DTU Management
Department of Technology, Management and Economics

Optimization of train cleaning for DSB using integer programming

Bachelors project

Optimization of train cleaning for DSB using integer programming
Bachelor subtitle

Bachelors project
June, 2023

By
Sif Egelund Christensen & Emma Victoria Lind

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Technology, Management and Economics, Akademivej, Building 358, 2800 Kgs. Lyngby Denmark
www.man.dtu.dk
ISSN: [0000-0000] (electronic version)
ISBN: [000-00-0000-000-0] (electronic version)

ISSN: [0000-0000] (printed version)
ISBN: [000-00-0000-000-0] (printed version)

Approval

This project has been prepared over 18 weeks at the Department of Technology, Management and Economics, at the Technical University of Denmark, DTU, in partial fulfilment for the degree Bachelor of Science in Engineering, BSc Eng.

It is assumed that the reader has a basic knowledge in the areas of operational analysis.

Sif Egelund Christensen & Emma Victoria Lind - s204208 & s191159

Sif Emma Lind

Signature

08/06-23

Date

Abstract

The intention of this report is to model a cleaning schedule based on a data set representing the train schedule of a week and the current cleaning schedule used by DSB.

Firstly, a data analysis of the provided cleaning schedule is made, where the data is described and some cleaning standards are determined based on this. These standards are set as goals for the model. The model is made to minimize the cost of cleaning whilst upholding these standards. The problem has been modelled and implemented using integer programming. In this process, three models have been constructed. The basic model, model 1, creates a cleaning schedule deciding when trains are to be cleaned. Model 2 is extended to ensure that the trains are able to be joined, travel together and then separate again and how they are to be cleaned while they are joined. Model 3 includes an implementation of two tendencies from the data analysis. This ensures that model 3 constructs a cleaning schedule with a comparable ratio between the two different cleaning types. Model 3 provides a solution with 39 % cheaper cleaning than the solution provided by DSB, which still upholds the cleaning standards derived. It is discussed that the difference may come from different priorities in how the solution is constructed or from the automation of the process.

The report shows that it is possible to make an automatic and optimal way to plan a cleaning schedule for DSB.

Contents

1 Introduction

DSB, Danske statsbaner (Danish state railways), is an independent public corporation owned by the Danish state. It operates a variety of passenger trains in all of Denmark and has more than 195 million passengers every year [1]. This constitutes a big part of the danish public transport system. Since so many passengers travel by public transportation in Denmark, it is natural that the trains must be cleaned so they can maintain a certain level of cleanliness. This has to comply with the train schedule, making sure enough time is available between train rides, for the train to be cleaned according to necessity. The planning of both train routes, schedule and making sure cleaning happens when needed and possible, is a bigger task. DSB plans the cleaning schedule based on the train schedules, and this creates obstacles in making sure trains are approximately equally clean at all times, while still getting the cleaning to the lowest cost possible. This report will try to model the process of scheduling the cleaning and thereby automatize and optimize the way to schedule the train cleaning for DSB. The work in this project will be based on integer programming.

2 Problem description

This project is based on a data set describing a train schedule in a ”normal week” in 2023 of the train network operated by DSB. The train network can be seen in Figure 2.1. A ”normal week” refers to a standard week from Monday to Sunday and the train schedule for this one week can therefore later be used for every other week of the year.

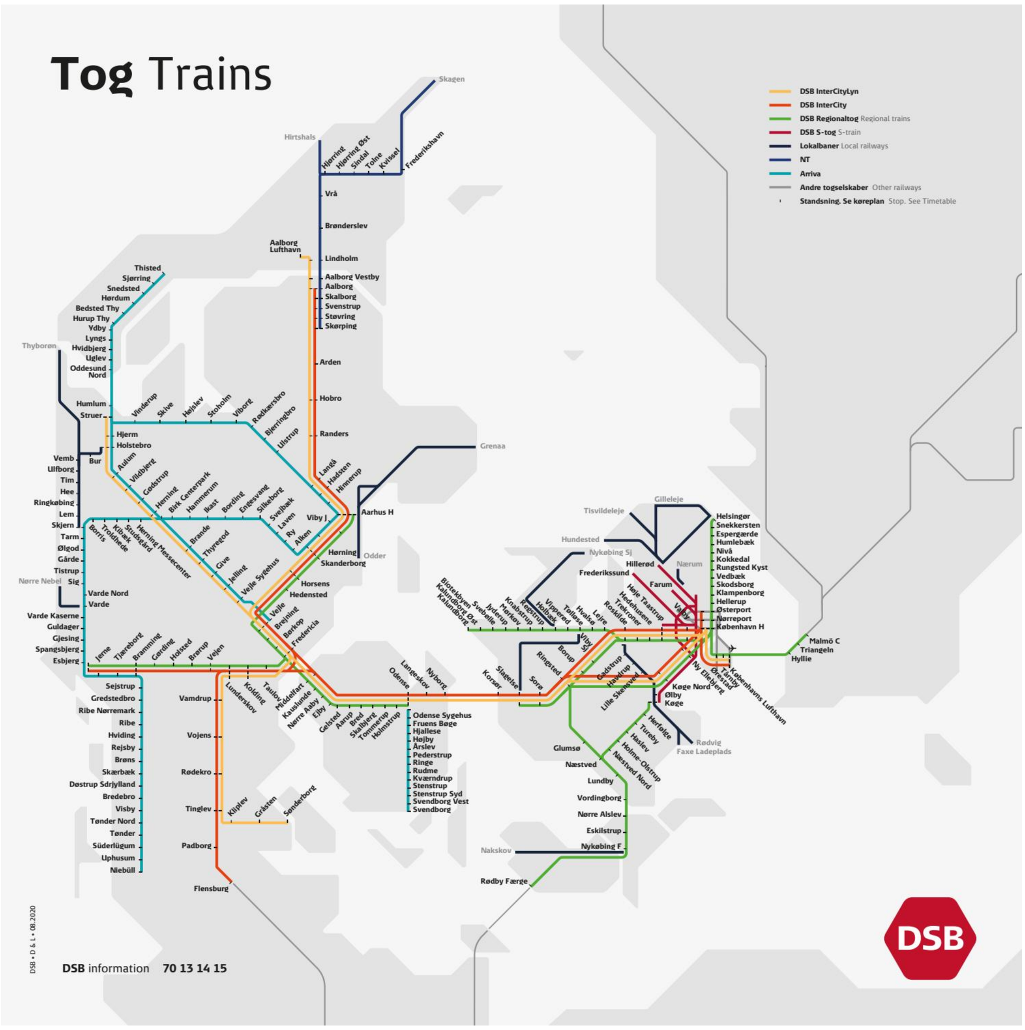


Figure 2.1: Train network operated by DSB [2].

During a week, a train performs many train rides. A train ride is represented by a start and end station and these will be called the departure and arrival station of the ride in this report. It is possible that there are multiple stops between the departure and the arrival station for a train ride, but these will not be taken into consideration. A stop is defined as every end station that the train passes through during the week. A stop is therefore both an arrival station for one ride and a departure station for the next train ride.

It is possible for several trains to be joined at a stop, travel together connected for one or several rides and then separate again. This is used to accommodate varying demands on train rides.

Each train that operates has a litra type. A litra type describes what kind of train it is, e.g. a passenger train, locomotive, freight train, motor driven, electric etc. There are 8 operating litra types in the train schedule: ERF, ETS, ICA, MGA, EB, ABS, B and BK. In Figure 2.2 the litra types are visualized.

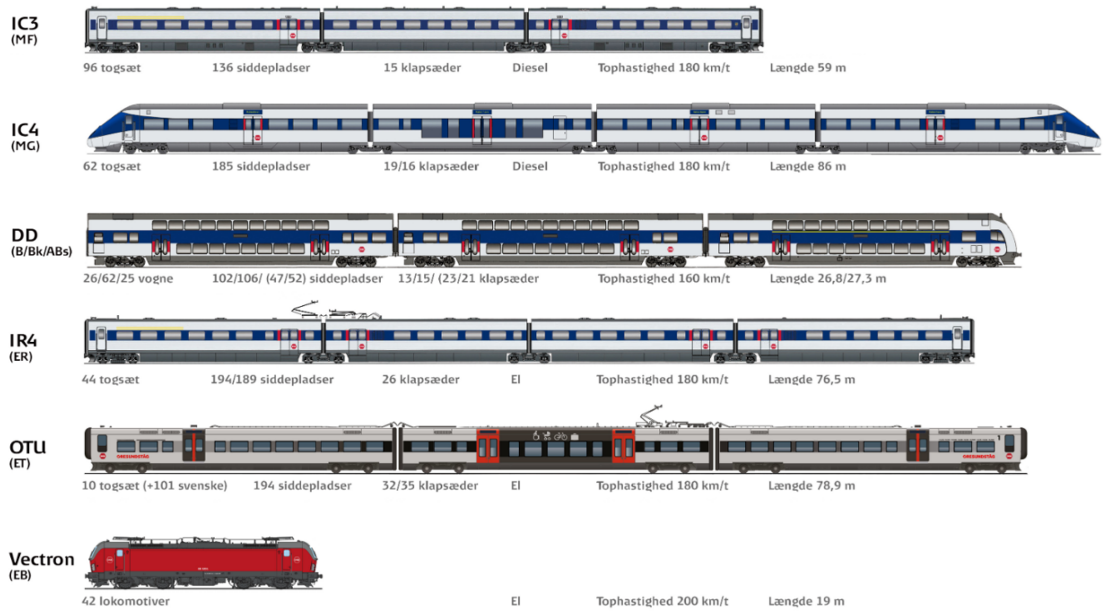


Figure 2.2: Litra types that are in operation [3]. The litra types matches as follows from the top: IC3 = ICA, IC4 = MGA, DD = {ABS, B, BK}, IR4 = ERF, OTU = ETS, Vectron = EB.

The trains need to be cleaned regularly between train rides. Cleaning the trains take up time and money, and DSB is interested in maintaining a cleaning standard of their trains at the lowest possible cost. The cleaning standard is not precisely defined, but DSB is interested in constructing a cleaning schedule that ensures clean trains at a minimal price.

DSB have outsourced the cleaning of their trains to a company providing cleaning crews. The exterior cleaning of the trains is not handled in this report. There are two types of interior cleanings that needs to be scheduled, called OR- and TR-cleanings.

