cancellation model.

<table>
<thead>
<tr>
<th>Weight</th>
<th>Case</th>
<th>CKM</th>
<th>SKM</th>
<th>Deviation</th>
<th>Cancel</th>
<th>Load</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1,1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>84156</td>
<td>9701</td>
<td>23566</td>
<td>64155</td>
<td>46.3590 s</td>
<td>5576.8000 s</td>
</tr>
<tr>
<td>II</td>
<td>II</td>
<td>127972</td>
<td>21366</td>
<td>50590</td>
<td>13235</td>
<td>72.0160 s</td>
<td>56660.0000 s</td>
</tr>
</tbody>
</table>

Table 3.4: Numbers of case III to V by using the cancellation model.

<table>
<thead>
<tr>
<th>Weight</th>
<th>Case</th>
<th>Line</th>
<th>CKM</th>
<th>SKM</th>
<th>Deviation</th>
<th>Cancel</th>
<th>Load</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1,1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>III</td>
<td>3000</td>
<td>100640</td>
<td>72031</td>
<td>7082</td>
<td>1825</td>
<td>7</td>
<td>256.0600 s</td>
</tr>
<tr>
<td></td>
<td>IV</td>
<td>3000</td>
<td>97326</td>
<td>42372</td>
<td>10396</td>
<td>31484</td>
<td>15</td>
<td>256.3000 s</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>3000</td>
<td>98914</td>
<td>62978</td>
<td>8808</td>
<td>10878</td>
<td>14</td>
<td>257.8300 s</td>
</tr>
<tr>
<td></td>
<td>(1,0)*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>III</td>
<td>3000</td>
<td>100640</td>
<td>73825</td>
<td>7082</td>
<td>1825</td>
<td>7</td>
<td>255.7200 s</td>
</tr>
<tr>
<td>IV</td>
<td>IV</td>
<td>3000</td>
<td>94718</td>
<td>73831</td>
<td>13004</td>
<td>15720</td>
<td>17</td>
<td>256.1600 s</td>
</tr>
<tr>
<td>V</td>
<td>V</td>
<td>3000</td>
<td>92082</td>
<td>73852</td>
<td>15642</td>
<td>15112</td>
<td>18</td>
<td>256.2000 s</td>
</tr>
</tbody>
</table>

3.3.4 Case III - V

In case III to V, we are searching for smaller deviations from the current schedule than the results for case I and II. For case III, we only allow canceling train rides while the rest of the current schedule must remain unchanged. The capacity of the combinations of the previous and next trips of a canceled train ride already need to be large enough for the extra passengers, since the combinations of these trips can not be changed. If the combinations do not have this extra room, the SKM will increase. Case IV starts from the current schedule and allows to delete train rides and to add extra units to any combination. It is not sustained to replace a combination for a shorter one. Finally, case V permits to delete train rides from the current schedule but unlike case IV it is only allowed to add extra units to the combinations of previous trips or next trips of a canceled train ride. The resulting schedule of case V will be as close or closer to the current schedule than the resulting schedule of case IV.

Remember that the current schedule is not a cyclic schedule, this means that the inventory at the beginning and the end of the day are not the same at each station. To make it cyclic, extra (empty) train rides are needed and therefore more carriage-kilometers will be inevitable. The current schedule is the basis of the solutions of case III to V hence we need to remove constraints (3.2.34) from the cancellation model to obtain the necessary results. These results are depicted in Table 3.4.

*Plus the additional restriction
This table has the same structure as Table 2.7 plus a column titled “Cancel”. The extra column contains the number of train rides of each train line that were canceled in the solution. As was the case in the first two cases, nearly all canceled train rides were scheduled outside the rush hours. The schedules of these cases can be found in Figure B.17 to B.22. These schedules are more structured than the schedules of case I and II.

All the cases are explained and the solutions are shown in the tables. We continue in the next section with conclusions about the results of every case.

3.4 Conclusions

Case I can be compared to the case of the combination model with weight factor (1, 1, 1) and the never strategy where we only included the 3000 line. Case II can be compared to the same case where we used only the 800 line. The conclusions of both comparisons are the same; namely that there are less carriage-kilometers in case I and II than in the cases of the combination model but there are more seat-shortage-kilometers. For the 3000 line, we save 7.6% extra on carriage-kilometers but lose 1.2% on seat-shortage-kilometers. For the 800 line, we save 5.9% extra on carriage-kilometers but lose 1.1% on seat-shortage-kilometers.

The consequences of canceling train rides are practically easier than cutting train lines. A cancellation removes all the work of that train ride while a cut introduces more work at the station of the cut. Therefore it would cost less money and effort to make a schedule of the cancellation model operative than a schedule of the combination model. On the other hand, NS offers less service to its customers by canceling train rides. This is a balance that only the company can make.

Cases III to V are closer to the current situation. Because of this, there are a lot of carriage-kilometers compared to case I and II and the seat-shortage-kilometers are even worse. The positive side of these cases is that they are easier to schedule and have a shorter solution time than case I and II. As we mentioned before, the current schedule is not cyclic and therefore the solutions of case III to V are not cyclic. It is possible that more empty train rides are needed to make the schedule of case III to V cyclic than to make the current schedule cyclic. This would result in even more carriage-kilometers for the cases.

We also tried to do the same calculation with weight factor (1, 0) but we encountered some problems. Case I and case II ran out of memory, consequently we only consider weight factor (1, 0) for cases III, IV and V. The effect of the weight factor (1, 0) is that the schedule needs the same number of carriage-kilometers or less than with weight factor (1, 1). Moreover, we cancel at least the same number of train rides.

An overall conclusion is that the canceled train rides were all scheduled outside the rush hours. This becomes clear in the schedules of Figure B.15 to B.22. These are the schedules of all the cases using weight factor (1, 1). We can also conclude that the schedules of cases III, IV and V in Figure B.17 to B.22 contain less combination changes than the schedules of cases I and II in Figure B.15 and B.16. The more carriage-kilometers and seat-shortage-kilometers NS wants so save, the more changes they have to make in the schedules of the 800 line and 3000 line.
Chapter 4

Line Model

Until now, we have only changed the frequency of a train line or cut an existing line in several parts. In this chapter, we will take line planning to the next level. No more modifications on the current line plan. We will develop a model that constructs a whole new line plan. Column generation is an important technique for the model. First, we will explain the model and the theory of column generation. Second, we will test the model on a small case. We will close this chapter by giving the conclusions of this line model.

4.1 Objective

The overall goal is reducing the number of carriage-kilometers. This is still the case for the line model but we need to be a bit more subtle. At this moment, many train lines consist of a part with many passengers and a part with almost no passengers such that a large combination of rolling stock is required but not necessary the entire ride. It is possible to save carriage-kilometers by defining train lines that have a more or less constant number of passengers on them during the entire ride. That case we can assign an appropriate combination of rolling stock to them. This gives us the opportunity to use small combinations on train lines with fewer passengers while this is currently not possible, so we can reduce the number of carriage-kilometers.

For this model, we allow no seat shortages. Every traveler should be able to have a seat the entire time. The reason for this is explained later. We do not exclude the possibility to have to change seats during the journey, for example, it is possible that a train unit is uncoupled at some station. If a passenger needs to change to a train of another line at some station, then the passenger has no direct connection at that station. It might be that the passenger has to wait for some time for the connecting train to arrive. This is not considered an advantage. Therefore, we want to reduce the number of bad connections by counting the effected passengers for each bad connection and minimize them in the objective.

The sum of the carriage-kilometers and the bad connections is called the cost of a train line. For the bad connections, we need to decide which bad connections count for a certain train line. We define this as follows: for each trip of the train line, the bad connections are the connections with the possible successor trips that are not the actual successor of the trip.

