Department of Precision and Microsystems Engineering

Spread-based Stochastic High Frequency Railway Optimization

D.J. van Rooijen, MSc

Report no	:	2020.046
Coach	:	P. Mohajerin Esfahani, PhD
Professor	:	H. HosseinNia, PhD
Specialisation	:	MSD
Type of report	:	Thesis
Date	:	12 October 2020

TU Delft Delft University of Technology

Challenge the future

Spread-based Stochastic High Frequency Railway Optimization

D. J. van Rooijen, MSc

Abstract

This paper outlines a mathematical model for the creation of a robust timetable for the optimization of the lightrail network in Rotterdam and The Hague. A novel spread-based goal function focuses on minimizing wait time as opposed to delay. The situation is modeled as a Quadratic Stochastic Programming problem, and solved by employing Monte Carlo techniques to make the Quadratic Program deterministic. The solution informs what the optimal rate of train departures per hour is and provides a timetable that has over 120% more train capacity than the status quo, which has a higher delay resistance than the current timetable. For the inevitable unpredicted delay, a real-time implementation of this algorithm is used dynamically to create a new optimal timetable on the fly.

October 12, 2020

Supervisors:

P. Mohajerin Esfahani, PhD (Delft University of Technology)

E. Tuinenburg, MSc (Siemens Mobility)

Contents

1 Introduction

1.1 Relevance

Railway networks are an important means of transport, providing a large part of society with a cost-effective way of getting around, as well as being a vital part of other logistical operations, like material transport. The National Market and Capacity Analysis by the Dutch government outlines a growth in capacity of 27% to 45% of railway km travelled by train as shown in fig. 1. Nationale Spoorwegen, however, has shown an increase in travelers of 4.6% in the first half of 2019, leading to reaching the theoretical maximum capacity of the network significantly sooner[1]. RET, the company that runs the metro line in Rotterdam, has been warning of a capacity issue as well[5]. Because of limited resources, it is necessary for current railway networks to be used more efficiently. However, standard avenues of increasing efficiency are already being exhausted. Trains can no longer increase in length because they won't fit on some of the busy stations, and a maximum rate of train departures is being reached at several points, taking into account current safety systems.

The underlying works of a train system are extremely complex, and research into more efficient use extends in several directions. Often, this type of research focuses on more efficient driving strategies for single trains on a single piece of track. Other strategies include research into more efficient scheduling, combining train scheduling with other means of public transit, and decreasing headway on railway sections, mostly by changing safety rules.

Maintaining proper safety, while maybe changing the safety mechanism on rails, is a challenge in increasing current track utilization. The old system in the Netherlands, Automatische Trein Beveiliging (ATB), works well and is safe. However, it is unable to handle the necessary future capacity. Furthermore, having specific and incompatible safety systems for different EU countries is inconvenient. Therefore, the European Rail Research Institute (ERRI) has created the European Rail Traffic Management System (ERTMS). Instead of relying on the signals next to the tracks, a train operator can get all the information necessary to decide his optimal speed to his cabin instantly. The system also takes safety precautions, taking over the brakes in dangerous situations.

The implementation of ERTMS is extremely costly. The Dutch ministry has

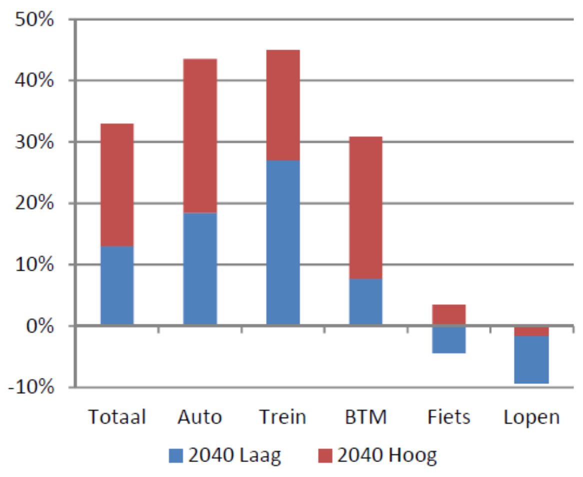

Figure 1: Development of amount of km travelled between 2014 and 2040 in the Netherlands. [21]

budgeted for 2.4 billion euro's for its implementation on the main network, and plans on adding on 100 million every year starting in 2031. Smaller networks, such as the Randstadrail in The Hague and Rotterdam, don't get the benefit of this new system, as it takes a lot of resources. With the huge growth of inhabitants in this region, a cost effective solution for the improvement of the efficiency of this network is necessary.

1.2 State of the Art

A lot of time has been poured into railway analysis. The first analytical model of a single piece of track for two way traffic stems from 1966 [12]. Several analytical models followed, studying the same problem of two-way traffic on a single railway. To this first model several additions were made, such as different types of trains, or variable demand. In the modern day, train networks have become vastly more complicated. The planning of a network is generally subdivided into different levels of tasks [7]. Sorted for length of timescale, these are Network Planning, Line planning, Timetabling, Vehicle Scheduling and Crew Scheduling. The first two focus on a long timescale: where do we put a network, and how do we plan lines between them? Pricing structures, as well as necessary capacity estimates, fall under these categories. The other categories focus on a medium timescale of months to weeks, and focus on planning daily scheduling. By definition, the timetables constructed allow for faultless operation daily, barring unforeseen delays. Other research areas focus on, for example, train headway control [23], error propagation [20], or energy efficiency [14]. Research for network utilization tends to focus on either headway control or timetabling

1.2.1 Headway Control

A way of increasing the network efficiency is allowing the trains to drive closer to each other. An active version, such as implemented in ERTMS, is a form of Positive Train Control (PTC). PTC systems employ GPS for the handling of the distance between trains, also called headway, for avoidance of possible collisions, and for the maintaining of speed limits. An active version of headway is proposed by Zhao et. al. [23]. Their system makes use of the characteristics of the trains on the network to maken sure collisions never happen, by allowing trains to only get as close as their deceleration profile allows. Unfortunately, PTC solutions are prohibitively expensive for a lot of smaller, existing networks.

1.2.2 Timetabling

Proper timetabling allows a significant increase in the efficiency of a network. For simple, low capacity systems this can be trivial. Drive the train as often as necessary for the required demand. However, in the real world systems have variable capacity, different types of trains on the tracks, or only one set of tracks between two stations instead of two. The timetable resulting from this scheduling problem should not only be viable, it should be robust against inevitable disturbances in the network.

Disturbances in a train network are generally an external stochastic event that slow down, or completely stop a train. These are called 'primary disturbances'. Eventually, the system will become so chaotic that a second train will be delayed because of the first delayed train. These are then 'secondary disturbances'. To deal with delays, operators can add time buffers to a proposed timetable. Increasing these time buffers will increase the robustness of the network, at the cost of railway efficiency and capacity. There is a balance to be attained. Another issue is that robustness buffers only protect against relatively small disturbances. In the event of a large event active traffic control is necessary, at which point these buffers become useless. Furthermore, it is argued that allowing a process to take more time will result in the possibility of the process taking up more time in its realization, filling up that time [9].

Most research takes the Periodic Event Scheduling Problem (PESP) formulation from Serafini and Ukovich [22] as the basis for their model. Multiple mathematically computed timetables that are taken into physical use are based on this model, such as that for the Dutch railway network, and the Berlin underground network [8]. PESP poses timetabling as a Mixed Integer Programming model, and handles headway by creating departure times that take them into account. Often this model uses a goal function that minimizes the delay experienced by travellers.

An addition to these models is adding robustness to the network. Making the network completely robust, and accounting for any possible delay would require a significant buffer at every station, which is infeasible. Several solutions have been used for timetable creation. One approach to this is to add a penalty for robustness into the goal function of the problem [19]. Others have employed stochastic programming approaches to set these buffers, where an initial timetable is modified with these calculated buffers [17]. The 'Light Robustness' approach allows for slight deviations in the solution to make solving significantly faster [10], while performing only slightly worse than the expensive stochastic programming technique[11]. This paper combines the robust stochastic optimization techniques to create a robust timetable, and draws inspiration from the idea of recoverable robustness[18], where the possible recovery possibilities are taken into account. Instead of taking them into account initially, however, the model is built in such a way that the recovery for any delay can be calculated by the system.

1.3 Paper Overview

The exact problem will be outlined in Section 2. After this, this paper will follow the structure of the three-phase model that is being used. Section 3 will outline the Quadratic Programming approach to solve the Stochastic Programming problem, after which Section 4 will talk about transforming the solution into a timetable and its performance. Section 5 will touch on how this timetable can be adapted in real-time to deal with delays that inevitably appear in a network. Finally, Section 6 concludes.