- OR-cleanings: OR, stands for ”OpRydning” (Tidying), is a smaller cleaning, where trash is collected and bathrooms are cleaned.
- TR-cleanings: TR, stands for ”TrafikRengøring” (traffic cleaning), is a thorough cleaning of the entire train where windows and glass doors are polished, seats and carpets are vacuumed and ceilings, floor and furniture are washed, in addition to what is done in an OR-cleaning.

The two types of cleanings does not require the same amount of time to perform. The time it takes to perform a TR-cleaning or an OR-cleaning is different for each litra type but overall it is known that a TR-cleaning takes longer than an OR-cleaning to perform. Nor is there a clear ratio between the two cleanings, this depends on the individual litra type. The cleaning time can be seen in Table 2.1 and it shows that trains with litra type EB have no cleaning time. This is because they are locomotives and does not travel with passengers and thus does not need cleaning.

Table 2.1: Litra types and the corresponding cleaning time of each type of cleaning.

Litra type	ERF	ETS	ICA	MGA	EB	ABS	B	BK
TR-cleaning time [min]	80	104	72	109	-	175	175	175
OR-cleaning time [min]	20	15	16	20	-	47	47	47

The cleaning company does not have cleaning crews at all stations, thus it is not possible to clean the train at every stop. The stations where cleaning can be done can be seen in Table 2.2. A cleaning can be determined to happen on an arrival station of a train ride, if and only if the train has enough time on the current station before departing for the next trip.

Table 2.2: Stations where cleanings can be performed.

Station	Abbreviation
Aalborg	AB
Aarhus H	AR
Cph. Airport	CPH
Esbjerg	ES
Fredericia	FA
Frederikshavn	FH
Flensborg	FLB
Helsingør	HG
Helgoland	HGL
Kalundborg	KB
København H	KH
Østerport	KK
Lindholm	LIH
Nykøbing F	NF
Næstved	NÆ
Odense	OD
Søndenborg	SDB
Struer	STR
Tinglev	TE

When two trains are joined and travel together, it is important that the trains are cleaned in the same way. Meaning connected trains will need to be cleaned either together and by the same cleaning type, or not at all, while connected. So if one train is cleaned, all trains connected to this train must also be cleaned and with the same cleaning type. An assumption is made that an unlimited amount of cleaning crews are available on the stations where it is possible to clean. In this way, it will require the same amount of time to clean several trains that are connected, as cleaning a single train, since a cleaning crew can be provided for each train.

This report will provide DSB with inputs on how to optimize the way they schedule their interior train cleanings. The data set provided by DSB was accompanied by DSB's own cleaning schedule for a normal week. Currently, DSB's scheduling of the cleanings are made based on a long term plan followed by manual adjustments. It would be a great advantage for DSB to have a way of fully automating the process. That way they can save money and time by not having workers doing the scheduling partly manually.

This report will focus on modelling the train schedule and develop a mathematical model that can construct a cleaning schedule based on minimizing the cost of the cleaning, while ensuring the cleaning standard of DSB is upheld. The modelling of the problem will involve finding a way on how to quantify how dirty the trains are, since no information have been given by DSB regarding how to measure dirtiness, and how to ensure that the trains are approximately equally clean at all time. The schedule will decide on what stations cleanings will happen. The mathematical model will be developed using methods of integer programming.

The data set was not accompanied by passenger information like how many passengers usually travel on each train ride and there has been no clarification of the cleaning standard that should be upheld. A problem to be handled, is therefore finding a way to define DSB's cleaning standards based on their current cleaning schedule.

Overall this report aims to create a model that complies with the cleaning standard of DSB for the lowest cost possible.

3 Theory

The main method used in solving the problem of this report is Mixed-Integer Linear Programming (MILP). This method is used for optimization of a linear problem where some or all of the variables are fixed to be integer. This means the problem contains a linear objective function, only have linear constraints and the variables can either be continuous, binary or integer. These problems are most often solved to optimality by a method called branch and bound. In this project we have used the programming language Julia to implement different models and we have used the solver Gurobi to solve the MILP problems.

The Gurobi solver uses mainly branch and bound along with other methods to solve the problems. Many methods can be used to simplify the problems and make them easier to solve and of these we will elaborate on cutting planes and heuristics. [4]

3.1 Branch and bound

The overall idea of branch and bound is to break the MILP problem into a series of smaller sub-problems that are easier to solve. After solving them individually, the optimal solution is found by comparing the solutions to the sub problems. The division of the subproblems starts with the branch and bound algorithm solving the LP relaxation of the problem. This gives a problem defined by the same objective, the same constraints and with relaxed variables, so the integer and binary decision variables are allowed to be fractional. Meaning that integer variables are now allowed to be all real numbers and binary variables are allowed to take any value between zero and one. The relaxation will induce:

$$\begin{array}{rcl} x \in \mathbb{Z} & \rightarrow & x \in \mathbb{R} \\ x \in \{0,1\} & \rightarrow & 0 \leq x \leq 1 \end{array}$$

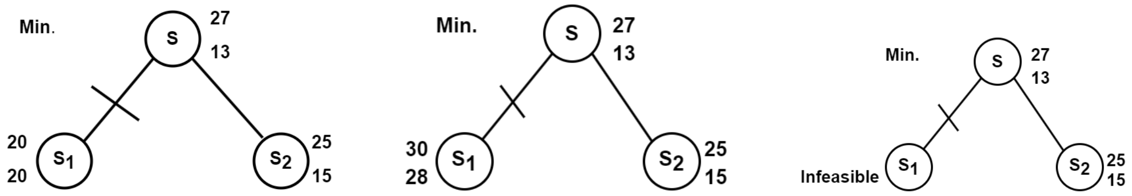
The solution from the relaxation can have two outcomes. If a solution with only non-fractional variables is found the problem is already solved to optimality. This rarely happens though. Otherwise, if some of the integer or binary variables from the original problem are fractional, the method chooses one of these to start branching on. Branching on a variable means creating two new sub-problems where the variable is constrained more tightly by upper and lower bounds. Suppose an integer variable has the value $x = 5.2$ after the LP relaxation. It is then desired to exclude the solution with this fractional value. Then two branches will be made, restricting the variable to $x \leq 5$ and $x \geq 6$ on each branch and solving each sub-problem with this as an added constraint. This is repeated on every fractional variable until a complete enumeration is found. Since it is too time consuming for all problems to find a complete enumeration there are three reasons that allows the method to prune the branch and thus removing a large number of enumerations implicitly. Pruning means stopping a branch from being further divided into sub-problem. A branch can be:

1. Pruned by optimality - this means that the lower and upper bound of a node in the tree are equal. An example of this can be seen in Figure 3.1 (a) . In this case there is no reason to examine the sub-problem from this branch further and the branch can be pruned by optimality.

2. Pruned by bound - this means that for a maximization problem, the upper bound in a node is less than or equal to the lower bound of a node higher in the tree. This because

the lower bound of a node ensures that the optimal value can be found higher that this. For a minimum problem it is the other way around. Suppose in a minimization problem an upper bound of 27 is found and a lower bound of 13 and the branching looks as seen in Figure 3.1 (b). In node S_1 it can be seen that the lower bound is 28, which is higher than the upper bound in node S. Since the upper bound in node S ensures that the optimal solution is no higher than this, no optimal solution can be in the set S_1 and this branch is pruned by bound.

3. Pruned by infeasibility - which means that a branching with a restricted variable turns out to be infeasible and therefore has no solutions. See Figure 3.1 (c).



(a) S_1 pruned by optimality (b) S_1 pruned by bound (c) S_1 pruned by infeasibility

Figure 3.1: The three ways to prune the enumeration tree.

If it is not possible to prune the tree, the branch and bound algorithm will continue [5].

3.2 Cutting planes

Cutting planes are made to remove undesirable fractional solutions from the relaxation, thus to find the smallest polyhedral which contains all feasible integer solutions. This means that cutting planes tighten the formulation which gives better lower bounds from start. This is visualized in Figure 3.2.

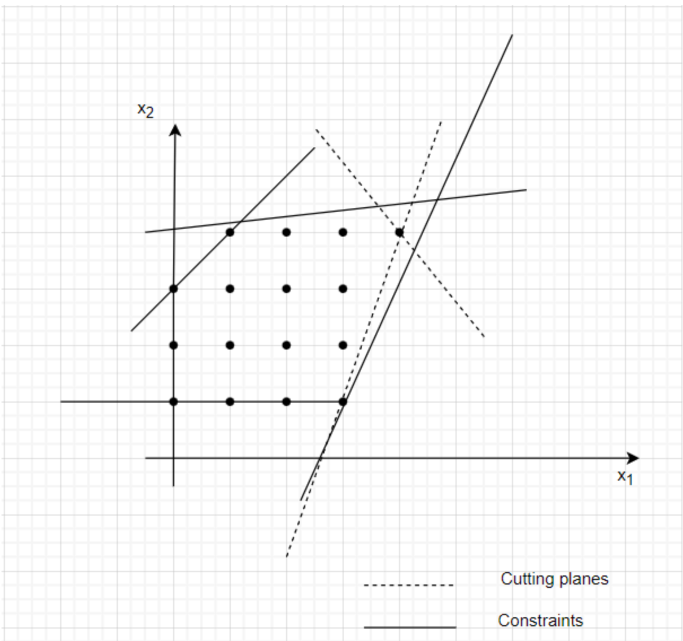


Figure 3.2: Constraints and cutting planes visualized. The dots represent integer solutions in the feasible solution space.

The cutting planes are produced during the solution process which provides better lower bounds and ensures that the branch and bound algorithm does not create too many sub-problems [4]. A small example of a cutting plane to a problem is as follows. Suppose the original problem contains the constraint:

$$2x_1 + 2x_2 + 3x_3 + 4x_4 \leq 6$$

where all variables x_i, $\forall i \in \{1, 2, 3, 4\}$ are binary variables. A solution from the LP-relaxation gives $x_1 = 0$, $x_2 = 1$ and $x_3 = x_4 = \frac{3}{4}$. From this an observation can be made: $3 + 4 = 7 > 6$ which implies that $x_3 = x_4 = 1$ is not possible since it would violate the original constraint. Thus the valid inequality

$$x_3 + x_4 \leq 1$$

can be added as a constraint to the problem. This will define our cutting plane, by cutting of a part of the solution space. Since the solution from the LP-relaxation would give $\frac{3}{4} + \frac{3}{4} = \frac{3}{2}$ which is greater than 1, the new valid inequality has cut of this fractional solution which is what the cutting plane intended.