There is still one problem for the model. We can not construct a model like the previous two. It requires too much memory of a computer to construct all the possible train lines and then select the right ones. Therefore we use another technique namely column generation. In the next section we explain the general purpose of column generation. Thereafter, the technique is applied to our problem.
4.2 Column Generation

Column generation is a well studied subject. Since we want to make clear that our problem indeed can be formulated by means of column generation, we start by considering column generation in full generality. Therefore, we cite Section 2.1 of Chapter 1 from [9], pp. 8-9, to explain column generation:

Let us call the following linear program the *master problem* (MP).

\[
\begin{align*}
\bar{z}^*_{MP} & := \min \sum_{j \in J} c_j \lambda_j \\
\text{subject to} & \sum_{j \in J} a_j \lambda_j \geq b \\
\lambda_j & \geq 0, j \in J.
\end{align*}
\]  

(4.2.1)

(4.2.2)

(4.2.3)

In each iteration of the simplex method we look for a non-basic variable to price out and enter the basis. That is, given the non-negative vector \( \pi \) of dual variables we wish to find a \( j \in J \) which minimizes \( \bar{c}_j := c_j - \pi^t a_j \). This explicit pricing is a too costly operation when \( |J| \) is huge. Instead, we work with a reasonably small subset \( J' \subseteq J \) of columns - the restricted master problem (RMP) - and evaluate reduced costs only by implicit enumeration. Let \( \lambda \) and \( \pi \) assume primal and dual optimal solutions of the current RMP, respectively. When columns \( a_j, j \in J \), are given as elements of a set \( A \), and the cost coefficient \( c_j \) can be computed from \( a_j \) via a function \( c \) then the subproblem

\[
\bar{c}^* := \min_{a \in A} \{c(a) - \pi^t a\}
\]  

(4.2.4)

performs the pricing. If \( \bar{c}^* \geq 0 \), there is no negative \( \bar{c}_j, j \in J \), and the solution \( \lambda \) to the restricted master problem optimally solves the master problem as well. Otherwise, we add to the RMP the column derived from the optimal subproblem solution, and repeat with re-optimizing the RMP. The process is initialized with an artificial, a heuristic, or a previous (“warm start”) solution. In what regards convergence, note that each \( a \in A \) is generated at most once since no variable in an optimal RMP has negative reduced cost. When dealing with some finite set \( A \) (as is practically always true), the column generation algorithm is exact. In addition, we can make use of bounds. Let \( \bar{z} \) denote the optimal objective function value to the RMP. When an upper bound \( \kappa \geq \sum_{j \in J} \lambda_j \) holds for the optimal solution of the master problem, we have not only an upper bound \( \bar{z} \) on \( \bar{z}^*_{MP} \) in each iteration, but also a lower bound: we can not reduce \( \bar{z} \) by more than \( \kappa \) times the smallest reduced cost \( \bar{c}^* \):

\[
\bar{z} + \kappa \bar{c}^* \leq \bar{z}^*_{MP} \leq \bar{z}.
\]  

(4.2.5)

Thus, we may verify the solution quality at any time. In the optimum of (4.2.1), \( \bar{c}^* = 0 \) for the basic variables, and \( \bar{z} = z^*_{MP} \).

The citation above is a short explanation of column generation. The set \( J \) for the line model is the set of all possible train lines. The costs are the carriage-kilometers and the bad connections. The main deviation from the citation is that the line model can not be formulated as a linear problem (LP). It is an integer problem (IP). This issue can be solved by relaxing the IP so it becomes an LP. Then, we apply column generation on the linear relaxation. After the
optimum of the LP is found, we can solve the IP using the set of train lines that is constructed for the linear relaxation. This is not the optimal solution of the original problem since the solution of the linear relaxation may not contain the optimal train lines for the original IP. To solve the original IP, we need to apply a branch-and-bound algorithm. We only solve the root node and leave the rest as an extension. In the following section, we will formulate the line model.

4.3 Constraints

The purpose of the line model is to find out which train lines out of all possible train lines are to be selected to minimize the carriage-kilometers and the bad connections. Therefore, we need to know the set of all stations and the set of all couples of stations that are directly connected with rails. This means that the rail-network should be known in order to use the line model. We also need to know which types of train lines we need to distinguish. The characteristics of these types usually involve the set of stations where the train must stop and the preferred type of material. For each type of train line, we construct a type graph. This type graph contains all the stations where the trains of this type are allowed to stop. These stations are the nodes of the graph. There is an edge between two nodes of the graph if there are rails between the two stations and the route does not include another station of the graph. Each edge has a frequency. The edge should be part of at least as many train lines as the frequency.

To determine the bad connections and the number of passengers that use an edge, we use an origin-destination-matrix (OD-matrix). This matrix contains for every two stations how many passengers travel from the first station to the second station. It does not specify which route they take since this depends on the line planning that we are constructing. Consequently, we need to define which routes the passengers use. Passengers typically choose the fastest way to get to their destination. It is impossible to predict which route every passenger will take. Therefore we need an assumption. We assume that every passenger choses the shortest route concerning distance. If there are multiple types of train lines covering the same part of the trip, then the passenger chooses the type that stops at as few stations as possible.

The allowed combination changes for the material remains the same as for the combination model and cancellation model. At most two units can be coupled or uncoupled at the same station. Coupling and uncoupling at the same station is also prohibited. If it is a combined train line, then the combinations of the two ends are counted as one to check whether the combination changes are allowed.

To finish this introduction of the data that we need for the line model, we mention that the term “path” is used as a synonym for the term “train line”.

4.3.1 Restricted Master Problem

Before we start with the explanation of the restricted master problem, Table 4.1 gives an overview of the constants, variables and sets that are used in this problem. The passenger demand δ_i is the demand of one hour. There are train rides along the selected train lines ones every hour. If we know the length of the path, then we can define how many units we actually need. In the model we assume that we need new material for each train ride.

The restricted master problem selects, from the subset of all paths \( P^\prime \), a set of paths that covers the entire type graph with the right frequency. This results in the following minimization problem.
Chapter 4. Line Model

$c_p$ : cost of path $p$, i.e. the sum of the CKM and the bad connections,
$f_i$ : frequency of edge $i$,
$\delta_i$ : passenger demand of edge $i$ of the type graph,
$\nu(j)_m$ : number of units of type $m$ in combination $j$,
$\kappa_j$ : number of seats in combination $j$,
$n_m$ : number of available units of type $m$,
$u_{m,p}$ : number of units of type $m$ that are necessary for path $p$,
$Q_{p,i,j} = \begin{cases} 1 & \text{if path } p \text{ contains edge } i \text{ and uses combination } j \text{ for it,} \\ 0 & \text{otherwise,} \end{cases}$
$\mathcal{P}'$ : set of all constructed paths (this is a subset of $\mathcal{P}$, the set of all paths),
$\mathcal{I}$ : set of all edges of the type graph,
$\mathcal{J}$ : set of all combinations,
$\forall p \in \mathcal{P}'$: 
$X_p = \begin{cases} 1 & \text{if path } p \text{ is used,} \\ 0 & \text{otherwise.} \end{cases}$

Table 4.1: A list of constants, sets and variables used in the restricted master problem.

Minimize

$$\sum_{p \in \mathcal{P}} c_p X_p,$$

subject to

$\forall i \in \mathcal{I}$:

$$\sum_{p \in \mathcal{P}'} \sum_{j \in \mathcal{J}} Q_{p,i,j} X_p \geq f_i,$$

$\forall i \in \mathcal{I}$:

$$\sum_{p \in \mathcal{P}'} \sum_{j \in \mathcal{J}} Q_{p,i,j} \kappa_j X_p \geq \delta_i,$$

$\forall m \in \mathcal{M}$:

$$\sum_{p \in \mathcal{P}'} u_{m,p} X_p \leq n_m.$$

$\forall p \in \mathcal{P}'$

$$X_p \in \{0, 1\}$$

Constraints (4.3.2) guarantee that each edge of the type graph is selected for a path at least the minimum frequency. Constraints (4.3.3) make sure that there are enough seats for every edge of the type graph. Unlike the combination and cancellation model, we do not minimize the seat shortage kilometers. For this model, we want everyone to have a seat. If it is not possible to give everyone a seat using the largest combination on a certain edge, the frequency should be augmented. Due to constraints (4.3.4) the selection of paths does not use more than the available units of each type of material. These are the only restrictions concerning the inventory at each station. The combinations model and the cancellation model both have constraints that indicate the inventory at the beginning and ending of the day. They also make sure that
the units that leave a station are in fact present. Since in this model, we do not know which train lines arrive first, we can not keep track of the inventories at the stations. When the time schedule of the train lines is known, it is possible to check how many units are needed in reality. This might turn out to be less units than we provide in the model. The reason is that all the units of all the train lines are counted while in fact some units can be used for multiple train lines. Finally, constraints \(4.3.5\) define the variable domains.