2 Problem Overview

2.1 Situation Overview

In 2019, the RET and HTM transported 142 million, and 87 million passengers in their railed vehicles in a year [15]. RET's metro passengers, as well as HTM's tram passengers grew more than the predicted growth of 2% per year. During rush hour, 60 trams drive around carrying passengers to their destinations. Aside from safe scheduling and vigilant train drivers, an underlying system, ATB, maintains the safety in the network by making sure trains don't get too close to each other. For this purpose, the tracks are divided up into several blocks, called safety sections. If one safety section is occupied, the section before and after it are not allowed to be occupied. To this effect, the system maintains a database of the location of every train by means of axle counters. If a train nears a block that it's not allowed to enter, it will encounter a red sign. If the driver chooses to ignore this red sign, the system will take action, and stop the train.

The stretch of rail that will be the focus of the research is called 'De Samenloop' or 'The Convergence', and is located between station Leidscheveen and station Den Haag Laan van NOI. This piece of track is occupied by both the Rotterdam metro system, as well as the Zoetermeer lightrail system. At its inception, the plan was to have 30 trains driving on this stretch of rail per hour each way. However, because of technical and planning issues, this has proven to be impossible in practice. To solve these problems, tram company HTM proposes several solutions[4]:

- Create another track to no longer have entanglement between the two rails.
- Create 'Timestops', allowing the trains to enter The Convergence at a specific time, allowing for a more predictable schedule.
- Other solutions, such as 'have less delay causing disturbances' and 'cancel a few lines'

These options are fairly expensive. Building another line would cost around 200 million euros, and waiting stations would add a significant delay to travel times. Another complication in this system is that the different trams have different floor heights. Every station along the Convergence has two places travelers can embark or disembark. The first place will have a low floor, and the second a high floor. These are in different safety blocks, and the only place in the system where two concurrent blocks are allowed to be occupied at the same time. This means that while driving onto the Convergence trains have the option to halt at the same time, if they are ordered properly.

With 30 trains each hour, the time between departures will be expected to be around 2 minutes per train. With 40 per minute, just 90 seconds. At that point, it no longer matters if a train is slightly delayed, since you would just be able to get on the next one two minutes later. Therefore, minimizing delay times, as generally done in the state of the art, is not especially relevant for this issue. When a delay does occur on a piece of track, it is more important that trains don't bunch up. From the perspective of a traveler that would result in a long wait time, followed by a lot of trains in rapid succession. Therefore, instead of minimizing the delay, the spread of the train departures will be optimized.

The goal of the optimization is to create a robust timetable, by distributing departure buffers around the track as efficiently as possible. Departure buffers, in this case, are the amount of time a train will, on average, superfluously wait on a station. Increasing this buffer will improve robustness, as a delay can be neutralized by eating into this buffer. However, there is a balance, because letting the trains wait too long will decrease the capacity of the network, increase travel times, and require more trains.

By creating a dynamic real time implementation of the model for delays, whether they are because of boarding delays at the station or because of system malfunctions, the system has the possibility to efficiently get back to its initial setup. The proposed Timestop stations might become unnecessary, as the network adjusts the buffers in real time.

2.2 Data Overview

The section of rail in this paper is driven on by two companies, HTM with line 3 and 4, and RET with line E. These lines drive onto the convergence at section W861, and lines 3 and 4 split off at section W815. The order of tracks can be seen in table 1. Travel times will be taken from a data set provided by Siemens Mobility b.v., detailing times each train has taken in traversing specific sections from 9 through 18 February 2020. These 100k data points can be seen in fig. 2. The data set lacked information on one

Station	Section	3	4	E	Station	Section	3	4	E
Nootdorp	0901			x		301	x	x	x
	0322			x		W853	x	x	x
	0319			x		0219	x	x	x
	W893			x	Forepark	0217	x	x	x
	W891			x		W845	x	x	x
	0317			x		0215	x	x	x
	0315			x		W841	x	x	x
Centrum-West	0713	x	x			213	x	x	x
	W869	x	x			0211	x	x	x
	W867	x	x		Leidschendam-Voorburg	0209	x	x	x
	0711	x	x			0207	x	x	x
	0709	x	x			0205	x	x	x
Voorweg Laag	0707	x	x		Voorburg 't Loo	0203	x	x	x
	0705	x	x			0201	x	x	x
	0703	x	x			0123	x	x	x
	0701	x	x			0121	x	x	x
	0311	x	x			0119	x	x	x
	W865	x	x			W817	x	x	x
	W863	x	x		Laan van NOI	0117	x	x	x
	0309	x	x			0115	x	x	x
	0307	x	x			W815	x	x	x
	0323	x	x			0107			x
	W861	x	x	x		0105			x
	0306	x	x	x		W813			x
Leidschenveen	0304	x	x	x		W811			x
	W859	x	x	x	Den Haag Centraal	0101			x

Table 1: Tracks and sections driven by current train lines

section between 0311 and 0701. The time on track of this section was interpolated for every train in the data and is shown as 'interpolated 0311'.

2.3 Model Setup

This paper will handle the timetabling as a three-part model. The first two parts will be similar to [16], in the sense that a timetable will be made, which can then be used in a simulation model. Initially, a stochastic programming problem will be posed, and solved as a quadratic programming problem by randomly selecting values for our random variables from our database. The quadratic problem will then be solved multiple times, and the resulting solutions will be used to create a timetable.

Using this timetable, the next model can be run: a simulation. This simulation will adhere to the rules of the safety system implemented on the Randstadrail, and will allow for a comparison between different timetables. The time a train takes to finish a track will be determined by randomly sampling the time on track database.

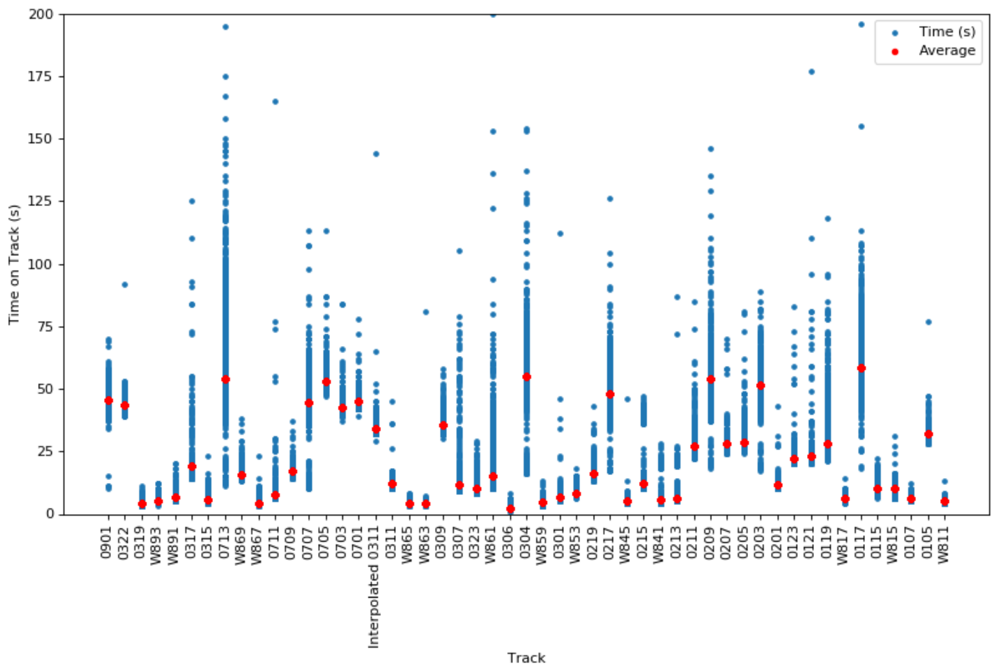

Figure 2: Time on Track data for the safety sections in the dataset provided by Siemens Mobility b.v.

Lastly, a dynamic real time solution will be implemented. During the simulation, there will eventually be a train that experiences a significant delay. The third part of the model will use a version of the initial quadratic programming problem with extra constraints to calculate the most efficiently spread departure buffers to deal with delays.

3 Quadratic Programming

The first stage of this paper is the quadratic programming solution used to solve for optimal buffers. First, the variables in the problem will be detailed, after which the model will be explained.