3.3 Heuristics

A heuristic is a method to solve a MILP problem, by using a set of rules that, when followed, can give a feasible solution, yet rarely the optimal solution. The idea of heuristics are therefore to find ”good” feasible solutions in a short amount of time. In a minimization problem a heuristic can be used to find a feasible solution, also called the incumbent value, to the MILP problem. This is desirable since all values found by LP-relaxations that exceeds the incumbent value can then be disregarded. Thus the heuristic helps, just as cutting planes, to limit the size of the branch and bound tree and thereby create a faster solution. There are many different heuristics that can be used to solve a MILP problem such as relax-and-fix, cut-and-fix and dive-and-fix. [6].

An example of a heuristic could be dive-and-fix, which takes the solution from the LP-relaxation and attempts to change it into a feasible solution to the MILP problem. Briefly explained, the dive-and-fix heuristic fixes a variable and repeats the relaxation. Suppose a problem is binary and in the LP-relaxation the variables get fractional values. Then the dive-and-fix algorithm fixes a variable to 0 if the solution gives a fractional value of $<$ 0.5 or fixes the variable to 1 if the solution gives a fractional value of $\geq$ 0.5. The algorithm then makes a new LP-relaxation, now with one variable value fixed. This continues until all variables are binary and a solutions to the problem has been found. If this solution is feasible and better than the incumbent value this will now be the new incumbent value [5].

3.4 Linearization and the Big M method

In this report the ”Big M method” is used to linearize non-linear constraints. This is done by introducing an artificial variable z that should equal the non-linear term, e.g. $x \cdot y$ where x is a binary variable and y is a non-negative continuous variable. z then substitutes the non-linear term in the constraints, meaning that $z = x \cdot y$. Several linear constraint are then set to ensure that z gets the correct value corresponding to $x \cdot y$. If $x = 0$ the whole term equals zero and so should z. This is done by setting the constraint:

$$z \leq x \cdot M \tag{3.1}$$

where M is a sufficiently large number to only bound z if $x = 0$. When $x = 1$, then z is upper bounded by M, so as long as M is of big enough value to not constrict z in any way, this works.

Since the product will never be less than 0 or greater than y, z can be bounded by:

$$\begin{aligned} z &\geq 0 \\ z &\leq y \end{aligned} \tag{3.2}$$

Now it is only needed to ensure that $z = y$ when $x = 1$.

$$z \geq y - (1 - x) \cdot M. \tag{3.3}$$

In equation 3.3 it is seen that when $x = 1$ the negative term $-(1 - x) \cdot M$ equals to zero and the constraint sets that $z \geq y$, so then z is both upper and lower bounded by y, as seen in equation 3.2, meaning it is set to equal y. When $x = 0$ then M is subtracted from y, and it is set that $z \geq y - M$. This shows the importance of choosing M sufficiently large. That means that M should always be greater than the highest possible value of y, for this constraint not to set a lower bound on z.

This method can be used to linearize different non-linear terms, based on what kind of variables are being multiplied. This makes it easier and quicker for the program to solve, and will be used in this report [7].

4 Data Analysis

4.1 Data description

The data set given by DSB is an excel sheet of 13,852 rows and 21 columns. Each row represents one train ride in chronological order. From the 21 columns only 10 are relevant for this project.

The week that the data set represents runs from the 6th to the 12th of March. Some of the trains have a train ride starting on the 5th of March. This is because the train arrives at the arrival station on the 6th of March and this train ride is therefore included in the data. Even though the provided data has the specific dates, it is still a representation of a normal week that could be used for every other week of the year. The model will not take seasonal changes into consideration. The data set does not include passenger information like how many passengers usually travel on each train ride, and this will not be taken into consideration in this report.

The representation of a train ride includes columns with the departure station, the arrival station and the kilometers between these two station, meaning the distance of the train ride. There are 30 end stations in total visited in a week. Some of them are visited several times by different or the same train. It is known that there cannot be cleaned on all the stations. Corresponding to the departure and arrival stations, there are two columns with the respective departure and arrival time.

The representation of a train ride includes a column describing the litra type of the train performing the train ride, a column called ”serial number” and one called ”train number”. The difference between the train number and serial number needs to be defined. The serial number is connected to a certain physical train, meaning all rows with the same serial number and the same corresponding litra type, are all the train rides performed by the same physical train. The train number is connected to a certain train ride performed by one or several trains that are connected for one or several train rides. During the week of train rides a train therefore has many different train numbers. An example of this is the train with serial number 1 and litra type ERF. Two subsequent train rides performed by this train, has different train numbers. The first train ride departs from HG and arrives at KH, and has the train number is 3429. The next train ride departing from KH and arriving at FA has the train number 129.

In total, there are 401 different trains. The first 46 serial numbers is of the litra type ERF, the next 14 serial numbers are of the litra type ETS and so on. To summarize, there are 46 ERF trains, 14 ETS trains, 101 ICA trains, 63 MGA trains, 42 EB trains, 30 ABS trains, 74 B trains and 31 BK trains operating in a normal week. Two trains that have the same litra type do not necessarily have the same number of train rides in a week. Several trains, of the same litra type and of different serial number can sometimes have the same train number. This means that the trains are joined on the departure station of the corresponding train ride and are connected for that ride. There are 1,250 stops where two or more trains are traveling through together connected.

The last relevant column is called ”total kilometers” and describes the accumulated kilometers that the train has driven so far. Thus it is reset when a new train starts.

4.2 Analysis of the train schedule

Some relevant numbers are extracted from the train schedule in the data set.

Table 4.1: Maximum, mean and minimum number of train rides performed by a train in a week for each litra type.

	ERF	ETS	ICA	MGA	ABS	B	BK
Max. [No.]	80	104	77	47	79	79	79
Mean [No.]	49.58	91.57	49.88	27.38	55.67	54	55.67
Min. [No.]	20	83	6	15	21	21	21

Table 4.1 describes the maximum, minimum and mean number of train rides performed by a train of each litra type. Eg. the ERF train with the most train rides in a week, performs 80 train rides and all the trains of litra type B performs a mean of 54 train rides in a week. The data set includes some maintenance rides that has a length of 0 kilometers, these are not included in this table. The trains of litra type ABS, B and BK are seen to have similar maximum, minimums and almost equal mean values of the number of trips in a week.

ETS stands out in having the highest numbers in all three rows. Since it is the train type that performs most trips in a week, this might indicate it has shorter train rides than the other litra types, since it has time to perform so many train rides in a week. Looking at Figure 4.1 this does not look to be the case.

In the same Table it can be seen that the litra type ICA has a train that only performs six train rides in a week, this could be because these are six very long train rides.

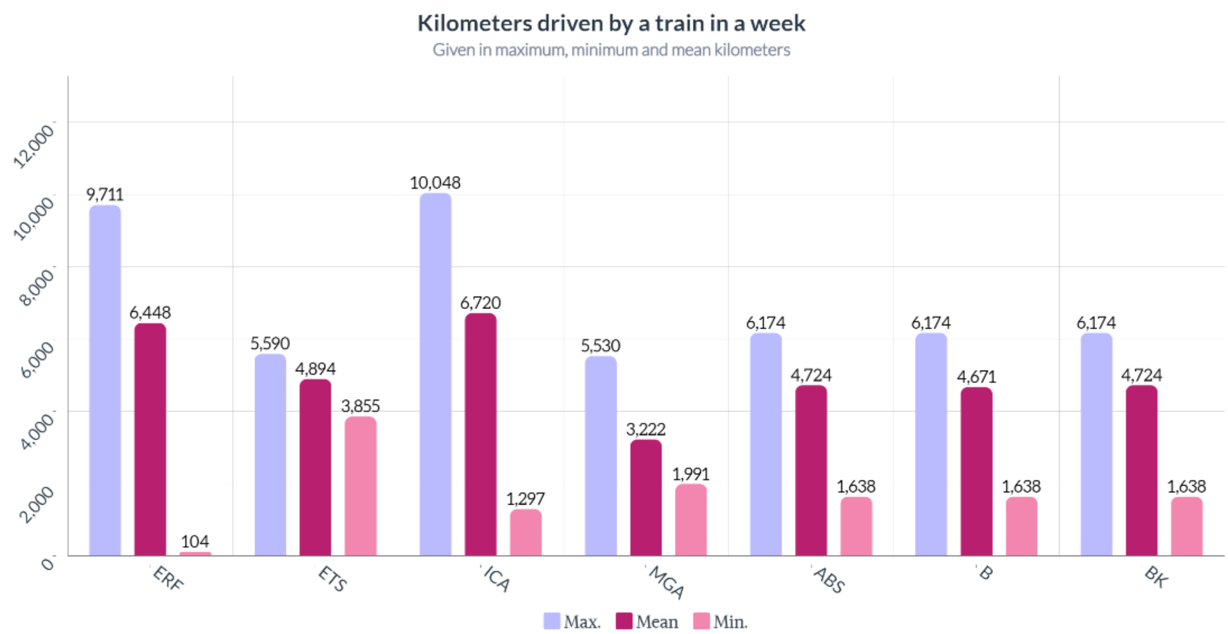


Figure 4.1: Maximum, minimum and mean kilometers of all the train rides performed by a train in a week per litra type.

Figure 4.1 show the maximum, minimum and mean of kilometers driven by a train per litra type. Eg. the train that covers the most kilometers in a week, has a total distance of 10,048 km and is a train of litra type ICA. The train that covers the least kilometers in a week has a total distance of 104 km and is a train of litra type ERF.

Overall trains of litra type ICA covers a high number of kilometers in a week, since it has the highest maximum and mean kilometers.

On this figure it can be seen again that the ABS, B and BK trains have very similar numbers, and this reinforces the presumption that they may have very similar schedules, and perform some of the same rides. The trains of litra type ERF stands out as being the type with most spreading in the different schedules of the trains being the litra type with the lowest minimum and the second highest maximum and mean values.

4.3 Analysis of the cleaning schedule

The data set was accompanied by DSB's cleaning schedule for the week. This corresponds to DSB's current solution. In Table 4.2 some numbers are drawn from the provided solution to draw some conclusions to what DSB's cleaning standards could be.

Table 4.2: The number of cleanings in total and the number of TR- and OR-cleanings, as well as the maximum, mean and minimum of kilometers between cleanings, for the solution given by DSB.