The initial set of paths \(P'\) should be provided by us. Then, in each step the subproblem selects a path that can make the solution of the restricted master problem better. This path is added to the set \(P'\). Thereafter the restricted master problem is again solved using this new set \(P'\). The definition of a path is specified in the subproblem and the initial set \(P'\) contains paths of the same form.

### 4.3.2 Subproblem

We solve the subproblem to determine whether a path of negative reduced cost exists. This path should contain edges that fit together, the edges have to describe a connected route through the network. The subproblem also determines which combination is used for each edge of the path.

Before we start with the explanation of the subproblem, we define some new constants, sets and variables that are used. See Table 4.2 and Table 4.3 for an overview. The subproblem also uses some of the constants and sets of the restricted master problem. These are not stated again in Table 4.2 but can be found in Table 4.1.

| \(\pi_1(i)\) | shadow price of edge \(i\) of constraints \(4.3.2\) |
| \(\pi_2(i)\) | shadow price of edge \(i\) of constraints \(4.3.3\) |
| \(\pi_3(m)\) | shadow price of material \(m\) of constraints \(4.3.4\) |
| \(\mu_j\) | number of carriages in combination |
| \(d_i\) | length of edge \(i\) |
| \(\varphi_{i_1,i_2}\) | number of passengers with a bad connection between edge \(i_1\) and \(i_2\) |
| \(s_a(i)\) | station of arrival of edge \(i\) |
| \(s_d(i)\) | station of departure of edge \(i\) |
| \(\tau_1(i)\) | type of edge \(i\) |
| \(\tau_2(j)\) | type of combination \(j\) |
| \(\beta_{1,j_1,j_2}\) | indicates if the change of combination \(j_1\) to \(j_2\) is an allowed change |
| \(\beta_{2,j_1,j_2,j_3}\) | indicates if the change of combinations \(j_1\) and \(j_2\) to \(j_3\) is an allowed change (in case of splitting or combining) |
| \(S\) | set of all stations |
| \(M\) | set of all rolling stock types |

Table 4.2: A list of constants and sets used in the subproblem.

Variables \(S^k_0(s)\) and \(S^k_{\infty}(s)\) are only defined for \(k \in \{1, 2\}\). This means that there can be at most two origins and at most two endings of the path. This prevents the path from branching over the type graph. We decided that such paths are too complicated to use as a train line.

We are now ready to state the entire minimization model of the subproblem.

Minimize

\[
C - \sum_{i \in I} \sum_{j \in J} \left( \pi_1(i) Y_{i,j} - \pi_2(i) Y_{i,j} \kappa_j - \sum_{m \in M} \pi_3(m) U_m \right),
\] (4.3.6)
\( C \): cost of the path,
\( U_m \): number of units of type \( m \) that are necessary for path,
\( N_{s,m}^1 \): number of units of type \( m \) that are necessary for edges that arrive at station \( s \),
\( N_{s,m}^2 \): number of units of type \( m \) that are necessary for edges that depart at station \( s \),

\[ \forall i \in I, j \in J: \]
\[ Y_{i,j} = \begin{cases} 1 & \text{if the path contains edge } i \text{ with combination } j, \\ 0 & \text{otherwise}, \end{cases} \]

\[ \forall (i_1, i_2) \in I: s_a(i_1) = s_d(i_2): \]
\[ R_{i_1, i_2} = \begin{cases} 1 & \text{if the path contains edge } i_1 \text{ and does not contain edge } i_2, \\ 0 & \text{otherwise}. \end{cases} \]

\[ \forall i_1, i_2 \in I: \]
\[ K_{i_1, i_2} = \begin{cases} 1 & \text{if the path contains edge } i_1 \text{ and edge } i_2, \\ 0 & \text{otherwise}. \end{cases} \]

\[ \forall s \in S, k \in \{1, 2\}: \]
\[ S_0^k(s) = \begin{cases} 1 & \text{if the } k^{th} \text{ origin of the path is at station } s, \\ 0 & \text{otherwise}. \end{cases} \]
\[ S_\infty^k(s) = \begin{cases} 1 & \text{if the } k^{th} \text{ end of the path is at station } s, \\ 0 & \text{otherwise}. \end{cases} \]

\[ \forall s \in S: \]
\[ S_S(s) = \begin{cases} 1 & \text{if the path is split at stations}, \\ 0 & \text{otherwise}. \end{cases} \]

\[ S_C(s) = \begin{cases} 1 & \text{if the path is combined at stations}, \\ 0 & \text{otherwise}. \end{cases} \]

Table 4.3: A list of variables used in the subproblem.

subject to

\[ \forall i \in I, j \in J: \]
\[ C = Y_{i,j} \mu_j d_i + \sum_{i_1, i_2 \in I, s_a(i_1) = s_d(i_2)} \varphi_{i_1, i_2} R_{i_1, i_2}, \]  \( (4.3.7) \)

\[ \forall s \in S, m \in M: \]
\[ N_{s,m}^1 = \sum_{i \in I: s_d(i) = s} \sum_{j \in J} Y_{i,j} \nu(j)_m, \]  \( (4.3.8) \)

\[ \forall s \in S, m \in M: \]
\[ N_{s,m}^1 = \sum_{i \in I: s_a(i) = s} \sum_{j \in J} Y_{i,j} \nu(j)_m, \]  \( (4.3.9) \)

\[ \forall m \in M: \]
\[ U_m = \sum_{i \in I} \sum_{j \in J} Y_{i,j} \nu(j)_m - \sum_{s \in S} \left( \min_{k \in \{1, 2\}} N_{s,m}^k \right), \]  \( (4.3.10) \)
∀i ∈ I:
\[ \sum_{j \in J} Y_{i,j} \leq 1, \]  
(4.3.11)

**Shape of the path**
\[ \sum_{s \in S} S_0^1(s) = 1, \]  
(4.3.12)
\[ \sum_{s \in S} S_0^1(s) = 1, \]  
(4.3.13)
\[ \sum_{s \in S} S_C(s) \leq 1, \]  
(4.3.14)
\[ \sum_{s \in S} S_S(s) \leq 1, \]  
(4.3.15)
\[ \sum_{s \in S} S_C(s) = \sum_{s \in S} S_0^2(s), \]  
(4.3.16)
\[ \sum_{s \in S} S_S(s) = \sum_{s \in S} S_\infty^2(s), \]  
(4.3.17)
∀s ∈ S:
\[ S_0^1(s) + S_0^1(s) + S_\infty^1(s) + S_\infty^2(s) + S_S(s) + S_C(s) \leq 1, \]  
(4.3.18)
∀s ∈ S:
\[ \sum_{i_1 \in I, j_1 \in J: s_d(i_1) = s} Y_{i_1,j_1} - \sum_{i_2 \in I, j_2 \in J: s_a(i_2) = s} Y_{i_2,j_2} = S_0^1(s) + S_0^2(s) - S_\infty^1(s) - S_\infty^2(s) + S_S(s) - S_C(s), \]  
(4.3.19)
∀s ∈ S:
\[ \sum_{i \in I, j \in J: s_d(i) = s \land s_a(i) = s} Y_{i,j} \leq 2 - S_0^1(s) - S_0^2(s) - S_\infty^1(s) - S_\infty^2(s) + S_S(s) + S_C(s), \]  
(4.3.20)
∀s ∈ S:
\[ \sum_{i_2 \in I, j_2 \in J: s_a(i_2) = s} Y_{i_2,j_2} \geq 2S_C(s), \]  
(4.3.21)
∀s ∈ S:
\[ \sum_{i_2 \in I, j_2 \in J: s_d(i_2) = s} Y_{i_2,j_2} \geq 2S_S(s), \]  
(4.3.22)

**Type of the path**
∀i_1, i_2 ∈ I:
\[ \tau_1(i_1)K_{i_1,i_2} = \tau_1(i_2)K_{i_1,i_2}, \]  
(4.3.23)
∀i ∈ I, j ∈ J:
\[ \tau_1(i)Y_{i,j} = \tau_2(j)Y_{i,j}, \]  
(4.3.24)