3.1 Model variables

$[1, ..., N]$	Set of all trains in the system	(1)
$[1, ..., R]$	Set of all rails in the system	(2)
$n \in$	$[1, ..., N]$ Train number	(3)
$r \in$	$[1, ..., R]$ Railway section in the system	(4)
R^n	Numbers of railway sections for train n	(5)
r^n	1xR^n Size vectors of railway sections for each train n	(6)
D^n	1xR^n Size vectors of departure times for each train n	(7)
T	Time between departures at a station	(8)
L	Time a train has to spend on a station safety section at minimum	(9)
S^n	1xR^n Size vectors containing station information for each train n	(10)
F	1xR Size vector of fixed, minimum time needed to cross a safety section	(11)
ξ	1xR Size vector of stochastic time needed to cross a safety section	(12)
b^n	1xR^n Size vectors of departure buffers per railway r for each train n	(13)
M^n	Train type marker for train n	(14)
Y	Amount of optimization iterations	(15)

Variables of note in this model are the station information vectors S^n, departure buffer vectors d^n, and the train type markers M^n. The station information vector is a way of keeping track of which track sections are stations, as well as the order in which trains pass stations. Entries for the vector are as follows:

$$S_i^n = \begin{cases} 1, & \text{if } r_i^n \text{ is a station for train n} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The train type markers M^n have the function of regulating what company a train is from. On the Convergence, each station has a spot for company 1 at the start of the station, and company 2 at the end. The train type marker variable is to take into account that different train orders can be more efficient at halting.

$$M^n = \begin{cases} 1, & \text{if train n is from train company 1} \\ 0, & \text{if train n is from train company 2.} \end{cases} \tag{17}$$

3.2 Quadratic Program

The railway network, which is a stochastic problem, will be modeled using a quadratic programming approach. For ease of notation, the departure time D of train n at railway r_i^n or r_{i-1}^n will be noted as D_r^n and D_{r-1}^n respectively, instead of $D_{r_i^n}^n$ and $D_{r_{i-1}^n}^n$.

$$\min W \tag{18}$$

where

$$W = \sum_{n=1}^{N}\sum_{r=0}^{R} S_r^n (D_r^n - D_r^{n-1} - T)^2 \tag{19}$$

$s.t.$

$$D_r^n = D_{r-1}^n + F_r + b_r^n \tag{20}$$

$$D_r^n = (-S_r^n + 1)(D_{r-1}^n + F_r + \xi_r^n) \tag{21}$$

$$D_r^n \geq S_r^n (D_{r-1}^n + F_r + \xi_r^n) \tag{22}$$

$$D_r^n \geq S_r^n (D_{r-1}^n + L) \tag{23}$$

$$D_r^n \geq D_{r+2}^{n-1}\ (-S_{r+2}^{n-1} + 1) \tag{24}$$

$$D_r^n \geq D_{r+2}^{n-1}\ S_{r+2}^{n-1}\ (-f_m(n, n-1) + 1) \tag{25}$$

$$D_r^n \geq D_r^{n-1}\ S_r^{n-1}\ f_m(n, n-1) \tag{26}$$

Here D_r^n and d_r^n, the departure time and departure buffer of train n at railway r, are control variables. T, S, F, and f_m are set by the problem and are fixed variables. ξ_r^n, the delay at rail r for train n, is a stochastic variable. To make this stochastic programming problem solvable, ξ will be randomly sampled from the available data. This way, the above program becomes deterministic. Solutions are calculated multiple times, after which results can be averaged to create an estimate of the solution.

3.2.1 Goal Function

The objective of the program is to minimize the square of the time difference between departures at a station minus a goal time.

In a normal train network, and in the state of the art, generally the delay from a set departure time is minimized [16]. The PESP, which the model in this paper is based on, is not defined in a way that it has a solution. Adding a goal function and a solving method adds this. A goal function for delays that has been used is

$$minimize\ D = \frac{\sum_{e \in E} \sum_{r=1}^{R} \sum_{h=1}^{H} w_e D_{e,r,h}}{|E_a| \cdot R \cdot H} \tag{27}$$

where E is a set of events, R is a number of simulations, H is the number of hours the simulation ran, w_e are the weights given to certain delays, and D are the delays. These delays originate from a constraint somewhere else in the definition in [16]

$$\bar{v}_{e,r,h} - (v_e + h \cdot T) \leq D_{e,r,h} \tag{28}$$

Where $\bar{v}_{e,r,h}$ is the actual event time, and v_e is the planned event time.

This is also considered a KPI by train operators [3]. However, in practice, lightrail networks have a fairly high percentage of trains arriving several minutes late. When driving as many as 40 trains per hour on the network, it no longer matters whether your train is on time, it matters how long you have to wait until the next one. Therefore, we choose the square of the time delta to optimize the spread of trains waiting. By implementing a goal time, the parameters like demand can be taken into account by the network operator.

The function of the station vector S_{ij}^{n} in the goal function is to ascertain that only departure times at stations are taken into account in the optimization, as the time at which trains depart the safety sections is not relevant to the traveler, and would increase the complexity of the problem significantly.

3.2.2 Constraints

The constraints are a result of the systems that make sure trains don't crash into each other, and of the properties of the system.

First, constraint 20 sets the control variable d_n^r to be equal to the difference between necessary departure time, and actual departure time. These d_r^n variables will be used eventually to create the timetable resulting from the model.

Constraints 21 and 22 are the physical leaving constraints for the safety sections and the stations, respectively. At a safety section, a train has to keep driving, and has to leave as soon as it is able to. On a station, a train can choose to wait longer than its minimum departure time if it improves the spread of train departures. Constraint 23 sets the amount of time a train should minimally spend on a station safety section, which is observed from the data to be $L = 50$

The last three constraints regulate the safety systems of the network. Constraint 24 is for a safety section that doesn't have a station following it closely. Constraint 25 is for a safety section that does have a station 2 sections ahead, but does not have the trains driving in the proper order to halt at the same time. Effectively, this makes it the same constraint as 24. Finally, constraint 26 is for the train that is allowed to halt at the same time as the train in front of it. This constraint regulates its leave time from a station, so it does not crash into its predecessor.

To achieve the order in the system, constraints 25 and 26 use the following function:

$$f_m(n_1, n_2) = \frac{(M^{n_1} - M^{n_2}) + (M^{n_1} - M^{n_2})^2}{2} \tag{29}$$

$$f_m(n_1, n_2) = \begin{cases} 0, & \text{if } M^{n_1} = M^{n_2} = 0 \\ 0, & \text{if } M^{n_1} = M^{n_2} = 1 \\ 0, & \text{if } M^{n_1} = 0 \;\&\; M^{n_2} = 1 \\ 1, & \text{if } M^{n_1} = 1 \;\&\; M^{n_2} = 0 \end{cases} \tag{30}$$

3.3 Solution

A railway network is subject to a significant amount of stochastic delays and variables. In this problem, this is modeled with the variable ξ. This is inputted into the quadratic optimization problem by randomly sampling an appropriate amount of delays from the data set. By solving the quadratic program A times, we can calculate expected values of the departure buffers per railway, which can be used for the creation of a robust timetable.

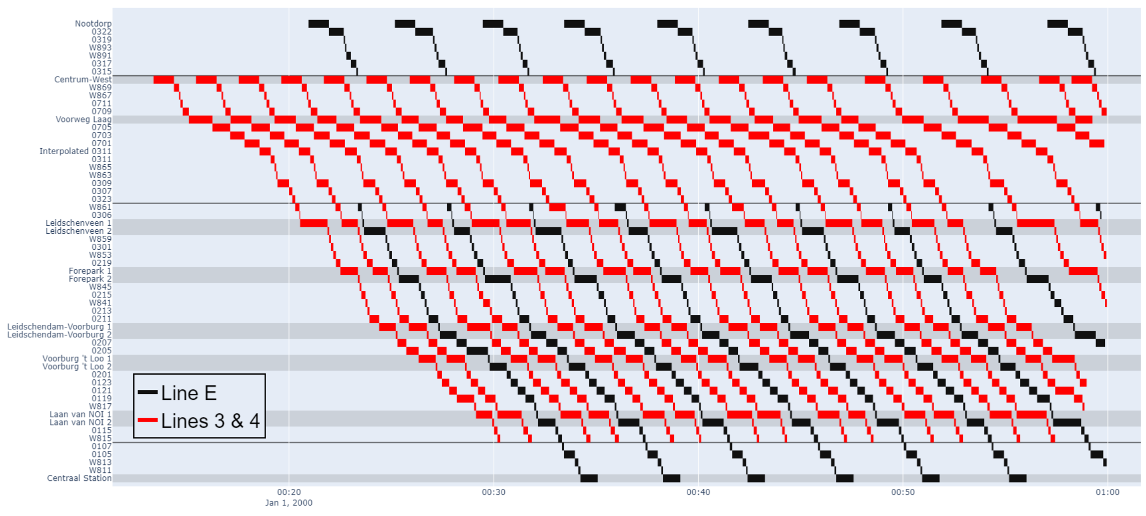

Figure 3: Solution of the quadratic programming problem

The program is rewritten into standard formulation to be inputted into quadratic program solving package CVXOPT in Python.

$$\text{minimize} \quad \frac{1}{2}x^T P x + q^T x \tag{31}$$

$$\text{subject to} \quad Gx \leq h \tag{32}$$

$$Ax = b \tag{33}$$

$$s \geq 0 \tag{34}$$

Which automatically solves the dual problem

$$\text{maximize} \quad \frac{1}{2}(q + G^T z + A^T y)^T P^T (q + G^T z + A^T y) - h^T z - b^T y \tag{35}$$

$$\text{subject to} \quad q + G^T z + A^T y \in \text{range}(P) \tag{36}$$

$$s \geq 0 \tag{37}$$

Running the problem for $N = 100$ trains and $Y = 100$ iterations results in the driving patterns seen in fig. 3, as well as the buffer times shown in table 2. Nootdorp, Centrum-West and Den Haag centraal are two departing stations, and an endstation. As such, they do not provide useful information. The other stations, however, do provide insightful data. It can be seen that stations Leidschendam-Voorburg and Voorburg 't Loo have a fairly low average buffer as compared to the others. Apparently, the optimization network prefers the trains to run through those stations as quickly as possible. These also have the lowest standard deviation of planned driving times.