	Cleanings [No.]	TR-cleaning [No.]	OR-cleanings [No.]	Max. [km]	Mean [km]	Min. [km]
All litra types	2,206	1,876	730	1,574.90	523.33	3.10
Litra type ERF	599	285	314	1,574.90	395.86	17.00
Litra type ETS	63	57	6	1,242.60	573.40	5.20
Litra type ICA	1,113	781	332	1,518.80	500.02	3.10
Litra type MGA	335	262	73	1,290.50	393,31	113.70
Litra type ABS	110	109	1	1,170.20	794.50	65.00
Litra type B	276	273	3	1,170.20	811.88	65.00
Litra type BK	110	109	1	1,170.20	794.75	65.00

The litra type EB has been left out in Table 4.2. This is because it is a locomotive which requires no cleaning and is thus not relevant here. The numbers in Table 4.2 are based on the kilometers between each cleaning, on each train, for a week, as well as the kilometers before the first cleaning and after the last cleaning on each train. If a train is not cleaned during the entire week, then the total number of kilometers driven by the train that week will be counted as the kilometers between two cleanings, assuming the train is clean at the beginning of the week.

A significant difference on both the maximum and mean kilometers between cleanings on the litra types can be seen. The maximum kilometers allowed between cleanings on the different litra types vary with up to 400 km, this is a difference of 34.5 % from 1,170.20 km. to 1.574.90 km.

ICA is the litra type with the most cleanings as can be seen in table 4.2. This makes sense as ICA is the train type that travels the most as seen in 4.1. It is also the litra type that defines the minimum kilometers between cleanings out of all the litra types. This means that at some point an ICA train is cleaned, then it performs a train ride of 3.10 kilometers and is then cleaned again. The trains of type ABS, B and BK has numbers very alike and very different from the other litra types. They have a lower maximum kilometers between cleanings, but their mean kilometers between cleanings are much higher than the first five litra types.

In the cleaning schedule from DSB there are significantly more TR-cleanings than OR-cleanings respectively 1,876 TR-cleanings and 730 OR-cleanings, as can be seen in Table 4.2. This gives a total of 2,206 cleanings in a week, where 71.9% of the cleanings are TR-cleanings and 28.1% are OR-cleanings. Looking at how the cleanings are placed, it can be seen that a TR-cleaning is often done as the first thing in the morning so all trains starts out clean every day. This does not apply for all trains every day, but is a tendency overall for all litra types. It can also be seen that DSB's solution contains several cleanings in continuation, which results in the very low minimum kilometers between cleanings.

The cleaning standard of DSB is determined as the maximum kilometers between cleanings as seen in Table 4.2. This will be used as a limitation, when modeling the problem, as the highest amount of kilometers that are allowed to be driven by a train without cleaning.

4.4 Revised data

The data has been revised to make it easier to model the problem based on the data set. Only the relevant columns are kept and some columns are added, to be used in modelling the problem.
The 11 columns not mentioned in the data description are removed for manageability. Some examples of these are ”week number”, ”placement on the station”, ”country codes” etc. that will not be needed for the model. The 10 remaining columns are seen in Table 4.3, along with the six added columns.

It is desired to know how long a train is at a stop. Hence the first added column is with the calculated time that the train is on the arrival station before leaving, and starting a new train ride. This column is called ”Stop time”. The stop time on the last train ride of each train of the week is not known, since no information is available about when the next train ride happens. The stop time in each of these instances are set to zero.

A column called ”Possible cleaning” is added. This column consists of binary values based on whether there are cleaning crews available on the arrival station or not. This is based on information from Table 2.2. If a row consists of a 1 in this column, it indicates that it is possible to clean at the arrival station of this train ride. On the other hand, if it consist of 0, it is not possible.

Two columns containing the different cleaning times depending on the litra type are also added. One column called ”OR Time” and one called ”TR Time”, each describing the time it takes for the respectively TR- or OR-cleanings to be performed on the arrival station of each train ride based on the litra type. The numbers are extracted from table 2.1.

Lastly, from the analysis of the cleaning schedule provided by DSB, a cleaning standard was determined. It is therefore desirable to have a column that, corresponding to the litra type, consists of the highest amount of kilometers that are allowed before a cleaning needs to be performed. This column is called ”C-value max” and contains the maximum kilometers per litra type from Table 4.2.

Some rows are removed as well, so only the rows symbolizing the train rides that should count in modelling the cleaning is left. The trains of the litra type EB are locomotives. Table 2.1 shows that EB trains has no cleaning time, since they do not require cleaning. The litra types ME are maintenance trains, these trains drive zero kilometers with passengers and do not require cleaning as well. The locomotives and the maintenance trains are removed from the data set, meaning the rows representing the train rides performed by these two litra types, are deleted. The data now consists of 7 different litra types: ERF, ETS, ICA, MGA, ABS, B and BK.

After removing all trains with those litra types, 256 trains are left out of the 401 trains. These are 37 ERF trains, 8 ETS trains, 85 ICA trains, 40 MGA trains, 19 ABS trains, 49 B trains and 19 BK trains.

The revised data is now an excel sheet of 12,638 rows and 16 columns. A sample of the revised data can be seen in Table 4.3a and Table 4.3b.

Table 4.3: Data representation of the revised data, showing 4 of the 12,638 train rides.

(a) Showing column 1 to 8.

Row Number	Serial Number	Litra Type	Departure Date [dd/mm]	Train Number	Departure Station	Departure Time [hh.mm]	Arrival Time [hh.mm]
1	1	ERF	05/03	4368	RO	22.38	00.04
2	1	ERF	06/03	3429	HG	07.11	07.55
3	1	ERF	06/03	129	KH	08.05	10.06
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
12,638	19	BK	12/03	2277	HG	19.18	21.09

(b) Showing column 1 and column 9 to 16.

Row Number	Stop Time [min]	Arrival Station	Kilometers Current trip [km]	Total Km [km]	Possible Cleaning [BIN]	TR Time [min]	OR Time [min]	C-value max
1	427	HG	77.5	77.5	1	80	20	1,575
2	10	KH	46.2	123.7	1	80	20	1,575
3	47	FA	219	342.2	1	80	20	1,575
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
12,638	471	NÆ	137	4,850	1	175	47	1,171

5 Model 1 - The basic model

The objective of the model is to minimize the cost of the cleaning on the entire week while upholding the derived cleaning standard from the data analysis. The first model will look at each train ride as an individual train, ignoring the need for them to be able to join, and thus be cleaned alike while connected. Later, the model will be expanded twice to increase the complexity.

Since the data does not include the cleaning costs, another way of minimizing the cost is needed. This is done by minimizing the sum of the cleaning time in the objective function, as DSB have clearly indicated that the cleaning cost is related to the cleaning time.

To model this, an amount D is introduced containing all indices in the set, $D = \{1, 2, ..., 12638\}$, representing each row in the revised data.

The first model has the objective function:

$$\textbf{Min} \sum_{i \in D} Xt_i \cdot TR_i + Xo_i \cdot OR_i \tag{5.1}$$

Where Xt_i and Xo_i are binary decision variables and TR_i and OR_i represent the i'th component of a vector of constants.

$$\begin{aligned} Xt_i &\in \{0, 1\} \quad \forall i \in D \\ Xo_i &\in \{0, 1\} \quad \forall i \in D \end{aligned} \tag{5.2}$$

Xt_i defines whether or not a cleaning of type TR is performed on stop i and Xo_i, whether a cleaning is performed of the type OR on stop i. Stop i refers to the arrival station of the i'th train ride. i is defined by the amount D, and will therefore refer to each row in the data set. TR_i is the time a TR-cleaning takes and OR_i is the time an OR-cleaning takes, given in minutes, respective to the litra type of the train. This way our objective function will minimize the time spend on cleaning.
The constraints of the model are presented and implemented in the following sections.

5.1 When can a cleaning take place

To make sure that cleanings only happen when it is possible, two constraints are set. Firstly, one making sure that a cleaning can be performed at the arrival station, since not all stations have cleaning crews available. This is done based on the binary ”possible cleaning” column that was added to the revised data, abbreviated to Pc in the model.

$$Xt_i + Xo_i \leq Pc_i, \quad \forall i \in D \tag{5.3}$$

This constraint also ensures that there will be no instances of both a TR- and OR-cleaning being done on the same stop. This happens since Pc is binary and only have values of one or zero. In case it is possible to clean at stop i the corresponding Pc would have the value one and either the Xt or the Xo variable can take the value of one or neither.
From this constraint some of the decisions variables will be forced to be zero because when it is not possible to clean at station i, $Pc_i = 0$, and both Xt_i and Xo_i will be set to zero.

Secondly, a constraint is set to ensure that there is enough time, between the arrival on the i'th train ride and departure of the $i + 1$'th train ride for a cleaning. The column ”Stop time" is used, and holds the time between the two subsequent train rides, in minutes. It is abbreviated to St in the model.

$$Xt_i \cdot TR_i + Xo_i \cdot OR_i \leq St_i, \quad \forall i \in D \tag{5.4}$$

This will force the decision variables to zero if the stop time, St, on station i is less than the time needed for the respective cleaning. This ensures that a cleaning is only set if there is enough time for it. Since the last train ride of the week for each train has $St = 0$, a cleaning will never happen at these stops.

5.2 Quantifying dirtiness

To ensure that the trains always uphold a standard of cleanliness there has to be introduced a variable that keeps track of how clean or dirty the train is. In this report it is measured by how many kilometers the train has travelled without being cleaned. This variable is called "Kilometers of Dirtiness" and will be abbreviated to KD in the model. This will sum up the kilometers driven that has "added dirtiness" to the train and will be subtracted from, whenever the train is cleaned. KD will be a vector and KD_i will refer to the kilometers of dirtiness on stop i.

When a Tr cleaning is done on stop i, KD_i will be set to zero, meaning that all the dirty kilometers have been discarded by the full cleaning. To decide how much will be subtracted when an OR-cleaning happens, the cleanings need to be quantified in comparison to each other. This is hard to estimate, since some cleaning tasks are in the TR-cleaning that are not in the OR-cleaning, making it hard to say exactly how much cleaning the OR-cleaning does compared to the TR-cleaning. This will be based on the assumption that the amount of cleaning in an OR-cleaning is about half the amount of a TR-cleaning. Hence, when an OR-cleaning is performed on stop i, KD_i will be reduced by multiplying it with a constant q given by $q = \frac{1}{2}$.