**Allowed combinations of the path**
∀(i_1, i_2) ∈ I, j_1, j_2 ∈ J, s_a(i_1) = s_d(i_2):
\[ \beta_{j_1,j_2}^1 \geq Y_{i_1,j_1} + Y_{i_2,j_2} - 1 - 2S_C(s_a(i_1)) - 2S_S(s_a(i_1)), \]  
(4.3.25)
∀(i_1,i_2,i_3 \in I, j_1,j_2,j_3 \in J: s_u(i_1) = s_u(i_2) \wedge s_u(i_3) = s_u(i_1) \wedge s_u(i_3) \neq s_u(i_2) \wedge s_u(i_1) \neq s_u(i_2)):
\[ \beta^2_{j_1,j_2,j_3} \geq Y_{i_1,j_1} + Y_{i_2,j_2} + Y_{i_3,j_3} + S_C(s_u(i_1)) - 3, \]  
(4.3.26)
∀(i_1,i_2,i_3 \in I, j_1,j_2,j_3 \in J: s_u(i_1) = s_u(i_2) \wedge s_u(i_3) = s_u(i_1) \wedge s_u(i_3) \neq s_u(i_2) \wedge s_u(i_1) \neq s_u(i_2)):
\[ \beta^2_{j_1,j_2,j_3} \geq Y_{i_1,j_1} + Y_{i_2,j_2} + Y_{i_3,j_3} + S_S(s_d(i_1)) - 3. \]  
(4.3.27)

Variable domains
\[ C \in \mathbb{R}_+, \]  
(4.3.28)
∀i \in I, j \in J:
\[ Y_{i,j} \in \{0, 1\}, \]  
(4.3.29)
∀i_1,i_2 \in I:
\[ R_{i_1,i_2}, K_{i_1,i_2} \in \{0, 1\}, \]  
(4.3.30)
∀s \in S, k \in \{1, 2\}:
\[ S^k_0(s), S^k_\infty(s), S_C(s), S_S(s) \in \{0, 1\}, \]  
(4.3.31)

The objective function has the same form as the objective function of definition (4.2.4). The cost of the path is defined in constraint (4.3.7). The number of units of each rolling stock type that are necessary for the resulting path is determined in constraints (4.3.8) to (4.3.10). Constraints (4.3.11) make sure that at most one combination can be assigned to an edge.

Constraints (4.3.12) to (4.3.20) restrict the shapes of the paths. Constraints (4.3.12) and (4.3.13) guarantee that the path has a beginning and an ending. Constraints (4.3.14) and (4.3.15) allow the path to split and combine once. If the path is combined and/or split, there are two beginnings and/or two endings of the path. Constraints (4.3.16) and (4.3.17) provide the extra beginning and/or end. We define a special station as being a station where the path begins or ends or where the path is split or combined. Constraints (4.3.18) prevent a station to be special for multiple reasons. For example, it is not allowed to combine and split at the same station or to split and end at the same station. Constraints (4.3.19) make sure that the path is connected while constraints (4.3.20) to (4.3.22) allow at most two edges to begin or end at a station unless there is a split of combination at that station, then three edges are allowed.

Constraints (4.3.23) guarantee that all the edges of the path are of the same type. For example, all the edges are of the inter-city type. To make sure that the right type of material is used for a certain type of path, we included constraints (4.3.24).

The allowed changes of combinations are the same as for the combination model and the cancellation model. No more than two units can be coupled or uncoupled and, coupling and uncoupling at the same station is not allowed. Constraints (4.3.25) to (4.3.27) regulate the allowed changes of combinations for stations with a simple arrival and departure but also for stations where there is splitting and combining involved. Constraints (4.3.28) to (4.3.31) define the variable domains.

In some cases, it takes very long for the subproblem to find a positive optimal value but it gets close to positive relatively fast. To prevent this from happening, we stop cyclically solving the linear relaxation of the restricted master problem and the subproblem when the optimal value of the subproblem is strictly larger than \(-1 \times 10^{-6}\). Then the restricted master problem is solved as an integer problem. In the next section, we will apply this theory on a small case.
4.4 Solutions

The line model might change a basic assumption of NS, because often the line planning is not seriously changed. The case that we consider is small compared to the rail-network of the Netherlands but it gives an indication of the possibilities of the line model. This section contains information about the used data, the settings, PC and software. Results and conclusions can be found at the end of this section.

4.4.1 Data and Settings

The input of the line model can not be taken in a straight forward way from any NS dataset. Therefore, we manipulate existing datasets such that they can be used for the line model.

The basic data concerns the stations and the rails between these stations. This is the network for which the train lines have to be defined. Also the type (R or IC) of the station is important because it determines which types of train stop at the station. So, each train line also has a type related to where it stops on its path. A train line of type \( x \) stops at all the stations of type \( y \leq x \). We use the current types of the stations for the line model which are the regional type (R) or type 2 and the inter-city type (IC) or type 1.

In Section 4.3, we already mentioned the role of the OD-matrix. A disadvantage of calculating with the OD-matrix is the confidentiality factor of it. The OD-matrix is sensitive information because it is used to determine, among other things, the prices of the train tickets. We were only allowed to work with a dummy version of the OD-matrix. This dummy version contains some disturbances. It is not known to us where these disturbances are and how many there are. The results of our calculations are therefore not entirely realistic and we can not determine how far off we are. The dummy matrix contains the number of passengers that travel from one station to another, during one day. This means that we need to spread the total number of passengers over the day. In reality, we need to take the rush hours into account but, since we can only guess this distribution, we divide the passengers equally over the day. A day starts at 5 am and ends at 2 am, which means that we divide the numbers in the OD-matrix by 21.

With the knowledge of the stations, their type, the rails and the OD-matrix, we construct the type graph, define the type of each edge, and determine the number of passengers that use the same edge and the number of passengers that need a certain connection. We start by constructing the shortest path between every two stations. Therefore we use Dijkstra’s algorithm. Consider a graph with \( n \) nodes, then Dijkstra’s algorithm determines the costs of a node \( s \) to all the other nodes. By calling the algorithm \( n \) times, each time with another source node, we can determine the costs between every two stations. The cost for us, is the distance between the stations. In algorithm 1 we use \( V \) for the set of nodes and \( E \) for the set of edges. Node \( s \) is the source node. The variable \( l(i) \) indicates the cost of node \( s \) to node \( i \). Constant \( c_{i,j} \) is the cost of the edge between nodes \( i \) and \( j \). Variable \( \pi(j) \) determines the previous node of \( j \) on the shortest path to \( s \).

Algorithm 1 (p.11, [16]). Dijkstra.

Step 1: Initialization

Assume \( Q = \{s\} \), assume \( l(i) = \infty \) for all \( i \in V \setminus \{s\} \) and \( l(s) = 0 \).

Step 2: Node selection

Let \( i \in Q \) be a node for which \( l(i) = \min\{l(j) : j \in Q\} \). Node \( i \) is removed from \( Q \). The label of node \( i \) is now permanent.

Step 3: Label update
For all \((i,j) \in E\) applies: \(l(j) \) becomes \(\min\{l(j), l(i) + c_{i,j}\}\). If \(l(j)\) is adjusted in this step then \(\pi(j) = i\) and \(j\) is added to \(Q\), unless \(j\) is already an element of \(Q\).

**Step 4: stop condition**

The algorithm stops if \(Q\) is empty, If \(Q\) is not empty, then go to step 2.

Dijkstra’s algorithm is not the fastest shortest path algorithm. Article [16] gives an overview of some faster algorithms. We did not use them because finding all the shortest paths is just an initialization step of the line model. The solution time of Dijkstra’s algorithm is not included in the solution time of the line model.

The type graph is a derivate from the rail network. Based on the rail network without the unused edges, we construct all the possible paths of each type. If an edge occurs multiple times, then only one of each type remains. If there are identical edges each of a different type, then only the highest types remains. This is a consequence of the assumption that passengers use the highest type of train line.

Every edge in the type graph has a certain type (R, IC) depending on the stations it connects. We can also find the total number of passengers that use a certain edge of the type graph, since the shortest path between any two stations is known and the OD-matrix gives us the number of passengers that travel between these stations. The same yields for the connections, if two edges in the type graph are connected by a station, then we know exactly how many passengers will travel from one edge to the other. Therefore we can determine the number of bad connections later on.