Station	Time on Track	Buffer	Time STD
Nootdorp	60	13	2
Centrum-West	60	15	21
Voorweg Laag	69	25	10
Leidschenveen	76	24	11
Forepark	65	18	7
Leidschendam-Voorburg	59	10	8
Voorburg 't Loo	60	11	6
Laan van NOI	71	16	16
Centraal Station	69	59	0

Table 2: Average time on track and buffer times resulting from the optimization of the quadratic program for N=100, A=100, as well as driving time standard deviations

Analysing the solution of the dual problem gives us values for the multipliers of the constraints in the primal program. For every station, the constraints are averaged for every train and every simulation, resulting in fig. 4 and 5.

In fig. 4, the average Lagrange multiplier values of constraints 22 and 23 are plotted. It can again be seen that some constraints have significantly higher average constraint multipliers than others. For example, the two stations that had a low average buffer in table 3 have a fairly high constraint multiplier. It can be interpreted as a more important station to save time on, as every won second on this station will contribute more to the overall network spread than at another station.

Figure 5 shows the importance of constraints at different safety sections. These constraints are a little more difficult to interpret, but useful nonetheless. One thing to note is that not all safety sections are present, as only the safety sections with an average multiplier above zero have been plotted. The figure shows that somehow section W863 is very important to pass as quickly as possible, even though its average track time is only 4.3 seconds. It will be difficult to improve there. However, on sections like 0705, the time improvements are the second most important, while the time on track is the second longest. It shows that this track could be a prime candidate for improvements.

The size of the multipliers can be important for scheduling, as it shows where an operator could add a buffer time with the lowest possible cost. Moreover, it gives a very nice insight into the sensitivity of different parts of the tracks, and where its weaknesses are. As an example, in this example the system could significantly benefit from increasing the speed on track 0705, or considering changing the safety system layout in that area.

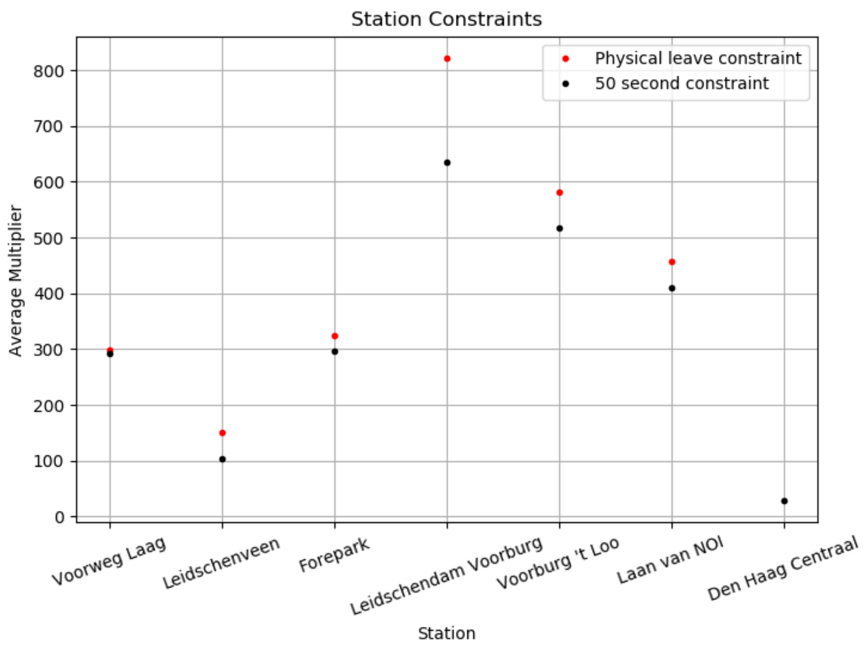

Figure 4: Average Lagrange multipliers for the constraints pertaining to the stations in formulae 22 and 23 for the physical leave constraint, and the 50 second constraint respectively.

3.4 Network Capacity

In calculating a timetable for the optimal usage of a railway network, the question 'what is the ideal rate of trains driving on this track?' quickly arises. Solving for an optimal spread arguably answers this question, as having trains leave closer together than optimal will result in delays, and therefore a worse spread. In the solution, trains leave every 90 seconds from every station on the Convergence. Therefore, it can be argued that this piece of track can robustly accommodate 40 trains per hour from every station. A significant increase of over 120% from the current 18.

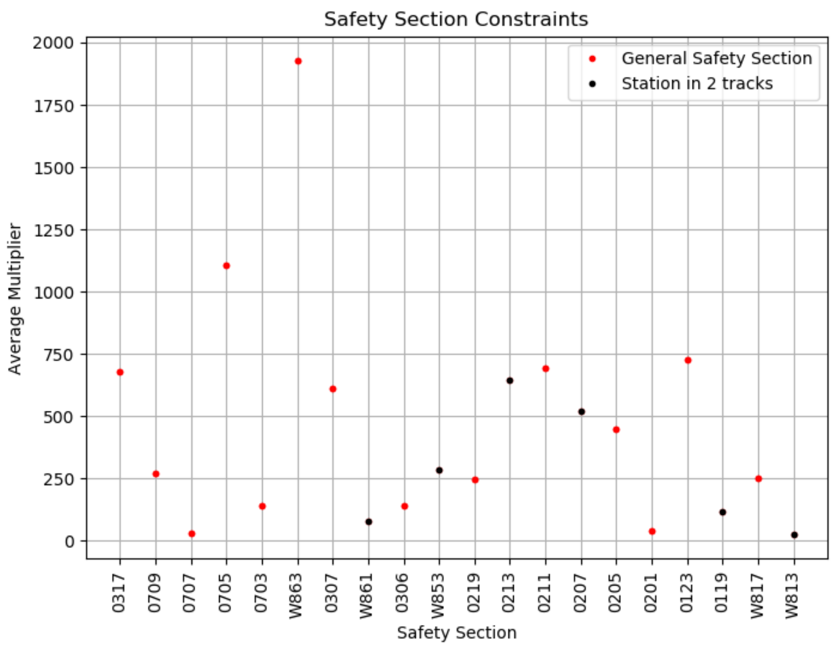

Figure 5: Average Lagrange multipliers for the constraints pertaining to the safety sections in formulae 24 and 25 for the 'general safety section' constraint and 'station in 2 tracks constraints' respectively.

Section	Time	Section	Time
0317	19.2	0219	16.1
0709	17.0	0213	6.1
0707	44.3	0211	27.0
0705	52.8	0207	28.0
0703	42.7	0205	28.5
W863	4.3	0201	11.9
0307	11.7	0123	22.1
W861	15.4	0119	27.9
0306	2.4	W817	6.0
W853	8.4	W813	5.0

Table 3: Average time on track for safety sections in fig. 5

3.5 Yellow Light Considerations

Something to take into account in this situation is the yellow lights on the network. The optimization model, as it currently stands, does not take into account the effect of a train driving into a yellow light. However, the question is whether this is important. Table 4 shows the railway sections where a yellow light delay should have occurred, as well as the size of this delay. It can be seen that the delays incurred are not very frequent, and when they do occur the delay generated is smaller than the buffers shown in table 2. The 90 second spread network is still robust for the incurred yellow light delays.

Track	% of trains	Delay (s)
0201	8%	8
0123	4%	9
0309	2%	4
0307	2%	4

Table 4: Delays created by yellow lights that don't interfere with station in the mathematical optimization problem

The standard deviation of the delays at the safety sections of interest are shown in table table 5. When comparing these standard deviations with the delays in table 4, it can be seen that all of the yellow light delays that have to be dealt with fall inside an interval of 2 standard deviations from the time on track of the tracks corresponding to the yellow light delays. Not only is the original optimization robust against these extra delays, they aren't even an abnormality.

Track	Std (s)
201	4.6
123	6.8
309	7.2
307	2.0

Table 5: Time on track standard deviations from the mean of delays aligning with those in table 4

Finally, a train operator could choose to change the optimization algorithm to try and minimize hitting the yellow lights. Penalizing a yellow sign like a red sign, and not allowing the train to pass through until the train track three ahead is cleared, is a way to minimize this yellow sign driving behaviour. This changes constraints 24 and 25 to

$$D_r^n \geq D_{r+3}^{n-1} \left(-S_{r+3}^{n-1} + 1\right) \tag{38}$$

$$D_r^n \geq D_{r+3}^{n-1} \, S_{r+3}^{n-1} \left(-f_m(n, n-1) + 1\right) \tag{39}$$

and results in a new model that finds an optimal spread of 120s. This is still an increase of 67% over the current 18 trains per hour, and coincides with the train goal of 30 per hour initially stipulated for the track. 120 seconds overstates the importance of hitting a yellow sign, and is conservative. Hitting a red sign will stop the train dead in its tracks, and force it to wait. In this case, since it will wait for trains at a station as well, the added constraint adds a comparatively large amount of time onto the network. Hitting a yellow will add a few seconds of travel time to its travel time in its next section, which is often catchable by the buffer.