The impact of the value of q on the solution will be determined later on in the report. KD will be calculated based on the decision variables as a recursive function. To do this an amount E is introduced given by $E = \{2, 3, ..., 12638\}$.

$$KD_1 = Km_1 \tag{5.5}$$

$$KD_i = KD_{i-1} + Km_i - (Xt_{i-1} \cdot KD_{i-1} + q \cdot Xo_{i-1} \cdot KD_{i-1}) \forall i \in E \tag{5.6}$$

Equation 5.5 sets the base case of KD to be equal to the amount of kilometers driven on the first train ride, since km_i is a vector corresponding to the kilometers of train ride i. The rest will be calculated recursively as in Equation 5.6 by adding the *Kilometers of dirtiness* from the last train ride to the kilometers from the current train ride and subtracting the *cleaned kilometers*. The *cleaned kilometers* is expressed by the term

$$Xt_{i-1} \cdot KD_{i-1} + q \cdot Xo_{i-1} \cdot KD_{i-1}$$

which means that if a TR-cleaning happened on the previous stop $i - 1$, then $Xt_{i-1} = 1$ an then the *kilometer of dirtiness* will be set to zero.

$$q \cdot Xo_{i-1} \cdot KD_{i-1}$$

will remove the q'th part of the *Kilometers of dirtiness* on the previous stop if an OR-cleaning has happened. Overall this evaluates KD_i based on whether or not the train has been cleaned on previous stops.

The *Kilometers of dirtiness* will need to be reset whenever the data set gets to the start of a new train, setting it as a new base case to equal the kilometers of the first ride of the train of the week, and restarting the recursion for each train. This is because when a new train starts, it is assumed that it is clean from the start. A set F of indices i is introduced, for which it holds that the serial number of train ride i and the serial number of train ride $i-1$ are not equal to each other. This way it is marked that a new train starts its week on the i'th index. For this definition the serial number will be denoted Sn.

$$F = \{i \in D \mid Sn_i \neq Sn_{i-1}\}$$

This gives the constraint:

$$KD_i = Km_i \quad \forall i \in F \tag{5.7}$$

KD_i is bounded to ensure that the train upholds a certain cleaning standard. This cleaning standard is derived in Chapter 4 and is determined to be the maximum kilometers between two cleanings in the current cleaning schedule provided by DSB. For this model the maximum on all litra types are used, found to be 1,574.90 $\approx$ 1,575.00 kilometers as seen in Table 4.2.
KD_i is then bounded by a constant introduced as C:

$$\begin{aligned} &KD_i \leq C, \qquad \forall i \in D \\ &C = 1,575.00 \end{aligned} \tag{5.8}$$

KD will now be evaluated backwards, based on if it will exceed C, without a cleaning on the previous stations.

5.2.1 Implementation of the constraints for *Kilometer of dirtiness*

The recursive constraint for calculating KD seen in equation 5.6 is non-linear. The model is solved with a MILP-solver, thus the recursion should be linearized. The linear version of this is set up by multiple constraints introducing two new variables: Zt_i and Zo_i, referring to the TR- and OR-cleanings on the i'th station. Here the Z-variables is an expression of the "cleaned kilometers" that should be subtracted from KD for each station. The "Big M" method is used to bound the variables. M is set to equal C, as this will be the biggest value KD is allowed to take.

$$\begin{aligned} Zt_i \leq Xt_i \cdot M \quad \forall i \in D \\ Zo_i \leq Xo_i \cdot M \quad \forall i \in D \end{aligned} \tag{5.9}$$

$$Zt_i + Zo_i \leq KD_{i-1} \quad \forall i \in E \tag{5.10}$$

$$\begin{aligned} Zt_i \geq KD_{i-1} - (1 - Xt_{i-1}) \cdot M \quad \forall i \in E \\ Zo_i \geq KD_{i-1} - (1 - Xo_{i-1}) \cdot M \quad \forall i \in E \end{aligned} \tag{5.11}$$

$$\begin{aligned} KD_i \geq KD_{i-1} + Km_i - (Zt_i + q \cdot Zo_i) \quad \forall i \in E \\ KD_i \leq KD_{i-1} + Km_i - (Zt_i + q \cdot Zo_i) \quad \forall i \in E \end{aligned} \tag{5.12}$$

$$\begin{aligned} 0 \leq Zt_i \quad \forall i \in D \\ 0 \leq Zo_i \quad \forall i \in D \end{aligned} \tag{5.13}$$

Equations 5.9 use the "Big M" notation to create a lower and upper bound for each of the Z-values. If Xt_i or Xo_i equals zero, then the corresponding Z-variable will be bounded by zero. Because it also has the lower bound of zero set in Equation 5.13, it will equal zero in this case. Since Z represents the cleaned kilometers, meaning what should be subtracted from KD when a cleaning is set, it should equal zero when no cleaning is set. If Xt_i or Xo_i equals one, then it will have the upper bound of $M = C$, and the lower bound of zero. The values will then be further defined by Equation 5.10 and 5.11.

Equation 5.10 sets another upper bound for the Z-variables. This ensures that there will be no more ”cleaned kilometers” than there are ”dirty kilometers”.

Equation 5.11 puts a lower bound on the Z-values, saying that the Z-values should always be greater than or equal to the previous KD value if no cleaning was performed on the last station and greater than or equal to zero, if a cleaning was performed.

This means that Equations 5.9, 5.10 and 5.11 together ensure that the cleaned kilometers are both upper and lower bounded in both cases of a cleaning happening on the previous stop or not. So if a cleaning has occurred on the previous stop, the Z_i-value will equal to KD_{i-1} and if not it will equal to zero.

Equation 5.12 subsequently ensures that KD will equal its predecessor plus the kilometers ridden on the current trip minus the ”cleaned kilometers” defined by the Z-values, and has avoided the non-linearity.

The constrain set in equation 5.7 is implemented as follows:

$$\begin{aligned} KD_i &\leq Km_i \quad \forall i \in F \\ KD_i &\geq Km_i \quad \forall i \in F \end{aligned} \tag{5.14}$$

This will be written as Equation 5.7 for the rest of the report for simplicity.

5.3 The model

The model written with all constraints and all variables defined.

$$\begin{array}{lll}
\textbf{Min} & \sum_{i \in D} Xt_i \cdot TR_i + Xo_i \cdot OR_i & \\
 & & \\
\textit{s.t.} & Xt_i \cdot TR_i + Xo_i \cdot OR_i \leq St_i, & \forall i \in D \\
 & Xt_i + Xo_i \leq Pc_i, & \forall i \in D \\
 & & \\
 & Zt_i \leq Xt_i \cdot M & \forall i \in D \\
 & Zo_i \leq Xo_i \cdot M & \forall i \in D \\
 & Zt_i + Zo_i \leq KD_{i-1} & \forall i \in E \\
 & Zt_i \geq KD_{i-1} - (1 - Xt_{i-1}) \cdot M & \forall i \in E \\
 & Zo_i \geq KD_{i-1} - (1 - Xo_{i-1}) \cdot M & \forall i \in E \\
 & & \\
 & KD_1 = Km_1 & \\
 & KD_i = Km_i & \forall i \in F \\
 & & \\
 & KD_i \geq KD_{i-1} + Km_i - (Zt_i + q \cdot Zo_i) & \forall i \in E \\
 & KD_i \leq KD_{i-1} + Km_i - (Zt_i + q \cdot Zo_i) & \forall i \in E \\
 & KD_i \leq C & \forall i \in D \\
 & & \\
 & 0 \leq Zt_i \leq C & \forall i \in D \\
 & 0 \leq Zo_i \leq C & \forall i \in D \\
 & KD_i \geq 0 & \forall i \in D \\
 & Xt_i \in \{0,1\} & \forall i \in D \\
 & Xo_i \in \{0,1\} & \forall i \in D
\end{array} \tag{5.15}$$

6 Model 2 - The expanded model

The second model is an expansion of Model 1, where it is incorporated that several trains can be joined on a station, travel together connected and then be separated again. When two or more trains are connected they will all be cleaned at the same stop and with the same cleaning type. For this model the objective function remains the same, since the goal still is to minimize the cost of cleanings.

To be able to indicate when trains are connected, a set G is introduced. G is a set of pairs of indices i and j, for which it holds that the train number denoted Tn, the departure date denoted Dd, the departure station denoted Ds and the stop time St are the same for both train ride i and train ride j. The amount G is defined by:

$$G = \{i, j \in D \times D \mid Tn_i = Tn_j \land Dd_i = Dd_j \land Ds_i = Ds_j \land St_i = St_j\}$$

The stop time for the pairs will be equal to each other when trains arrive to a stop connected and leave a stop connected, and in these instances the trains will be forced to be cleaned the same way if to be cleaned. A set of constraints are set for these pairs:

$$\begin{aligned} Xt_i = Xt_j \quad \forall (i,j) \in G \\ Xo_i = Xo_j \quad \forall (i,j) \in G \end{aligned} \tag{6.1}$$

This ensures there will not be performed a TR- or OR-cleaning on one train and not on the other, when they are connected. These constraints are implemented as such:

$$\begin{aligned} Xt_i - Xt_j \leq 0 \quad \forall (i,j) \in G \\ Xt_j - Xt_i \leq 0 \quad \forall (i,j) \in G \\ Xo_i - Xo_j \leq 0 \quad \forall (i,j) \in G \\ Xo_j - Xo_i \leq 0 \quad \forall (i,j) \in G \end{aligned} \tag{6.2}$$

This will be written as in Equation 6.1 for simplicity in the rest of this report.

Since it is not specified how many cleaning crews are available at the stations where it is possible to clean, an assumption has been made that an unlimited amount of cleaning crews are available. That way, the constraint ensuring that the stop time is long enough for the cleanings are the same as in model 1, and set with the rest of the constraints in Equation 6.3.