The network of our case only contains rails and stations that are used by NS and not by any other transportation company. Although the OD-matrix also includes the numbers of passengers traveling from or to a station not used by NS, the dummy version does not contain this information. This is to protect the other transportation companies. The rail network of our case is depicted in Table 4.4.

![Rail Network Diagram](image)

**Table 4.4: The rail network of the case.**

We assign the types of the stations the same way as they are assigned currently. This means all the stations in Table 4.4 are type 2 stations except Rm, Std, Mt and Hrl. These four stations are type 1 stations. The next step is to construct the type graph. The result is shown in Table 4.5.
4.4. Solutions

Table 4.5: The type graph of the case.

There are only two types of stations so there are only two types of edges in the type graph. For each undirected edge in Table 4.5 there are in reality two edges in the type graph. One for each direction.

What is left to decide is the frequency of the edges of the type graph. For simplicity we assume that the frequency $f_i$ is equal to 1 for all edges $i$ of the type graph. Therefore, each edge should be part of at least one path.

Another topic that is not yet discussed is the settings for the available materials. We provide three types of materials namely AD and OA units, like for the previous models, and RR units. An RR unit is used for type 2 train lines. It consists of two carriages and it has 24 first class seats and 136 second class seats. We make no distinction between first class and second class since the OD-matrix does not specify this difference. There is only one restriction for the combinations namely a combination may consist of no more than 15 units. This gives us the opportunity to predict how many units should be available. The inventory is set at 50 AD, 50 OA and 200 RR. This is more than needed but we do not want a restriction due to the inventory, even though it is possible. After the calculations have been done, we can determine how many units were actually needed.

Finally, we need to provide an initial solution for the restricted master problem. This initial solution is standard an expensive solution since we have to provide it. We are not going to search for the cheapest solution since the line model is going to do that. The paths in our initial solution are all the edges of the type graph using the largest composition of the right type of material. This means that the frequencies are optimal (1) but carriage-kilometers and the bad connections are as high as they can get (for these frequencies).

With all the data described above, the line model can be solved. We just need to specify the properties of our PC and the software that is used to solve it.

4.4.2 PC and Software

We use the same PC and the line model is also implemented in IBM ILOG OPL IDE 6.3 and solved by IBM ILOG CPLEX 12.1.0. CPLEX is allowed to calculate on only one core because it is faster. The memory available for working storage is set to 2048.0 MB. No other standard settings are changed. Specifying an order for the decision variables is not useful for the line model.
4.4.3 Results and Conclusions

The results of the line model are a confirmation or a confrontation for NS. If the result selects the same train lines as NS currently uses, it is a confirmation and NS is sure that no extra profit can be made by changing the train lines. If the result is different from the current situation, NS has to confront itself with the differences and find out if it is an option to change the line planning taking all the practical restrictions into account. These practical restrictions exist since we used some simplifications.

Our case is just a small part of the national network and the OD-matrix is just a dummy version. In other words, the results and conclusions are solely based on our case and are not necessarily representative for the national network. By all means, the resulting train lines of type 2 are depicted below. The used combinations are indicated on the arrows where \( a \) stands for an AD unit, \( b \) stands for an OA unit and \( c \) stands for an RR unit.

The type 2 train lines are the same but in opposite directions. This kind of symmetry is not a constraint of the line model. Although it would be an interesting and realistic constraint, it is not simple to include in the model. The reason for this is that in each iteration step only one path is added to the restricted master problem. Therefore it is impossible to add a constraint to the restricted master problem that has to select for each path also the same path in the other direction since the path in the other direction might not yet be added to the restricted master problem.

These paths include some bad connections but have the lowest possible carriage-kilometers. We can conclude that, for the type 2 train lines, the carriage-kilometers are the most important factor of the objective function.

The following figures are the resulting train lines of type 1.
These type 1 train lines are clearly not symmetric. This is probably the result of the number of passengers on these lines. The consequence is that the bad connections are more important than the carriage-kilometers. The solution of the line model for type 1 does not have bad connections at all. On the other hand, it contains three times the total distance of the type 1 network instead of two times for the symmetric solution.

It strikes us that, for all the selected train lines (type 1 and type 2), the smallest combinations of its type was chosen. This could change if we distributed the numbers of the OD-matrix in another way over the day. We distributed the travelers equally over the day while rush hours obviously contain more travelers. As we mentioned before, our distribution is just an assumption and we have no further knowledge of the actual distribution. NS does have this information, so they are in the possibility to calculate a more realistic result.

Another characteristic that applies for both types is the coupling or uncoupling in case of a split or combination. If there is a split, a unit is always coupled while if there is a combination, a unit is uncoupled. In fact, this means that passengers still have to change seats. If we prohibit coupling and uncoupling at stations where there is a split or combination, we get another result. For the type 1 paths, the common part of the two legs gets an extra AD unit. For the type 2 paths, the paths change completely therefore some figures are given below.
The common part of the legs is no longer the path between Roermond and Sittard but between Maastricht Randwyck and Sittard. The bad connections change too. Before it was the connection Gln-Std-Lut while now it is Gln-Std-Srn. The solution is inventive but still not realistic since some travelers still have to change seats and in practice Geleen-Lutterade is not used to couple or uncouple.

The last solution is more expensive than the first one. This is possibly due to the bad connections but it is certainly due to the carriage-kilometers caused by the extra units. Both solutions are optimal i.e. the gap between the solution of the linear relaxation and the integer solution is zero. This is rather an exception than a rule.

In comparison to the current line planning, both new line plans are partly different. Whether it is smart to change the line plan is not clear at this point since the type 1 train lines do not stop at Roermond but continue up to Alkmaar. As this case suggests, there are probably some improvements for the line planning. Whether or not they are realistic depends on the strategy of NS.
Chapter 5

Extensions

All the models in this thesis are partly based on simplifications. The results are therefore not directly usable in practice but NS is interested in the effect of relaxing some restrictions. More specifically, those restrictions that are based on experience rather than on facts. By relaxing these restrictions we deviate further from reality. Removing an extra restriction or adding one changes the results, hence expansions of the models are an option for further research. In this chapter, we will give some other extensions for each of the models.

5.1 Combination Model

Adding or removing constraints can have a great impact on the results. Even the conclusions can change. We only discussed three main strategies. There are of course more of them like only allowing cutting at large stations or outside rush hours. We never looked at the effected passengers when we introduced a cut. This is also an option.

We also encountered that the solution time of the combination model is rather long. Even with two train lines, the calculation get out of memory. It might be interesting to change the model such that it can handle more train lines simultaneously.

In the end, we were looking at deviations from the never strategy to save units. We mentioned that some units are used for multiple lines. It would be a good extension to look into this some more. One could select a group of units that are solely used for a selection of train lines and calculate the number of units that are actually needed for that selection of train lines. Another option is accepting the schedule for the other train lines and reserving for a certain period units to do the extra activities for those train lines. This can be done by adding dummy train rides to the model while in fact the assigned combination is working on an other train line. We did not completely look into this.

5.2 Cancellation Model

For the cancellation model, we tested five cases that looked interesting to us. It is possible that there are other cases that give new insights. NS also has a lot of rules that are specific for certain train lines and stations. Therefore it might be useful to look into some of these cases in more detail.

An extension for the version of the cancellation model that we used, is introducing a priority order for some of the decision variable as we did for the combination model. This might reduce the solution time of the cancellation model.
5.3 Line Model

The line model is mainly based on column generation. This technique has many more options than the ones we used. For example, it is possible to add more than one path at the same time to the restricted master problem. This way, we could add the constraint that the solution should be symmetric. This means that for each path that we select also the path in the other direction should be selected.

In our case the solution was optimal. This means that the gap between the solution of the linear relaxation and the integer solution was zero. This is not always the case. If there is a gap, one can use a branch-and-bound algorithm to reduce the gap. More research on this topic is necessary to make the line model more usable for NS.