4 Scheduling

4.1 Setup

After the optimization problem of section 3 has been solved, a good overview of the system and its variables has been attained. The solution of the system, specifically the buffer times, can then be used to create a timetable for the network. Several timetabling methods will be compared.

- The current Randstadrail timetable
- A timetable based on data on average track travel time
- A timetable based on buffers calculated from the quadratic programming optimization model
- A timetable based on optimization model average track time
- A timetable based on optimization model average train time
- A timetable based on optimization model average timetable time

The first of these will be taken from the websites of HTM[2] and RET[6]. The second option is based on the data, and the average travel time for each section. The last four options are all based on the results of the quadratic programming solution of the previous chapter, in three different ways.

4.1.1 Timetable: Average Data

The data based model will be created using the data somewhat naively. The departure times are calculated by taking the average leaving time over the entire dataset. For every time on track data point the fixed travel time F_r for each railway is extracted by searching for the minimum value. The stochastic value for that data point ξ_r is the difference between the time on track and the fixed travel time. The average stochastic delay is calculated as

$$\bar{\xi}_r = \frac{1}{M} \sum_{m=0}^{M} \xi_r^m. \tag{40}$$

where M is the amount of point in the data set. The timetable is subsequently calculated as

$$D^n_{r+x} = D^n_r + \sum_r^{r+x} F_r + \sum_r^{r+x} \bar{\xi}_r + n \cdot T \tag{41}$$

where the departure times D are taken from the data and x is the amount of rails between two stations.

4.1.2 Timetable: Optimization Buffer

After running the optimization detailed in the previous chapter thousands of optimal departure times D have been calculated. After departing from a station, a train will drive over some tracks before reaching the next station, and departing it. Each of these tracks has a fixed and a stochastic driving time, and the station has a fixed time as well. The buffer is the time the train will wait between being able to leave, and actually leave. It can be taken from the decision variables of the mathematical model of the previous chapter through eq. (20). For implementation in a timetable, an average buffer is calculated by summing over every train n and iteration y for every rail r:

$$b_r = \frac{1}{Y}\frac{1}{N}\sum_{y=0}^{Y}\sum_{n=0}^{N}(d^n_r)_y \tag{42}$$

The departure times of the timetable are calculated as

$$D^n_{r+x} = D^n_r + \sum_r^{r+x} F_r + \sum_r^{r+x} \xi^n_r + \sum_r^{r+x} b_r \tag{43}$$

Here $r + x$ signifies the next station track after the station at r.

4.1.3 Timetable: Average Track/Train/Timetable times

The last three timetables are similar, in that they average the solution of the optimization to recommend a timetable. Timetables are calculated by the average departure times per track with regards to, respectively, arriving at the track, the start of the journey of that train, or the start of the simulation. Mathematically this is described as
Average Track:

$$D^n_{r+x} = D^n_r + \frac{1}{Y}\frac{1}{N}\sum_{y=0}^{Y}\sum_{n=0}^{N}((D^{n*}_{r+x})_y - (D^{n*}_r)_y) + n \cdot T \tag{44}$$

			Optimization			
	Current	Data	Buffer	Track	Train	TT
Centrum-West	00:00	00:00	00:00	00:00	00:00	00:00
Voorweg Laag	01:00	01:34	01:53	01:54	01:47	01:46
Leidschenveen	07:00	07:04	07:34	07:35	07:15	07:14
Forepark	08:00	08:30	09:15	09:15	08:52	08:55
Leidschendam-Voorburg	10:00	10:20	11:09	11:10	10:48	10:50
Voorburg 't Loo	12:00	12:08	13:05	13:05	12:42	12:44
Laan van NOI	14:00	14:37	15:47	15:48	15:26	15:28
Centraal Station	17:00	15:41	17:52	17:53	17:31	17:33

Table 6: Train schedules for train 4 using different scheduling methods

			Optimization			
	Current	Data	Buffer	Track	Train	TT
Centrum-West	-	-	-	-	-	-
Voorweg Laag	-	00:11	00:25	00:25	00:25	00:25
Leidschenveen	-	00:40	00:34	00:45	00:40	00:40
Forepark	-	00:41	00:38	00:49	00:44	00:45
Leidschendam-Voorburg	-	00:44	00:40	00:53	00:48	00:58
Voorburg 't Loo	-	00:45	00:44	00:56	00:53	01:07
Laan van NOI	-	00:51	00:46	01:03	00:56	01:14
Centraal Station	-	00:51	00:46	01:03	00:56	01:14

Table 7: Train standard deviations for train 4 using different scheduling methods

Average Train:

$$D_r^n = D_r^n + \frac{1}{Y}\frac{1}{N}\sum_{y=0}^{Y}\sum_{n=0}^{N}((D_{r+x}^{n*})_y - (D_0^{n*})_y) + n \cdot T \tag{45}$$

Average Timetable:

$$D_r^n = \frac{1}{Y}\frac{1}{N}\sum_{y=0}^{Y}\sum_{n=0}^{N}(D_r^n)_y \tag{46}$$

Where D_r^{n*} variables with star are taken from the optimization, and the D_r^n variables without star are the timetable departures

4.2 Realisation

The schedule for the first train for every timetable scheme can be seen in table 6, and the standard deviations corresponding to these times in table 7.

It can be seen that using these timetables, the current schedule takes 17 minutes to complete the track, the data average calculation scheme takes 15 minutes and 41 seconds, and the optimized schemes either take around 17 minutes and 50 seconds, or 17 minutes and 30 seconds, depending on the calculation technique. It seems that calculating timetables taking into account buffers and average track times produce nearly the same timetable. Similarly, average train times and average timetables produce similar results. The mathematical timetables take slightly longer than the current timetable to complete the track. It should be noted that the current timetable is based on minute based departure times, whereas the other ones are based on seconds.

The standard deviations on these average timetable solutions are fairly wide. This is most likely the result of the chaos of a train system, and how that impacts the quadratic programming solution. The solution space for this problem is relatively limited, and this width leaves space for considerations outside of this system.

The train schedules are transformed into a timetable by adding more trains, and offsetting their departure times by a certain time, as shown by the $n \cdot T$ term in equations 41 through 45. An exception to this is the optimization based timetable average, for which the spread is inherent to the calculation. As the optimization has shown an optimal spread occurs at 90 seconds, an example timetable of the Optimization Buffer type is given for that spread in table 8 for 5 trains.

During simulation, any number of trains could be chosen. For calculation time purposes, 100 trains were simulated. The result of the simulations is shown in figures 6 through 9 for spreads of 90s, which is optimal, and 200s, which is currently used for both the current timetable as well as the Average Train timetable. For the other timetables at these spreads, please refer to appendix A. It is difficult to tell how the timetables perform from these images. Therefore, table 9 and table 10 outline the delays incurred at stations and safety sections. Because trains are not allowed to leave the station before having been there for 50 seconds, they will sometimes have to wait after their scheduled departure time has passed, immediately incurring a delay, and an error in the system. It can be seen in table 9 as well as table 10 that there is almost zero delay over the life of a train on average in any mathematical solution, as opposed to the current timetable, where every train incurs 85 seconds of delay on average on a high spread, and 155 seconds on a tight spread. The naive data averaging solution already performs better than this, despite being scheduled to drive through the network 79 seconds faster. This could very well be because of the more efficiently distributed departure buffers. Something else that might play a factor is the current timetable being based on minutes, and the new times being based on seconds.

Station	Line 3	E	4	3	E
Nootdorp		00:06:11			00:10:41
Centrum-West	00:00:00		00:03:00	00:04:30	
Voorweg Laag	00:01:53		00:04:53	00:06:23	
Leidschenveen	00:07:34	00:09:04	00:10:34	00:12:04	00:13:34
Forepark	00:09:15	00:10:44	00:12:15	00:13:45	00:15:14
Leidschendam-Voorburg	00:11:09	00:12:39	00:14:09	00:15:39	00:17:09
Voorburg 't Loo	00:13:05	00:14:34	00:16:05	00:17:35	00:19:04
Laan van NOI	00:15:47	00:17:16	00:18:47	00:20:17	00:21:46
Centraal Station		00:19:21			00:23:51

Table 8: Train timetable for the Optimization Buffer model, with a spread of 90s. The times indicate departure times from the stations.