6.1 The model

The model can be seen here in full with the new constraints ensuring that trains are cleaned the right way when travelling connected.

$$
\begin{array}{lll}
\textbf{Min} & \sum_{i\in D} Xt_i \cdot TR_i + Xo_i \cdot OR_i & \\
& & \\
\textit{s.t.} & Xt_i \cdot TR_i + Xo_i \cdot OR_i \leq St_i, & \forall i \in D \\
& Xt_i + Xo_i \leq Pc_i, & \forall i \in D \\
& & \\
& Zt_i \leq Xt_i \cdot M & \forall i \in D \\
& Zo_i \leq Xo_i \cdot M & \forall i \in D \\
& Zt_i + Zo_i \leq KD_{i-1} & \forall i \in E \\
& Zt_i \geq KD_{i-1} - (1 - Xt_{i-1}) \cdot M & \forall i \in E \\
& Zo_i \geq KD_{i-1} - (1 - Xo_{i-1}) \cdot M & \forall i \in E \\
& & \\
& KD_1 = Km_1 & \\
& KD_i = Km_i & \forall i \in F \\
& & \\
& KD_i \geq KD_{i-1} + Km_i - (Zt_i + q \cdot Zo_i) & \forall i \in E \\
& KD_i \leq KD_{i-1} + Km_i - (Zt_i + q \cdot Zo_i) & \forall i \in E \\
& KD_i \leq C, & \forall i \in D \\
& & \\
& Xt_i = Xt_j & \forall (i,j) \in G \\
& Xo_i = Xo_j & \forall (i,j) \in G \\
& & \\
& 0 \leq Zt_i \leq C & \forall i \in D \\
& 0 \leq Zo_i \leq C & \forall i \in D \\
& KD_i \geq 0 & \forall i \in D \\
& Xt_i \in \{0,1\} & \forall i \in D \\
& Xo_i \in \{0,1\} & \forall i \in D
\end{array}
\tag{6.3}
$$

7 Experiments and results

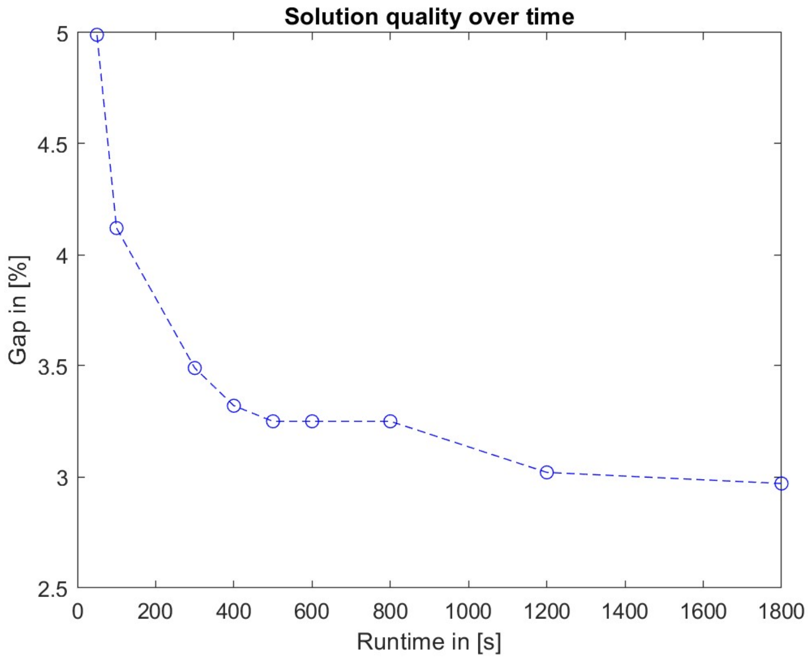
7.1 Run time experiment

A problem of run time arises when getting the Gurobi solver to run the implementation of model 2. After 5 hours the solver has not yet found an optimal solution for the problem. One reason for this is that the feasible solution space is very large and the solver is therefore forced to create a large enumeration tree when solving with branch and bound. To avoid this being a problem, a time limit is needed for the run time. When branch and bound is performed by the solver it will find the best dual bound, which in minimization problems is a lower bound, as well as the best primal bound, which is the best feasible solution.

The duality gap, is the gap between these two bounds given in percent, and indicates the solution quality. When the solver hits the time limit set, it will give the best feasible solution found along with the duality gap [5]. This duality gap can help understand how far away from the optimal solution the feasible solution might be in the worst case. There might not be a better solution in the gap, it is a gap containing the subproblems in the branching that has not been explored yet. This means that certain subproblems have not been evaluated to see if a better solution is there. If a better solution exists within those subproblems, it will at most be the duality gap percentage better than the already found feasible solution. If no feasible solution exist within the non-evaluated subproblems, then the found solution is the optimum. Often when a feasible set is very large, there will be so many subproblems to evaluate, that as the duality gap closes, the solution will begin to stall and very small gap changes will start to take a very long.

The time limit has been tested on model 2, with respect to the duality gap relative to the run time and from this chosen based on what time limit produced the smallest gap in a reasonable time.

Figure 7.1: This figure shows the duality gap in percent relative to the run time given in seconds.



In Figure 7.1 it is seen that the gap decreases significantly until around 400-500 seconds of run time, then the curve flattens out. After this point, only minor changes in the duality gap happens when the run time increases. Often when running complex problems, this shape of curve can be seen, and the limit should be chosen as the point where the curve has flattened out, to get the most efficient run time.

Based on the tests of this data set and as seen in Figure 7.1 the solution is stalling from 500 second to 800 seconds, meaning that the 300 seconds does not decrease the gap at all, and thus does not increase the solution quality. To achieve good results in a reasonable time limit, it has been chosen to prioritise a faster solution over a very little change in the duality gap. Therefore for this report the time limit of 500 seconds is set with a duality gap of 3.25 %. This is what the results from model 2 will be based on. It does not necessarily mean that the found feasible soultion is not the optimal solution, but it means there is a gap of unexplored branches that might contain a better feasible solution, but the optimum will at most be 3.25% better than the best feasible solution at this point.

7.2 Results

The results of the cleaning schedule that model 1 constructs are as seen in Table 7.1.

Table 7.1: Analysis on the solution from model 1. Shows the number of cleanings, as well as the maximum, minimum and mean kilometers between the cleanings, for the solution to model 1.

	Cleanings [No.]	TR-cleanings [No.]	OR-cleanings [No.]	Max. [km]	Mean [km]	Min. [km]
All litra types	1,527	72	1,455	1,574.90	772.81	5.20
Litra type ERF	271	0	271	1,573.10	762.16	17.00
Litra type ETS	38	0	38	1,572.40	835.17	140.40
Litra type ICA	663	31	632	1,574.90	757.39	11.80
Litra type MGA	113	0	113	1,550.00	848.31	142.50
Litra type ABS	98	8	90	1,533.80	767.34	5.20
Litra type B	246	25	221	1,533.80	776.11	5.20
Litra type BK	98	8	90	1,533.80	767.34	5.20

The same results of the cleaning schedule from model 2 is shown i Table 7.2.

Table 7.2: Analysis on the solution from model 2. Shows the number of cleanings as well as the maximum, minimum and mean kilometers between the cleanings, for the solution to model 2.

	Cleanings [No.]	TR-cleanings [No.]	OR-cleanings [No.]	Max. [km]	Mean [km]	Min. [km]
All litra types	1,562	77	1,485	1,572.40	757.93	5.20
Litra type ERF	275	5	270	1,563.40	752.39	17.00
Litra type ETS	38	0	38	1,572.40	835.17	165.40
Litra type ICA	693	31	662	1,566.80	728.19	11.80
Litra type MGA	114	0	114	1,550.00	842.80	110.60
Litra type ABS	98	8	90	1,533.80	767.34	5.20
Litra type B	246	25	221	1,533.80	776.11	5.20
Litra type BK	98	8	90	1,533.80	767.34	5.20

It can be seen in both solutions that there are several litra types where non of the trains are cleaned with a TR-cleaning. The solution from model 1 and the solution from model 2 are both based on the ratio factor q, that describes the amount an OR-cleaning cleans compared to a TR-cleaning, and is assumed to be $\frac{1}{2}$. This has the impact that the model chooses two OR-cleanings instead of one TR-cleaning, and that some trains are never cleaned with a TR-cleaning. This will be handled in model 3 in chapter 8.

From Table 7.1 and 7.2 it is seen that 35 more cleanings are performed in model 2 than model 1, which is an increase of 2.29 %. This means that there is only a very little change between the cleaning schedules made with model 1 and model 2. This change is induced by the added constraints in model 2, forcing trains that are connected to be cleaned either the same way or not at all, as can be seen in Equation 6.1. The lacking significance of the change means that the solution from model 1 does not break this constraint much hence only very little change in the solution of model 1 is needed to create a feasible solution to model 2. In total there are 1,250 stops where two or more trains, that are traveling together connected, can be affected by the new constraints implemented in model 2.

Table 7.1 and Table 7.2 shows that all litra types have approximately the same values of maximum and mean kilometers between two cleanings, indicating that both models produce very uniform solutions in regards to the different litra types.

It can also be seen that there are a lot more OR-cleanings than TR-cleanings on both solutions. This gives a ratio of TR-cleanings compared to all cleanings on the solution from model 1:

$$\frac{\textit{Number of TR-cleanings}}{\textit{Number of all cleanings}} = \frac{72}{1,527} = 0.047 \tag{7.1}$$

meaning that only 4.7% of the cleanings performed in a week are TR-cleanings.

Table 7.3: TR-cleaning ratio. The percentage of TR-cleanings out of all the cleanings.

	Model 1	Model 2	DSB's solution
Ratio of TR-cleanings	4.7 %	4.9%	71.9 %

In Table 7.3 it can be seen that the ratio of TR-cleanings in the cleaning schedule provided by DSB is significantly higher.

Model 1 produces a solution with an objective value of 46,616 minutes of cleaning. Model 2 produces a solution with an objective value of 47,496 minutes of cleaning.

8 Model 3 - The final model

In the data analysis in Chapter 4, it is observed that a tendency in DSB's cleaning schedule is that every day starts with a TR-cleaning. This fits with reality, where most people would expect a clean train every morning.

Another tendency from the data analysis is the variations in the maximum kilometers between cleanings for the 7 different litra types, compared to the more uniform results from model 2 in Section 7.2. It is seen in Table 4.2 that some litra types are allowed a longer interval of kilometers between each cleaning than others in DSB's solution. This might be based on the amount of passengers on these train rides.

Both of the above tendencies are to be implemented in the model, so the resulting cleaning schedule comes closer to the schedule used by DSB, so it is clear that the set cleaning standard is complied with. Thus this model will ensure that a TR-cleaning is performed first time possible each day, if possible, and that the trains are bounded on KD individually, based on the maximum kilometers between cleaning per litra type from Table 4.2.

8.1 Forcing a TR-cleaning every day

It is not always possible to clean a train after the first train ride in a day. This is either because the train arrives at a station where it is not possible to clean or that the cleaning time for a TR-cleaning for the specific litra type is longer than the stop time at the stop. Thus the model should be able to force a TR-cleaning the first time possible on each day, if ever. Given a day with train rides where the first possible stop to clean with a TR-cleaning is in the evening, then this will be the forced TR-cleaning of the day.