An expansion of the line model is to consider the hourly schedule of the trains when the solution is calculated. The hourly schedule orders the trains and makes it possible to keep track of the inventories at the stations. This is currently not included in the line model because we do not know when a train arrives at and leaves a station. Hence, we do not know when a unit is uncoupled or coupled and the inventory is a complete mystery. We only know the overall status of the inventory, this is the number of units that entered the inventory and the number of units that left it.

We can conclude that there is room for improvement for all three of the models. We leave it up to further research to make these improvements. A last remark is that minimizing railway rolling stock is not the only optimization problem that NS faces. Problems like scheduling crews for each train and rescheduling them when there is a delay or cancellation of a train, or planning maintenance tasks for the rolling stock, are also a concern of NS. There will always be new unforeseen circumstances that need a special solution.
Appendix A

Extended Combination Model

We start with some notation used especially for combined train lines:

- \( \sigma_1(t) \): first successor of trip \( t \) if trip \( t \) has more than one successor,
- \( \sigma_2(t) \): second successor of trip \( t \) if trip \( t \) has more than one successor,
- \( T_S \): the set of all trips that have two successors,
- \( T_C \): the set of all trips that will be combined at the end of the trip,
- \( K_1^t \): 1 if there is no cut between trip \( t \) and the first successor of trip \( t \), otherwise 0,
- \( K_2^t \): 1 if there is no cut between trip \( t \) and the second successor of trip \( t \), otherwise 0,
- \( Y_{1,t,b_1,b_2} \): 1 if combination \( b_1 \) is used for trip \( t \) and \( b_2 \) for \( \sigma_1(t) \), otherwise 0,
- \( Y_{2,t,b_1,b_2} \): 1 if combination \( b_1 \) is used for trip \( t \) and \( b_2 \) for \( \sigma_2(t) \), otherwise 0,
- \( R_{t,b,b_1,b_2}^S \): 1 if \( Y_{1,t,b_1,b_2} \) and \( Y_{2,t,b_1,b_2} \) are equal to 1,
- \( R_{\sigma_1(t),b,b_1,b_2}^C \): 1 if \( Y_{t_1,b_1,b} \) and \( Y_{t_2,b_2,b} \) are equal to 1,

All the other notations that occur in the extended combination model have the same definition as in Chapter 2. Now, the model can be formulated:

Minimize

\[
w_1 \sum_{t \in T} \sum_{m \in M} d_{t,m} N_{t,m} + w_2 \sum_{t \in T} \sum_{b \in B} d_{t,s,b} X_{t,b} + w_3 \sum_{t \in T \setminus T_\infty} D_t, \tag{A.0.1}
\]
∀∈T \cap (T_w \cup T_c \cup T_2), \forall b \in B_t:
\begin{align*}
D_t & \geq (1 - K_1) + X_{t,b} + X_{\sigma(t),b} - 2, \\
∀∈T_3 \cap T_2, (b \in B_t, b_1 \in B_{\sigma_1(t)}, b_2 \in B_{\sigma_2(t)}): (\forall m \in M: \nu(b)_m - \nu(b_1)_m - \nu(b_2)_m = 0): \\
D_t & \geq (1 - R_{t,b_1,b_2}^S) + X_{t,b} + X_{\sigma_1(t),b_1} + X_{\sigma_2(t),b_2} - 3,
\end{align*}
(A.0.2)

∀∈T_3 \cap T_2, (b \in B_t, b_1 \in B_{\sigma_1(t)}): (\forall m \in M: \nu(b)_m - \nu(b_1)_m = 0): 
\begin{align*}
D_t & \geq (1 - K_1^t) + (1 - K_2^t) + X_{t,b} + X_{\sigma_1(t),b_1} - 3, \\
∀∈T_3 \cap T_2, (b \in B_t, b_2 \in B_{\sigma_2(t)}): (\forall m \in M: \nu(b)_m - \nu(b_2)_m = 0): \\
D_t & \geq (1 - K_1^t) + (1 - K_2^t) + X_{t,b} + X_{\sigma_2(t),b_2} - 3,
\end{align*}
(A.0.3)

∀∈T_3 \cap T_2, (b \in B_t, b_1 \in B_{\sigma_1(t)}): (\forall m \in M: \nu(b)_m = 0):
\begin{align*}
D_t & \geq (1 - K_1^t) + (1 - K_2^t) + X_{t,b} + X_{\sigma_1(t),b_1} - 3, \\
∀∈T_3 \cap T_2, (b \in B_t, b_2 \in B_{\sigma_2(t)}): (\forall m \in M: \nu(b)_m - \nu(b_2)_m = 0):
\end{align*}
(A.0.4)

∀∈T_3 \cap T_2, (b \in B_t, b_1 \in B_{\sigma_1(t)}): (\forall m \in M: \nu(b)_m = 0):
\begin{align*}
D_t & \geq (1 - K_1^t) + (1 - K_2^t) + X_{t,b} + X_{\sigma_1(t),b_1} - 3, \\
∀∈T_3 \cap T_2, (b \in B_t, b_2 \in B_{\sigma_2(t)}): (\forall m \in M: \nu(b)_m - \nu(b_2)_m = 0):
\end{align*}
(A.0.5)

Cutting train lines
∀∈T:
\sum_{b \in B_t} X_{t,b} = 1,
\sum_{b \in B_t, b_1 \in B_{\sigma_1(t)}, b_2 \in B_{\sigma_2(t)}} Y_{t,b_1,b_2} = K_1,
\sum_{b \in B_t, b_1 \in B_{\sigma_1(t)}} Y_{t,b}^1 = K_1^t,
\sum_{b \in B_t, b_2 \in B_{\sigma_2(t)}} Y_{t,b}^2 = K_2^t,
X_{t,b_1} \geq \sum_{b_2 \in B_{\sigma_1(t)}} Y_{t,b_1,b_2},
X_{t,b_1} \geq \sum_{b_2 \in B_{\sigma_2(t)}} Y_{t,b_1,b_2},
X_{\sigma(t),b_2} \geq \sum_{b_1 \in B_t} Y_{t,b_1,b_2},
(A.0.9)
(A.0.10)
(A.0.11)
(A.0.12)
(A.0.13)
(A.0.14)
(A.0.15)
(A.0.16)
\[ \forall t \in T_0 \setminus T_\infty, b_1, b_2 \in B_{\sigma_1(t)}: \]
\[ X_{\sigma_1(t), b_2} \geq \sum_{b_1 \in B_t} Y_{t, b_1, b_2}^1, \quad (A.0.17) \]

\[ \forall t \in T_0 \setminus T_\infty, b_1, b_2 \in B_{\sigma_2(t)}: \]
\[ X_{\sigma_2(t), b_2} \geq \sum_{b_1 \in B_t} Y_{t, b_1, b_2}^2, \quad (A.0.18) \]

**Allowed combinations**

\[ \forall t \in T \setminus (T_\infty \cup T_0 \cup T_C), b_1, b_2 \in B_{\sigma(t)}: \]
\[ Y_{t, b_1, b_2} \leq \alpha_{b_1, b_2}, \quad (A.0.19) \]

\[ \forall t \in T_0 \setminus T_\infty, b_1, b_2 \in B_{\sigma_1(t)}: \]
\[ Y_{t, b_1, b_2}^1 \leq K_t^2 + \alpha_{b_1, b_2}, \quad (A.0.20) \]

\[ \forall t \in T_0 \setminus T_\infty, b_1, b_2 \in B_{\sigma_2(t)}: \]
\[ Y_{t, b_1, b_2}^2 \leq K_t^1 + \alpha_{b_1, b_2}, \quad (A.0.21) \]

\[ \forall (t_1, t_2) \in T_C \setminus T_\infty, b_1, b_2 \in B_{\varphi(t_1); \sigma(t_1) = \sigma(t_2)}: \]
\[ Y_{t_1, b_1, b_2} \leq K_{t_2} + \alpha_{b_1, b_2}, \quad (A.0.22) \]

\[ \forall (t_1, t_2) \in T_C \setminus T_\infty, b_1, b_2 \in B_{\varphi(t_2); \sigma(t_1) = \sigma(t_2)}: \]
\[ Y_{t_2, b_1, b_2} \leq K_{t_1} + \alpha_{b_1, b_2}, \quad (A.0.23) \]