Timetable	Station Delay (s)	Safety Section Delay (s)
Current	85	0
Data Average	66	0
Buffer	2	0
Average Track	1	0
Average Train	2	0
Average Timetable	2	0

Table 9: Average departure delays per train for a train spread of 200 seconds

Timetable	Station Delay (s)	Safety Section Delay (s)
Current	135	20
Data Average	76	6.2
Buffer	6	11
Average Track	6	12
Average Train	5	9
Average Timetable	7	8

Table 10: Average departure delays per train for a train spread of 90 seconds

4.3 Delay handling

Inevitably, an event will occur where a train is delayed. This might be because of a technical malfunction, a person on a track, or a crash. Since a robust network is better at taking up the impact of these delays, it is a good measure of the robustness of a timetable. Table 11 shows the impact of a delay of 300 seconds on the respective

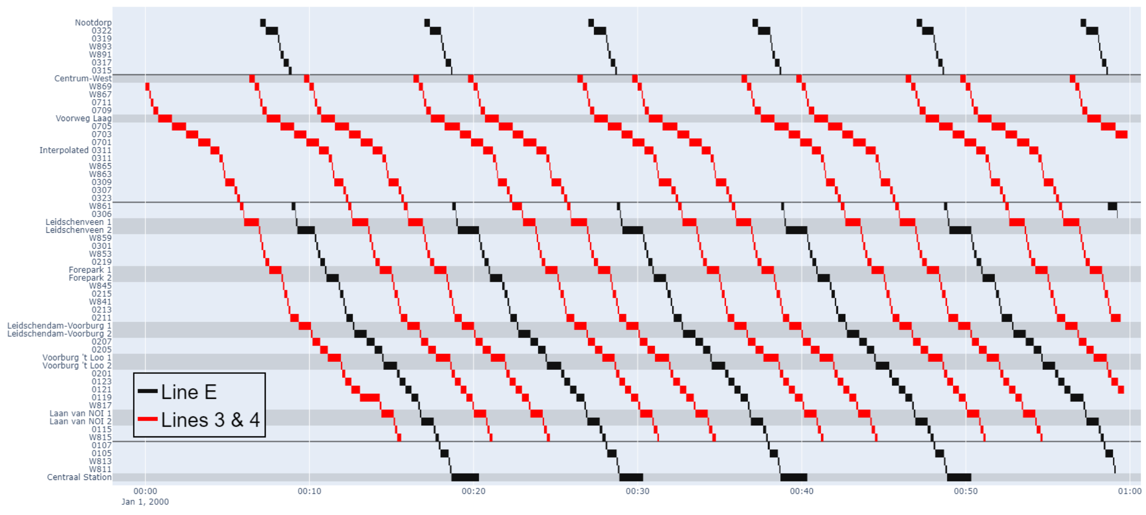

Figure 6: Train simulation using Current timetabling, with trains 200 seconds apart

timetables, for each degree of separation. Note that the Average Timetable model is only evaluated for the 90 second separation, as that is defined in the model. The delay is injected into the simulation for the 9th train at safety section 0213, between stations Forepark and Leidschendam-Voorburg. An example is shown in fig. 10 for an Average Train timetable. It can be seen that it takes around 8 trains including the delayed one before there is a train that will not encounter any red lights because it is too close to the one ahead of it. By analyzing table 11 once again it shows that the mathematical models outperform the current timetable. On the tighter spreads closer to 90 seconds, all of the mathematical models have less impacted trains than the current timetable. For example, the 5 trains difference at 90 second spread between Average Train and Current timetable means that the former recovers 7.5 minutes faster after a 5 minute delay. They take 45 minutes and 52.5 minutes, respectively.

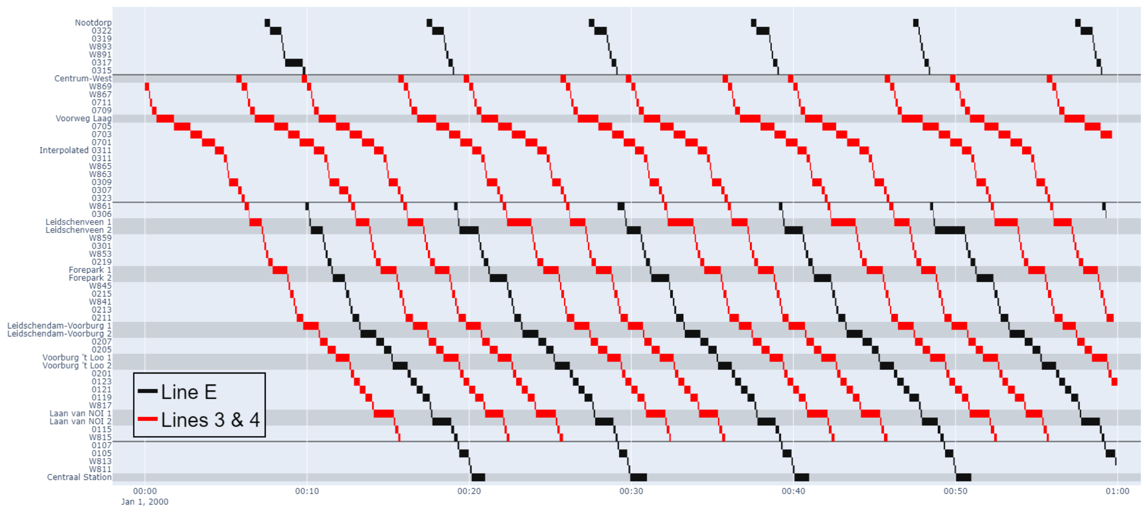

Figure 7: Train simulation using the Average Train timetable, with trains 200 seconds apart

	Spread								
Timetable	80	90	100	110	120	130	140	150	180
Current	No Recovery	35	18	13	10	7	6	8	7
Average Data	No Recovery	34	16	13	9	9	6	6	4
Buffer	No Recovery	31	15	10	8	7	6	5	4
Average Train	No Recovery	29	16	11	9	7	6	6	4
Average Track	No Recovery	30	15	10	8	7	6	6	4
Average TT	No Recovery	30	-	-	-	-	-	-	-

Table 11: Trains before the effects of a delay of 300s recover for spreads of 80 through 180 seconds, for every timetable.

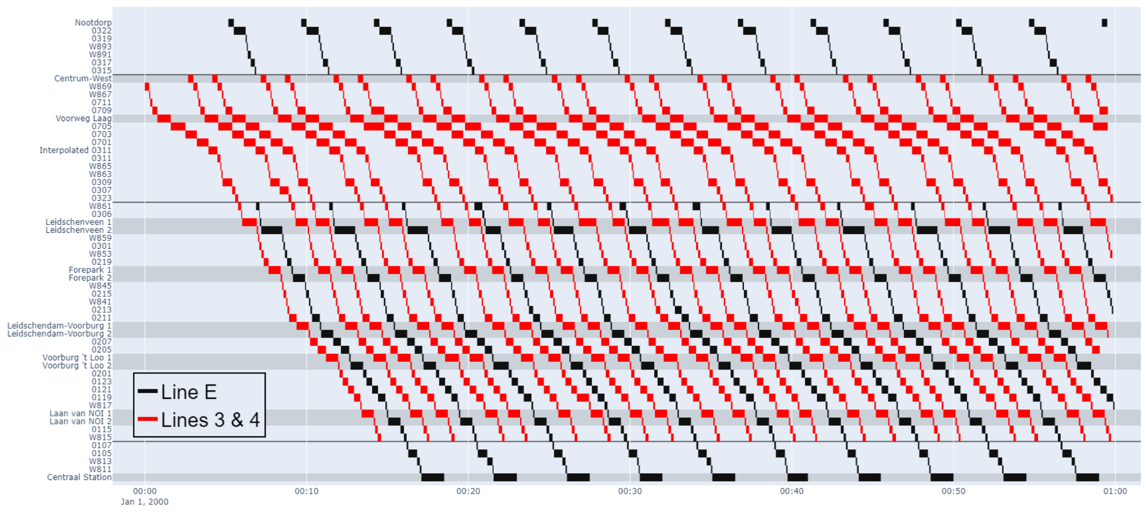

Figure 8: Train simulation using the Current timetable, with trains 90 seconds apart

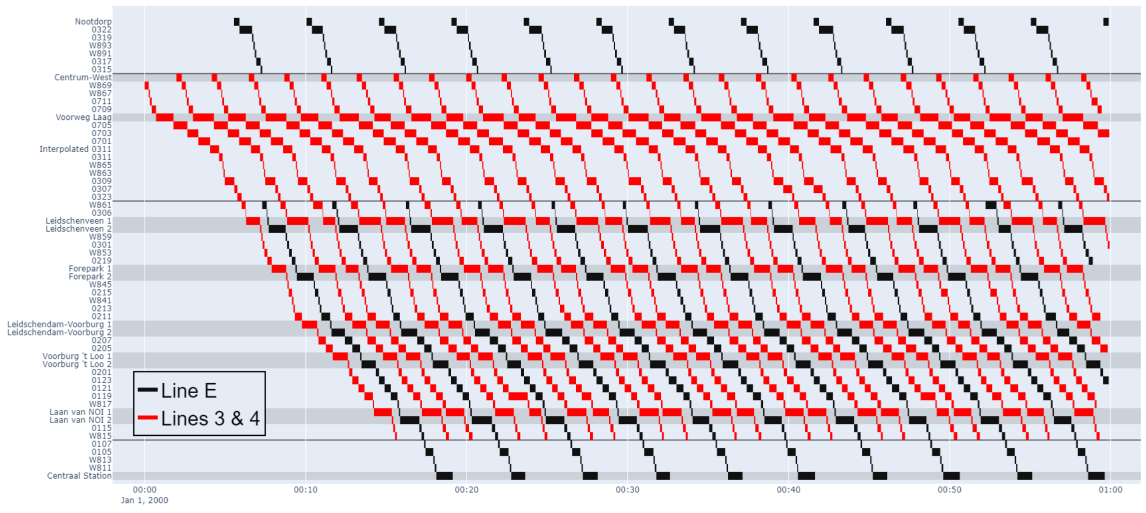

Figure 9: Train simulation using the Average Train timetable, with trains 90 seconds apart