A set H is introduced, to identify when a TR-cleaning should be forced. H is defined by a set of indices that each represent the first train ride each day, where it is possible to clean and where the stop time is equal to or greater than the time for a TR-cleaning on the arrival station. The days are distinguished by seeing when the date changes. Going through each train ride of a day chronologically, checking if $St_i \geq Tr_i$ and if $Pc_i = 1$. When the first occurrence of this is found, i will be a part of the set H, and the next day is searched. If a day does not have a ride where it is possible to set a TR-cleaning on an arrival station, no indices from the day will be in H.

The constraint is set:

$$Xt_i \geq 1 \quad \forall i \in H \tag{8.1}$$

This will ensure that a TR-cleaning is performed on the arrival station for each train ride that corresponds to the indices $i \in H$.

Some days there might not be an occurrence where a TR-cleaning is possible at all. In these instances the model will rely on the constraints introduced in the previous models, to set OR-cleanings when possible and needed. As described in the data description in Chapter 4 some of the trains start their week on the last ride of the 5th of March, since the train arrives on its arrival station on the 6th of March. This day will not be taken into consideration as a day where a TR-cleaning can be forced to be performed. This means that this constraint will only apply for the days from the 6th through the 12th of March.

8.2 Changing the upper bound on *kilometers of dirtiness* based on litra type

So far the upper bound on the kilometers of dirtiness C has been chosen based on the maximum kilometers between cleanings on all of the litra types, as seen in the top row of Table 4.2. The same table shows that this number varies up to 34.5 % between the different litra types, suggesting that DSB has different cleaning standards pr litra type. Model 3 is expanded to take this into consideration by using the C-column in the revised data set, defined in Chapter 4. This column holds the maximum kilometers between cleanings in DSB's solution for each of the corresponding litra types. An example can be seen in Table 4.3b. Hence the constraint that bounds KD is not changed, but the value of C is changed to no longer being a constant, but taken from the C-column, meaning that each train will be constraint by different C_i-values depending on the litra type of the train.

8.3 The model

The final model, an expansion on model 2 including the new constraint and the change in definition of the value C, can be seen here.

$$
\begin{array}{lll}
\textbf{Min} & \sum_{i\in D} Xt_i \cdot TR_i + Xo_i \cdot OR_i & \\
 & & \\
\textit{s.t.} & Xt_i \cdot TR_i + Xo_i \cdot OR_i \leq St_i, & \forall i \in D \\
 & Xt_i + Xo_i \leq Pc_i, & \forall i \in D \\
 & & \\
 & Zt_i \leq Xt_i \cdot M & \forall i \in D \\
 & Zo_i \leq Xo_i \cdot M & \forall i \in D \\
 & Zt_i + Zo_i \leq KD_{i-1} & \forall i \in E \\
 & Zt_i \geq KD_{i-1} - (1 - Xt_{i-1}) \cdot M & \forall i \in E \\
 & Zo_i \geq KD_{i-1} - (1 - Xo_{i-1}) \cdot M & \forall i \in E \\
 & & \\
 & KD_1 = Km_1 & \\
 & KD_i = Km_i & \forall i \in F \\
 & & \\
 & KD_i \geq KD_{i-1} + Km_i - (Zt_i + q \cdot Zo_i) & \forall i \in E \\
 & KD_i \leq KD_{i-1} + Km_i - (Zt_i + q \cdot Zo_i) & \forall i \in E \\
 & KD_i \leq C_i & \forall i \in E \\
 & & \\
 & Xt_i = Xt_j & \forall (i,j) \in G \\
 & Xo_i = Xo_j & \forall (i,j) \in G \\
 & & \\
 & Xt_i \geq 1 & \forall i \in H \\
 & & \\
 & 0 \leq Zt_i \leq C_i & \forall i \in D \\
 & 0 \leq Zo_i \leq C_i & \forall i \in D \\
 & KD_i \geq 0 & \forall i \in D \\
 & Xt_i \in \{0,1\} & \forall i \in D \\
 & Xo_i \in \{0,1\} & \forall i \in D
\end{array}
\tag{8.2}
$$

9 Experiments and results on model 3

9.1 Sensitivity experiment

In the results from model 1 and model 2 it can be seen that there are several litra types which are never cleaned with a TR-cleaning in the two solutions. These solutions are based on the factor $q = \frac{1}{2}$. This implies that two OR-cleanings are the same as one TR-cleaning, which is not an accurate conclusion based on the description of the two cleaning types in Chapter 2. OR-cleanings only contain a part of the cleaning tasks in the TR-cleaning, meaning that a TR-cleaning should be performed from time to time. When there are litra types that are never cleaned with a TR-cleaning, it means that some cleaning tasks are never performed on those trains, like washing the floor etc. A cleaning schedule consisting of only or mostly OR-cleanings would therefore not be preferable, since the thorough cleanings are needed.

The constant q is defined in Equation 5.6 and implemented in 5.12, as being the constant that defines how much the OR-cleaning cleans compared to the TR-cleaning. After implementing the constraint forcing a TR-cleaning first time possible each day, it is desired to examine how sensitive model 3 is to changes in the value of q. This, because the value of q is not easily quantifiable and therefore can be hard to determine.
It is known that the value of q impacts the calculation of the kilometer of dirtiness KD. The higher q is, the more is allowed to be deducted from KD when an OR-cleaning is chosen, even though the amount of cleaning is the same. This means that when the value of q is increased, the impact on KD of a TR-cleaning and an OR-cleaning becomes closer to each other.

To measure the sensitivity of q the objective value and the change in the TR-ratio, meaning the percentage of TR-cleanings out of all the cleanings are used. In Table 9.1 the results from the experiments are listed.

Table 9.1: The TR-cleaning ratio and the objective value relative to the q value. The objective value is given in minutes of cleaning.

	DSB's solution	$q = \frac{1}{4}$	$q = \frac{1}{3}$	$q = \frac{1}{2}$	$q = \frac{2}{3}$	$q = \frac{3}{4}$
Ratio of TR-cleanings	71.9 %	Infeasible	78.1 %	74.5 %	71.6 %	66.8 %
Object value [min]	-	-	138,329	129,569	121,458	113,775

Clearly there is a point where the value of q gets too low for the model to have a feasible solution. This is seen when $q = \frac{1}{4}$. The reason for the infeasibility of the problem, is that there is not time enough for TR-cleanings on a stretch of consecutive train rides and the OR-cleanings cannot subtract enough from KD, due to the low value of q, to comply with the upper bound. As an example, suppose the upper bound of KD is 50 kilometers. The train has just been cleaned at stop 0 and a TR-cleaning was performed, thus $KD_0 = 0$. The two following train rides covers respectively 40 km. and 25 km. At the stop between the two rides, corresponding to stop 1, there is only time for an OR-cleaning. This means $\frac{1}{4}$ of the driven kilometers on the first train ride will be subtracted and $KD_1 = 30$. Since the next train ride covers 25 km. KD_3 will become higher than the bound of 50. Hence there is no feasible solution here.

Since the objective function is minimizing the cost of the cleanings measured in time, the solution will consist of more OR-cleanings the higher the value of q, and the TR-ratio will fall. This can also be seen in Table 9.1.

From the table it can be seen that the solution is not very sensitive to changes in the q-value in regards to the TR-ratio. With big changes in q there's only a few percent change in the ratio. This is because of the constraint forcing a TR-cleaning every day, which leaves little room for change, since the feasible set has decreased so much. A bigger change is seen in the objective value when changing the value of q. The objective value is more affected, because when q is increased, more kilometers are subtracted from KD, but to the same cost of an OR-cleaning. Overall this means that, even though the price is cheaper for at higher value of q, there are fewer cleanings in the resulting schedule.

Because of the lack of sensitivity in the TR-ratio, the value of q is set to equal $\frac{1}{2}$ for the rest of this report, but should be further determined by DSB.

9.2 Results

The results of the cleaning schedule that model 3 constructs are as seen in Table 9.2.

Table 9.2: Analysis on the solution from model 3. Shows the number of cleanings as well as the maximum, minimum and mean kilometers between the cleanings, for the solution to model 3.

	Cleanings [No.]	TR-cleanings [No.]	OR-cleanings [No.]	Max. [km]	Mean [km]	Min. [km]
All litra types	1,538	1,146	392	1,574.90	768.07	5.20
Litra type ERF	261	240	21	1,574.90	887.74	5.20
Litra type ETS	53	30	23	1,242.60	629.80	5.20
Litra type ICA	610	466	144	1,516.80	815.15	11.80
Litra type MGA	193	141	52	1,290.50	557.04	14.90
Litra type ABS	95	57	38	1,170.20	787.53	5.20
Litra type B	231	155	76	1,170.20	817.68	5.20
Litra type BK	95	57	38	1,170.20	787.72	5.20

From the constraint that forces a TR-cleaning every day, there should at least be performed 1,792 TR-cleanings in the week, as this would be one each day for each train. As seen in Table 9.2 the cleaning schedule constructed by model 3 only consists of 1,146 TR-cleaning in the week. This is due to some days where there never is the possibility for a TR-cleaning, because of a combination of cleaning crews not being available at the stations and limited stop time.

To compare with the analysis of the cleaning schedule that was provided from DSB, Table 4.2 from Section 4.3 is repeated here, now referred to as 9.3.

Table 9.3: The number of cleanings, as well as the maximum, minimum and mean kilometers between the cleanings, for the solution given by DSB.

	Cleanings [No.]	TR-cleaning [No.]	OR-cleanings [No.]	Max. [km]	Mean [km]	Min. [km]
All litra types	2,206	1,876	730	1,574.90	523.33	3.10
Litra type ERF	599	285	314	1,574.90	395.86	17.00
Litra type ETS	63	57	6	1,242.60	573.40	5.20
Litra type ICA	1,113	781	332	1,518.80	500.02	3.10
Litra type MGA	335	262	73	1,290.50	393.31	113.70
Litra type ABS	110	109	1	1,170.20	794.50	65.00
Litra type B	276	273	3	1,170.20	811.88	65.00
Litra type BK	110	109	1	1,170.20	794.75	65.00

It can clearly be seen that DSB's cleaning schedule contains a lot more cleanings than the schedule that model 3 constructs. In DSB's schedule there are 2,206 cleanings performed in a week, held against 1,538 times in the solution from model 3. This is a result of model 3's objective function minimizing the number of cleanings with an upper bound on KD being DSB's maximum kilometers between cleanings on each litra type. In this way the solution from model 3 never surpasses the maximum kilometers between cleanings on each litra type, but it results in a larger difference in the mean kilometers between cleanings. The mean is higher because the cleaning schedule from model 3 allows that every time KD becomes zero there can be a maximum number of kilometers without cleaning, before a cleaning has to be performed again. The model will strive for the highest value of KD as possible, since it will be optimal, as its objective is to minimize the time spent cleaning. This means that the model 3 solution cleans a lot less than DSB's solution, but the trains are never dirtier in the model 3 solution than the worst case dirtiness of DSB's solution.