\[ \forall t \in T_0 \setminus T_\infty, \{ m_1, m_2 \in M, b_1, b_2 \in B_t, b_3 \in B_{\sigma_2(t)}; \nu(b_2) = m_1 + \nu(b_3) > \nu(b_1) \land \nu(b_2) = m_2 + \nu(b_3) > \nu(b_1) \}: \]
\[ R_{t, b_1, b_2, b_3}^S = 0, \quad (A.0.24) \]

\[ \forall (t_1, t_2) \in T_C \setminus T_\infty, \{ m_1, m_2 \in M, b_1, b_2 \in B_t, b_3 \in B_{\sigma(t_1); \sigma(t_1) = \sigma(t_2)}; \nu(b_2) + \nu(b_3) > \nu(b_1) \land \nu(b_2) = m_2 + \nu(b_3) > \nu(b_1) \}: \]
\[ R_{t_1, b_1, b_2, b_3}^C = 0, \quad (A.0.25) \]

\[ \forall t \in T \setminus T_\infty, \{ m \in M; \sum_{b \in B_t} \nu(b) m X_{t, b} \}: \]
\[ N_{t, m} = \sum_{b \in B_t} \nu(b) m X_{t, b}, \quad (A.0.28) \]

\[ \forall t \in T_0, m \in M: \]
\[ C_{t, m} = N_{t, m}, \quad (A.0.29) \]

\[ \forall t \in T_\infty, m \in M: \]
\[ U_{t, m} = N_{t, m}, \quad (A.0.30) \]

\[ \forall t \in T \setminus (T_\infty \cup T_C \cup T_S), m \in M: \]
\[ C_{\sigma(t),m} = \sum_{b_1 \in \mathcal{B}_2(t)} (\nu(b_1) - \nu(b_2)) Y_{t,b_1,b_2} + \sum_{b_2 \in \mathcal{B}_2(t)} \nu(b_2_m) \left( X_{\sigma(t),b_2} - \sum_{b_1 \in \mathcal{B}_{1(t)}} Y_{t,b_1,b_2} \right), \]

(A.0.31)

\[ \forall t \in \mathcal{T}_I(\mathcal{T}_{\omega(t)} \cup \mathcal{T}_I), m \in \mathcal{M}: \]

\[ U_{t,m} = \sum_{b_1 \in \mathcal{B}_2(t), b_2 \in \mathcal{B}_2(t)} (\nu(b_1) - \nu(b_2)) Y_{t,b_1,b_2} + \sum_{b_2 \in \mathcal{B}_2(t)} \nu(b_2_m) \left( X_{t,b_1} - \sum_{b_1 \in \mathcal{B}_{1(t)}} Y_{t,b_1,b_2} \right), \]

(A.0.32)

\[ \forall t \in \mathcal{T}_I(\mathcal{T}_{\omega(t)} \cup \mathcal{T}_I), m \in \mathcal{M}: \]

\[ C_{\sigma_1(t),m} = \sum_{b_1 \in \mathcal{B}_{1(t)}} \nu(b_1) m \left( X_{\sigma_1(t),b_1} - \sum_{b \in \mathcal{B}_1} Y_{t,b,b_1}^1 \right) + \sum_{b_2 \in \mathcal{B}_{2(t)}} \nu(b_2) m \left( X_{\sigma_2(t),b_2} - \sum_{b \in \mathcal{B}_1} Y_{t,b,b_2}^2 \right) + \sum_{b \in \mathcal{B}_2(t)} \nu(b_2_m) \left( Y_{t,b,b_2}^2 - \sum_{b_3 \in \mathcal{B}_{1(t)}} R_{t,b,b_3,b_2}^S \right) + \sum_{b \in \mathcal{B}_2(t)} \nu(b_2_m) \left( Y_{t,b,b_2}^1 - \sum_{b_3 \in \mathcal{B}_{2(t)}} R_{t,b,b_3,b_2}^S \right), \]

(A.0.33)

\[ \forall t \in \mathcal{T}_I(\mathcal{T}_{\omega(t)} \cup \mathcal{T}_I), m \in \mathcal{M}: \]

\[ C_{\sigma_2(t),m} = \sum_{b \in \mathcal{B}_2(t)} \nu(b_m) \left( X_{t,b} + \nu(b_2_m) - \nu(b_2) \right) R_{t,b,b_1,b_2}, \]

(A.0.34)

\[ \forall t \in \mathcal{T}_I(\mathcal{T}_{\omega(t)} \cup \mathcal{T}_I), m \in \mathcal{M}: \]

\[ U_{t,m} = \sum_{b \in \mathcal{B}_2(t), b_1 \in \mathcal{B}_{1(t)}, b_2 \in \mathcal{B}_{2(t)}} (\nu(b_m) - \nu(b_1)) R_{t,b,b_1,b_2} + \sum_{b \in \mathcal{B}_2(t)} \nu(b_m) \left( Y_{t,b,b_2}^1 - \sum_{b_2 \in \mathcal{B}_{2(t)}} R_{t,b,b_1,b_2}^S \right) + \sum_{b \in \mathcal{B}_2(t)} \nu(b_m) \left( Y_{t,b,b_2}^2 - \sum_{b_1 \in \mathcal{B}_{1(t)}} R_{t,b,b_1,b_2}^S \right) + \sum_{b \in \mathcal{B}_2(t)} \nu(b_m) \left( X_{t,b} - \sum_{b_1 \in \mathcal{B}_{1(t)}} Y_{t,b,b_1}^1 - \sum_{b_2 \in \mathcal{B}_{2(t)}} Y_{t,b,b_2}^2 + \sum_{b_1 \in \mathcal{B}_{1(t)}, b_2 \in \mathcal{B}_{2(t)}} R_{t,b,b_1,b_2}^S \right), \]

(A.0.35)

\[ \forall m \in \mathcal{M}, (t_1, t_2) \in \mathcal{T}_I(\mathcal{T}_{\omega(t_1)} = \sigma(t_2)): \]
\[ C_{\sigma(t_1),m} = \sum_{b \in B_{\sigma(t_1)}, \nu(b)m \in T_1} (\nu(b)m - \nu(b_1)m - \nu(b_2)m) R_{\sigma(t_1),b_1,b_2}^C \]
\[ + \sum_{b \in B_{\sigma(t_1)}, \nu(b_1)m < \nu(b)m} (\nu(b)m - \nu(b_1)m) \left( Y_{t_1,b_1,b} - \sum_{b_2 \in B_2} R_{\sigma(t_1),b_1,b_2}^C \right) \]
\[ + \sum_{b \in B_{\sigma(t_1)}, \nu(b)m < \nu(b_2)m} (\nu(b)m - \nu(b_2)m) \left( Y_{t_2,b_2,b} - \sum_{b_1 \in B_1} R_{\sigma(t_1),b_1,b_2}^C \right) \]
\[ + \sum_{b \in B_{\sigma(t_1)}} \nu(b)m \left( X_{\sigma(t_1),b} - \sum_{b_1 \in B_1} Y_{t_1,b_1,b} - \sum_{b_2 \in B_2} Y_{t_2,b_2,b} \right) \]
\[ + \sum_{b_1 \in B_1, b_2 \in B_2} R_{\sigma(t_1),b_1,b_2}^C \right) \; \right) \]
\[ \forall m \in M, (t_1,t_2) \in T_C \setminus \{ t_\infty \}; \sigma(t_1)=\sigma(t_2) : \]
\[ U_{t_1,m} = \sum_{b \in B_{\sigma(t_1)}, \nu(b)m \in T_1} (\nu(b_1)m - \nu(b)m) \left( Y_{t_1,b_1,b} - \sum_{b_2 \in B_2} R_{\sigma(t_1),b_1,b_2}^C \right) \]
\[ + \sum_{b \in B_{\sigma(t_1)}, \nu(b_2)m > \nu(b)m} (\nu(b_2)m - \nu(b)m) \left( Y_{t_2,b_2,b} - \sum_{b_1 \in B_1} R_{\sigma(t_1),b_1,b_2}^C \right) \]
\[ + \sum_{b_1 \in B_1} \nu(b_1)m \left( X_{t_1,b_1} - \sum_{b \in B_{\sigma(t_1)}} Y_{t_1,b_1,b} \right) \]
\[ + \sum_{b_2 \in B_2} \nu(b_2)m \left( X_{t_2,b_2} - \sum_{b \in B_{\sigma(t_2)}} Y_{t_2,b_2,b} \right) \; \right) \]
\[ \forall m \in M, (t_1,t_2) \in T_C \setminus \{ t_\infty \}; \sigma(t_1)=\sigma(t_2) : \]
\[ U_{t_2,m} = \sum_{b \in B_{\sigma(t_1)}, \nu(b_1)m + \nu(b_2)m > \nu(b)m} \nu(b)m R_{\sigma(t_1),b_1,b_2}^C \; \right) \]
\[ \forall t_1 \in T, m \in M: \]
\[ I_{t_1,m} = I_{s_d(t_1),m}^0 - \sum_{t_2 \in T: s_d(t_2)=s_d(t_1)} C_{t_2,m} - \sum_{t_2 \in T: \tau_d(t_2) \leq \tau_d(t_1)} U_{t_2,m} \; \right) \]
\[ \forall s \in S, m \in M: \]
\[ I_{s,m}^\infty = I_{s,m}^0 - \sum_{t \in T: s_d(t)=s} C_{t,m} + \sum_{t \in T: \tau_d(t) \leq \tau_d(t_1)} U_{t,m} \; \right) \]
\[ \forall m \in M: \]
\[ n_m = \sum_{s \in S} I_{s,m}^0 \; \right) \]
∀s ∈ S, m ∈ M:
\[ I_{s,m}^0 = I_{s,m}^\infty, \]  
\hspace{1cm} (A.0.42)