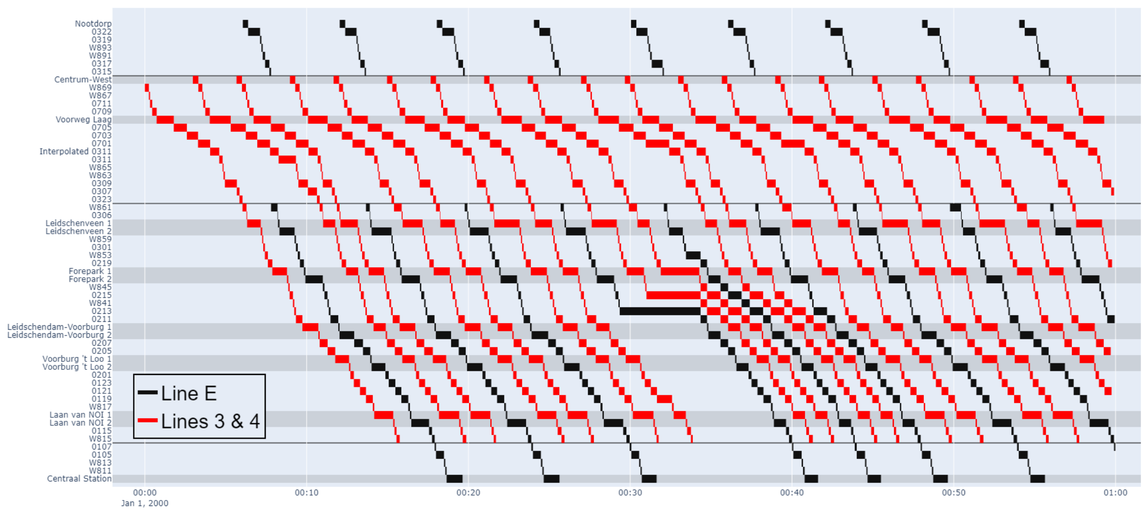

Figure 10: Delay of 300s for an Average Train timetable at a spread of 120s

5 Real-Time Control

The previous chapter has shown that the optimized timetables significantly outperform the current timetable based on how much delay the trains experience at stations, as well as the way delays are handled. The mathematical timetables don't show any significant difference, so this chapter will employ the 'Average Train' timetable. Out of the mathematical models it drives through the stations the fastest, and converting the single-train schedule into a timetable is both trivial as well as versatile. This chapter builds on the delay analysis by rerunning the optimization model as soon as a delay takes place to minimize waiting time at stations.

The same steps are taken as in the previous chapters. Timetables are calculated from an optimization scheme, and put into a simulation algorithm. After 9 trains, a delay is inputted into the system, and the simulation is cancelled. This can be seen in fig. 11. The green line at the 25.5 minute mark indicates the time at which the simulation is cut off. This is the first time at which a train leaves at the station before the delay, and the first time a decision has to be made about when trains should leave.

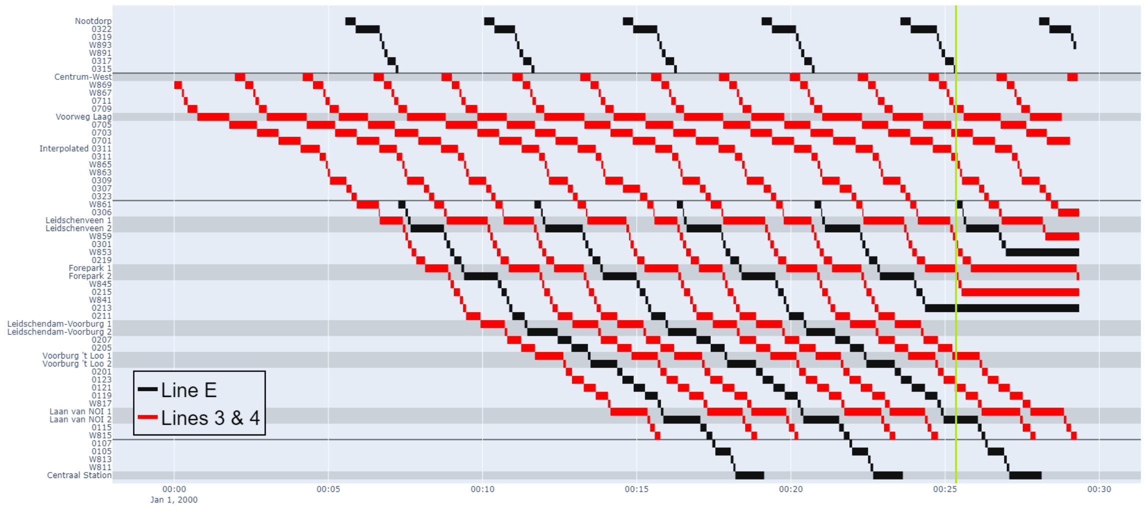

Figure 11: Cancelled simulation after a delay occurs for an Average Train timetable.

This situation is inputted into the quadratic programming model by setting the next departure time at a safety section before a station as fixed in the quadratic programming model. This is done by adding eq. (47) to the constraints listed in eq. (20) through eq. (26). Each train that is already driving needs to have a single point at which its fixed. If more points are defined, the optimization algorithm throws an error. Since the output is a timetable, the actual control that the system has on the network is at the station. Giving it a fixed arrival time at that station is the simplest way of constraining the program.

$$D_s^m = (D_f)_s^m \tag{47}$$

where D_f are the fixed departure times, $s \in r$ are the safety sections before stations where trains arrive after the 25.5 minute mark at the green line in fig. 11, and $m \in n$ are the trains on which this has an impact.

The quadratic programming model is run for Y=10 iterations, and the new timetable is calculated with

$$D_r^n = \frac{1}{Y}\frac{1}{N}\sum_{y=0}^{Y}\sum_{n=0}^{N}(D_r^n)_y \tag{48}$$

like in the Average Timetable timetable model. The timetable from the 300s delay at safety section 0213 can be seen in table 12.

	Line - Train						
Station	E - 10	4 - 11	3 - 12	E - 13	4 - 14	3 - 15	E - 16
Nootdorp	-			28:20			33:01
Centrum-West		-	-		-	27:02	
Voorweg Laag		-	-		26:56	29:01	
Leidschenveen	-	-	29:58	31:20	32:57	34:33	35:48
Forepark	-	30:18	31:40	33:16	34:45	36:06	37:53
Leidschendam-Voorburg	30:45	32:07	33:40	35:12	36:34	38:17	40:03
Voorburg 't Loo	32:47	34:07	35:35	37:05	38:42	40:21	42:02
Laan van NOI	35:25	36:50	38:18	39:56	41:33	43:09	44:45
Centraal Station	37:29	38:55	40:26	41:59	43:36	45:14	46:50

Table 12: Train schedule from the green line at 25.5 minutes in fig. 11. Stripes indicate departure times that don't apply for that train because they are in the past

After the timetable has been calculated it is loaded back into the simulation software, along with the post-delay train locations. This simulation then shows the impact of the real time control. Figure 12 shows the behaviour of the network when no attempts to fix it are made. Trains will keep driving and eventually interfere with each other. Looking at stations following the delay it can be seen that they are piled up, and often hitting red lights. This means they will be stopping and starting a lot, and rider comfort will go down. Spread is also not at an optimal level. Figure 13 shows the implementation of the timetable in table 12. Some trains inevitably have to wait on safety sections, but most waiting times take place at a station. By holding trains the stop-and-go nature of a delay in the train network has been mitigated, resulting in more passenger comfort, as well as a power usage decrease. It can be seen in fig. 13 that after 8 trains the network is back to a normal position.

The way the real time timetable changes the network can be analysed in table 13 and table 14, of which the former is the difference between train travel time in seconds, and the latter the difference in percentages. Obviously train 9 is the same for both networks. At the delayed train the differences creep in. The real-time solution holds this train and a few subsequent trains back some seconds to mitigate the delay, as well as get an optimal spread in the times immediately following the delay. This makes trains, as per the tables, between 3 and 11 percent slower through the track than when you just let them drive. However, after this, the trains are significantly faster. Especially at higher delays the time won is very significant, adding up to a travel time of over 8 minutes on a single train.