It can be seen in Table 9.2 and 9.3, that the differences between the two solutions are more present in some litra types than others. Especially the litra types ERF and ICA, that are being cleaned approximately twice as much in DSB's solution, than the solution model 3 constructs. Looking at the litra types ETS, ABS, B and BK, it can be seen that the two schedules are more alike, in regards to the total number of cleanings. Though the ratio of OR- and TR-cleanings is very different, this can also be seen in Table 9.5. Out of all the cleanings that are performed on the previously mentioned litra types, it is almost exclusively TR-cleanings in DSB's solution, where the solution in Table 9.2 has a more equally distributed ratio. This indicates that DSB might have different approaches on how they schedule cleanings on the different litra types. This can also be seen in the mean kilometers between cleanings on these litra types being very close to each other in the two solutions, where the mean kilometers between cleanings on ERF, ICA and MGA differs a lot more between the two solutions.

Table 9.4: TR-cleaning ratio. This gives the percentage of TR-cleanings of all the cleanings.

	Model 1	Model 2	Model 3	DSB
Ratio of TR-cleanings	4.7 %	4.9%	74.5 %	71.9 %

The q-value was set equal to $\frac{1}{2}$ in Section 9.1, and with this and the addition of the constraint that sets a TR-cleaning every day if possible, it was ensured that the TR-ratio in the solution from model 3 was as close to the ratio in the solution from DSB.

Even though the overall TR-ratio are very close between model 3's solution and DSB's solution, it is clearly seen in Table 9.2 and Table 9.3 that the ratios differ significantly more between the litra types in Table 9.3 compared to ratios between litra types in Table 9.2.

Table 9.5: TR-ratio out of all cleanings for each litra types for DSB's and model 3's solution.

	Model 3 solution	DSB's solution
Litra type ERF	91.50 %	47.58 %
Litra type ETS	56.60 %	90.48 %
Litra type ICA	76.40 %	70.17 %
Litra type MGA	73.05 %	78.20 %
Litra type ABS	60.00 %	99.09 %
Litra type B	67.10 %	98.91 %
Litra type BK	60.00 %	99.09 %

In Table 9.5 it can be seen that even though the TR-ratio is somewhat the same in the two solutions, the distribution on the different litra types are very different. In DSB's solution, almost all of the OR-cleanings are performed on trains of litra types ERF, ICA and MGA, and almost non of them on ABS, B and BK. In the solution model 3 constructs the litra types ABS, B and BK all have TR-ratios between 60 and 67.1 percent, which is a lot less than in DSB's solution where they are approximately 99 %.

To compare the price of the cleaning schedules, measured in time, it is necessary to compute the objective value of the schedule given by DSB. This is done the same way as the objective value in the solution from model 3 is calculated. The following was found:

$$\begin{array}{rll} \textit{Objective value DSB:} & 212,820.00 & \textit{[minutes]} \\ \textit{Objective value model 3:} & 129,560.00 & \textit{[minutes]} \end{array} \tag{9.1}$$

From this it is seen that the cost of the cleaning schedule that DSB has provided is significantly higher than the cost of the schedule that model 3 produces. The solution has a cost of 39 % less than DSB's. This corresponds to the fact that when more cleanings are performed the price is higher, since cleaning time and cost has been given to be proportional.

10 Discussion

Our model is made to meet some standards of DSB, as we have derived them from the data analysis. These may not be the most important standards for DSB though. DSB might prioritize some litra types to be cleaner than others, some routes based on passenger numbers to be cleaned more, or they may wish to either spread out, or concentrate the stations on which the cleanings are performed. We have decided to focus on their upper bounds on kilometers between cleanings, but they may prefer that the solution matches the mean value mostly and do not care about the outliers as much. This means our model and solution is based on the assumptions we made in our data analysis, and therefore may not meet some of the focus areas of DSB's scheduling.

Since our problem contains over 25,000 decision variables, the feasible solution space is very big which lead to problems with run time. This problem is handled in model 2 by setting a run time limit and evaluating the duality gap of the best feasible solution. This time limit is set based on the graph in Figure 7.1, describing run time compared to the duality gap. This could also have been set by talking to DSB about how much time they usually have available and what time limit would suit them. The approach would then be to obtain the best solution to the problem within that time limit.

When expanding to model 3, the run time problem is automatically handled by adding constraints, which reduces the feasible solution space, thus reducing the enumeration tree and reducing the run time. This is what entails the run time problem not being present in solving model 3, since the added constraints of fixing a TR-cleaning every day, reduces the feasible solution space so much, that the run time limit is not used.

The constraint implemented in Equation 8.1 is based on the definition of the amount H. This definition entails some risks for the solution quality. This, because the first time each day where a TR-cleaning is possible, may occur on one of the last arrival stations of the day. This occurrence combined with a cleaning first thing in the morning on the next day, will give rise to unnecessary cleaning. A solution to this could be setting a constraint not allowing two cleanings in a row, except when a train ride is over some length. This would avoid unnecessary cleaning consecutively, thus making the schedule cheaper, but could also entail infeasibility by decreasing the feasible solution space to much. By keeping the solution space bigger, more solutions can be found, which can result in a cheaper, and therefore better, optimal solution. Another solution could be defining H to exclude all train rides at the end of a day thus also entail more days where a TR-cleaning never happens, and it would lower the TR-ratio of all cleanings.

After finding that DSB's solution contains a difference in cleaning of the different litra types in Section 4.3, we implemented a difference in the upper bound on the kilometers of dirtiness, denoted C, based on the litra types, in model 3 Chapter 8. This was done to emulate the differences seen in Section 4.3. Looking at the results in Section 9.2 of the model where this was implemented, it can be seen that there is a 39% difference between the amount on cleaning in the solution our model produced and DSB's solution. Some litra types especially seem to be cleaned a lot more in DSB's solution than in the model 3 solution. This could be emulated by implementing a lower bound on the number of cleanings based on the litra type, and added in a subsequent model if we had more time.

The difference in cleaning standards of the litra types in DSB's solution may be caused by different amounts of passengers on the rides performed by the different litra types. Some litra types may travel in bigger cities or on routes used by more people. For this report we did not get access to that information and have thus treated every litra type and every ride as getting equally dirty pr. kilometer driven. Having this information could change the models a lot and could be why DSB's solution is so different from ours in regards to some of the litra types.

It can also be seen in DSB's solution that there are some clusters of cleanings happening consecutively. This could be caused by the way their solution is partly manually assembled or by DSB modelling around the passenger numbers, like the knowledge that some routes are more travelled than others. If we had this information our model could be expanded with regards to this by changing the q-value, defined in Chapter 5, so it is decreased when the passenger numbers are higher and increased when they are lower. That way an OR-cleaning would be less impact full when a train has lots of passengers and a TR-cleaning will more often be necessary. Other ways to include passenger numbers could be forcing a TR-cleaning whenever a train ride with many passengers has happened or by setting different upper bounds on KD based on passenger numbers.

An assumptions of having an unlimited amount of cleaning crews available, on each station where cleanings can happen, where made. If this is not the case then a further expansion of the constraints ensuring equal cleaning for connected trains, from Equation 6.1 should be made. This would include that the time used on cleaning a train should be multiplied by the number of trains that are connected, and the stop time required for cleaning would be a lot higher. This could induce that the model avoids choosing cleanings when trains are connected, whereas it is now unaffected by the connected trains, except for only needing enough stop time for the train that takes the most time to clean.

11 Conclusion

For further work on this project, it would be crucial to look at how to handle infeasible solutions. An example of this can be seen in table 9.1, where a low value of q induces infeasibility. An infeasible solution can either be handled by raising q, making an OR-cleaning subtract more from KD, or by raising C, allowing more kilometers of dirtiness between cleanings. It can also be handled by introducing a way the model will allow some cleanings, even though there is not quite enough time for them. Another addition that could be made, if we had the data, is adding an upper bound on the number of cleaning crews for each station, since this model assumes an unlimited amount of cleaning crews.

In this project we have modelled the train cleaning problem of DSB with various levels of complexity. By creating a simple model, and expanding it two times, we have succeeded in finding a way to plan DSB's cleaning schedule in an automated manner. This was the problem we set out to accomplish.

In the analysis of the solution given by DSB, we chose the cleaning standard to be the maximum kilometers between cleanings. This means that the amount of kilometers between cleanings on the scheduled trains in our solution will never exceed the amount of kilometers that DSB's cleaning schedule allows, between two cleanings. We end up with a cheaper solution than theirs, that cleans 39 % less but with the same upper bound of dirtiness.

Bibliography

[1] DSB. *Company Profile - About DSB*. 2023. URL: https://www.dsb.dk/en/about-dsb/.
[2] DSB Drift - Køreplanlægning. *Tog i Danmark - køreplaner*. 2022. URL: https://www.dsb.dk/globalassets/pdf/koereplaner/intercity/k23/togidk-k23.pdf.
[3] DSB. *Tog i drift*. 2023. URL: https://www.dsb.dk/om-dsb/virksomheden/tog-i-drift/.
[4] Gurobi Optimization. *Mixed Integer Programing (MIP) - A primer on the basics*. 2023. URL: https://www.gurobi.com/resources/mixed-integer-programming-mip-a-primer-on-the-basics/.
[5] Laurence A. Wolsey. *Integer programming*. 2nd ed. 111 River Street, Hoboken, NJ 07030, USA: John Wiley Sons, 2021.
[6] Laurence A. Wolsey. *Integer programming*. 1st ed. 111 River Street, Hoboken, NJ 07030, USA: John Wiley Sons, 2021.
[7] Stefan Røpke. *Slides from 42101 Introduction to Operations Research, Lesson 6, Slides 26-27*. 2022.

Technical
University of
Denmark

Akademivej, Building 358
2800 Kgs. Lyngby
Tlf. 4525 1700

www.man.dtu.dk