Variable domains

∀t ∈ T, b ∈ B:
\[ X_{t,b} \in \{0, 1\}, \]  
\hspace{1cm} (A.0.43)

∀t ∈ T \setminus (T_S \cup T_\infty), b_1 \in B_1, b_2 \in B_{e(t)}:
\[ Y_{t,b_1,b_2} \in \{0, 1\}, \]  
\hspace{1cm} (A.0.44)

∀t ∈ T_S \setminus (T_\infty), b_1 \in B_1, b_2 \in B_{e_1(t)}:
\[ Y_{t,b_1,b_2}^1 \in \{0, 1\}, \]  
\hspace{1cm} (A.0.45)

∀t ∈ T_S \setminus (T_\infty), b_1 \in B_1, b_2 \in B_{e_2(t)}:
\[ Y_{t,b_1,b_2}^2 \in \{0, 1\}, \]  
\hspace{1cm} (A.0.46)

∀t ∈ T:
\[ K_t \in \{0, 1\}, \]  
\hspace{1cm} (A.0.47)

∀t ∈ T_S:
\[ K^1_t \in \{0, 1\}, \]  
\hspace{1cm} (A.0.48)

∀t ∈ T_S:
\[ K^2_t \in \{0, 1\}, \]  
\hspace{1cm} (A.0.49)

∀t ∈ T_S \setminus (T_\infty), b_1 \in B_{e_1(t)}, b_2 \in B_{e_2(t)}:
\[ R^S_{t,b_1,b_2} \in \{0, 1\}, \]  
\hspace{1cm} (A.0.50)

∀t_1, t_2 ∈ T \setminus (T_\infty), b_1 \in B_{e(t_1)}, b_1 \in B_{e_1(t_1)}:
\[ R^C_{t_1,t_2,b_1,b_2} \in \{0, 1\}, \]  
\hspace{1cm} (A.0.51)

∀t ∈ T, m ∈ M:
\[ N_{t,m}, C_{t,m}, U_{t,m}, I_{t,m} \in \mathbb{R}_+, \]  
\hspace{1cm} (A.0.52)

∀s ∈ S, m ∈ M:
\[ I_{s,m}^0, I_{s,m}^\infty \in \mathbb{R}_+, \]  
\hspace{1cm} (A.0.53)

∀t ∈ T \setminus T_\infty:
\[ D_t \in \mathbb{R}_+. \]  
\hspace{1cm} (A.0.54)

We used this model to find the results of the 3000 line and the 800 line, stated in Chapter 2. To give an indication of the size of the model for these lines, we collected some numbers in Table [A.1]. This table contains the number of constraints, variables and non-zero coefficients. The number of variables is divided into binary variables and others. We not only look at the numbers of the original MIP, i.e. the MIP such as we defined it, but we also look at the numbers of the reduced MIP used by the solver.
<table>
<thead>
<tr>
<th>Weight</th>
<th>Strategy</th>
<th>Line</th>
<th>Original MIP</th>
<th>Reduced MIP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Constraints</td>
<td>Variables</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Binary</td>
<td>Other</td>
</tr>
<tr>
<td>(1,1,1)</td>
<td></td>
<td>800</td>
<td>105490</td>
<td>70494</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3000</td>
<td>28661</td>
<td>37522</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Both</td>
<td>134079</td>
<td>107971</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>800</td>
<td>145883</td>
<td>70494</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3000</td>
<td>60071</td>
<td>37522</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Both</td>
<td>205882</td>
<td>107971</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>800</td>
<td>106243</td>
<td>70494</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3000</td>
<td>29175</td>
<td>37522</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Both</td>
<td>135346</td>
<td>107971</td>
</tr>
<tr>
<td></td>
<td>Never</td>
<td>800</td>
<td>105492</td>
<td>70494</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3000</td>
<td>28663</td>
<td>37522</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Both</td>
<td>134081</td>
<td>107971</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>800</td>
<td>145885</td>
<td>70494</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3000</td>
<td>60073</td>
<td>37522</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Both</td>
<td>205884</td>
<td>107971</td>
</tr>
<tr>
<td></td>
<td>Limited</td>
<td>800</td>
<td>106245</td>
<td>70494</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3000</td>
<td>29177</td>
<td>37522</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Both</td>
<td>135348</td>
<td>107971</td>
</tr>
</tbody>
</table>

Table A.1: Size of the extended combination model.
Appendix B

Schedules

In this appendix, we give an overview of the schedules that go along with some of the previously stated results. We could have included more but this selection gives the most important information.

The first section contains the current schedules for the 3000 line and the 800 line. These schedules are a point of reference for all the other schedules. The second section includes a few schedules of the combination model. Specifically, the resulting schedules for the calculations of both train lines at the same time and the resulting schedules for the calculations of the never strategy with each of the extra constraints of both lines. The third section contains the schedules for each of the cases of the cancellation model.

The following figures indicate each train unit that moves from one station to another by a blue or red line. A red line stands for an AD unit and a blue line stands for an OA unit. We can also find the time it takes to make each connection at the top of the schedule. When units are added or removed from a combination, this is indicated by means of a circle. Except for Figure B.3 to B.8 the circle indicates places where there is a cut in the train ride. In these schedules, we do not mark the stations if there is a change of combination since it would make the schedule unclear.
B.1 Current Schedules

Figure B.1: Current schedule for the 800 line.
Figure B.2: Current schedule for the 3000 line.
B.2 Schedules of the Combination Model

Figure B.3: Result of the random strategy with minimum SKM for the 800 line
Figure B.4: Result of the random strategy with minimum SKM for the 3000 line
Figure B.5: Result of the limited strategy with minimum SKM for the 800 line
Figure B.6: Result of the limited strategy with minimum SKM for the 3000 line
Figure B.7: Result of the never strategy with minimum SKM for the 800 line
Figure B.8: Result of the never strategy with minimum SKM for the 3000 line
Figure B.9: Result of the never strategy with constraint 1 for the 800 line
Figure B.10: Result of the never strategy with constraint 1 for the 3000 line
Figure B.11: Result of the never strategy with constraint 2 for the 800 line
Figure B.12: Result of the never strategy with constraint 2 for the 3000 line
Figure B.13: Result of the never strategy with constraint 3 for the 800 line
Figure B.14: Result of the never strategy with constraint 3 for the 3000 line
B.3 Schedules of the Cancellation Model

Figure B.15: Result of case I.
Figure B.16: Result of case II.
Figure B.17: Result of case III for the 800 train line.
Figure B.18: Result of case III for the 3000 train line.
Figure B.19: Result of case IV for the 800 train line.
Figure B.20: Result of case IV for the 3000 train line.
Figure B.21: Result of case V for the 800 train line.
Figure B.22: Result of case V for the 3000 train line.
Bibliography