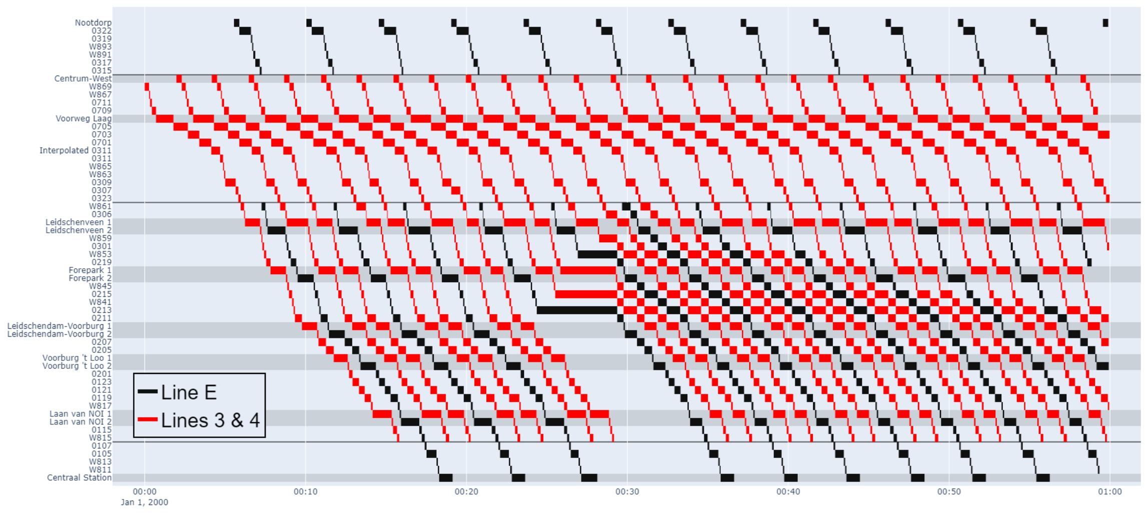

Figure 12: Situation for a spread of 90s, with a delay of 300s at station 0213, with an Average Train timetable, without real-time control

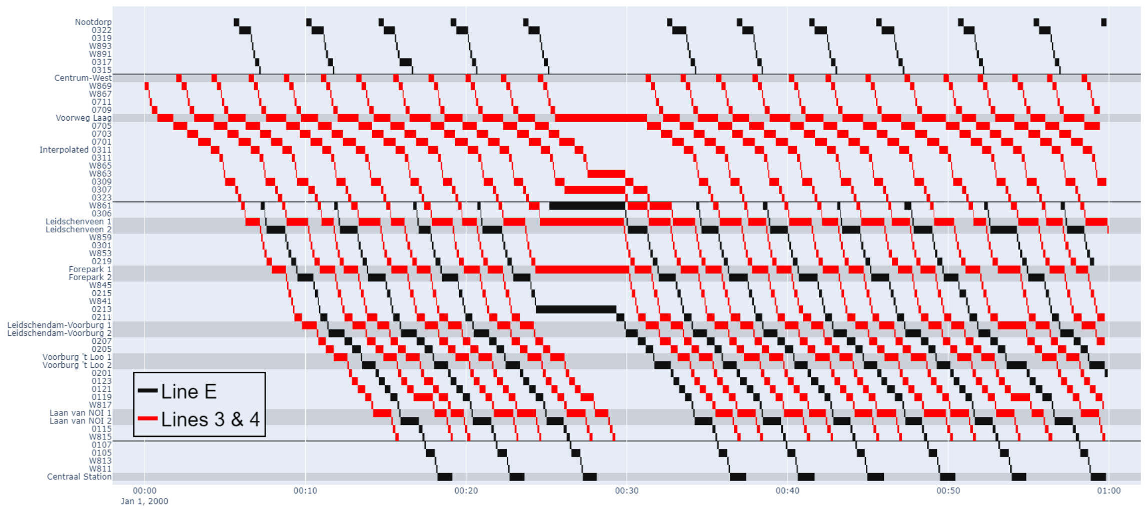

Figure 13: Situation for a spread of 90s, with a delay of 300s at station 0213, with an Average Train timetable, with real-time control

	Line	3	E	4	3	E	4	3	E	4	3	E	4	3
Delay(s)	Train	09	10	11	12	13	14	15	16	17	18	19	20	21
300		0	-88	-77	-90	-107	-99	-115	177	-115	168	175	142	144
360		0	-34	-66	-80	-97	-97	-119	215	-112	212	220	193	214
420		0	-43	-53	-52	-67	-62	-67	284	-133	278	285	259	259
480		0	-66	-40	-63	-90	-86	-112	374	-118	369	341	311	299
540		0	-41	-48	-47	-57	-77	-74	454	-63	429	401	395	387
600		0	-44	-49	-54	-49	-61	-65	489	-95	455	445	449	439

Table 13: Seconds difference between complete train ride from start to finish between Average Train timetable with different delays and the real-time control timetable

	Line	3	E	4	3	E	4	3	E	4	3	E	4	3	E	4
Delay(s)	Train	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23
300		0	-9	-6	-7	-11	-8	-9	18	-9	14	18	12	12	15	8
360		0	-3	-5	-6	-9	-7	-9	21	-8	17	22	15	17	19	15
420		0	-4	-4	-4	-6	-4	-5	26	-10	21	26	19	20	20	15
480		0	-5	-3	-4	-8	-6	-8	32	-8	26	30	22	22	24	19
540		0	-3	-3	-3	-5	-5	-5	36	-4	29	34	27	27	34	26
600		0	-3	-3	-3	-4	-4	-4	38	-6	30	36	29	29	35	27

Table 14: Percentage difference between complete train ride from start to finish between Average Train timetable with different delays and the real-time control timetable

6 Conclusion

This paper outlines a mathematical model for the creation of robust timetables through stochastic optimization techniques, based on earlier work [17]. Adapting the model to the situation and changing the goal function to be applicable to a high frequency railway has provided the system with a way of optimizing the capacity of the network as well as show that optimally 40 per hour is a good departure rate for this network. The real-time solution builds on the state of the art by providing a way to implement extra constraints to find a solution for a delayed situation.

For the current network, implementation of solutions is difficult. Adding additional systems to the current safety system will attract significant regulatory scrutiny, as well as accompanying costs. The power of the real-time solution outlined in this paper is that it is in the form of a timetable. The only thing necessary for implementation is communicating it to the drivers, which can be done using any number of modern communication methods. A driver can have a no-nonsense screen in front of him, outlining the exact time he has to try to leave the station, as well as the time he has to leave the next one. Updates to this screen from the timetabling system are then easily conveyed to the driver.

Implementation in a real-time solution could benefit from research into error sources, or better simulation software. For example, the simulation as programmed does not take into account either positive or negative acceleration. It just assumes that a train will take a certain time to finish a railway section, which has an impact on the way the red light in the safety system interferes with the trains. Furthermore, the network assumes that timetables will be followed as well as possible. However, in practice the human drivers that drive on this stretch of track have a significant amount of influence on how the train drives. An interesting research step would be their effect on delays or the network. Lastly, the errors in the network are currently randomly sampled. However, it is unlikely that errors at one point in the network are completely uncorrelated to other points in the network. A significant amount of research is being done into predictive delay modeling using neural networks and its application, such as DIRECTOR [13]. Implementing a solution such as this to bolster the techniques employed in this paper could result in better results in both scheduling and delay impact management.

A Timetables

A.1 90 second spread

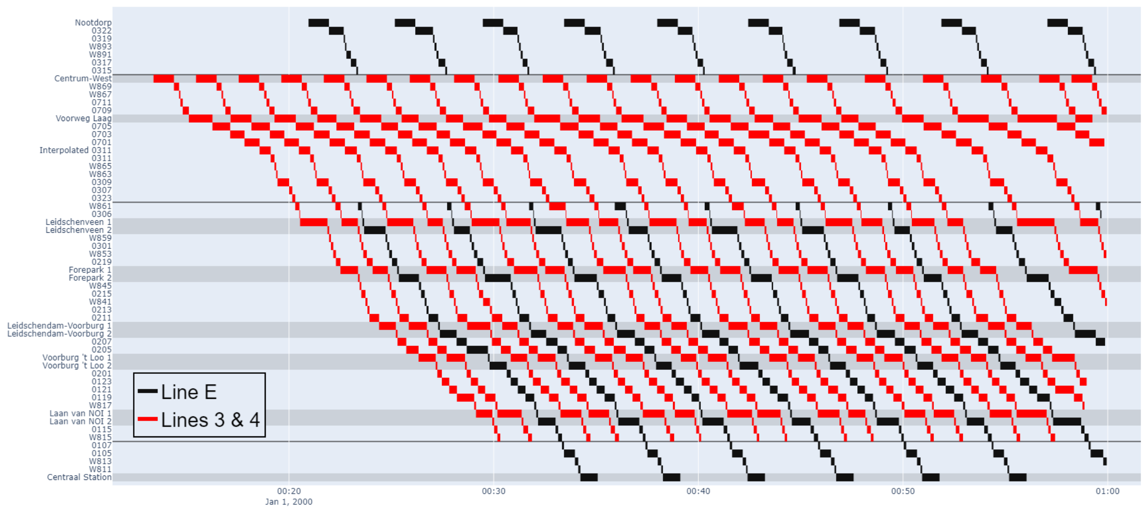

Figure 14: One of the solutions of the quadratic programming for a spread of 90s

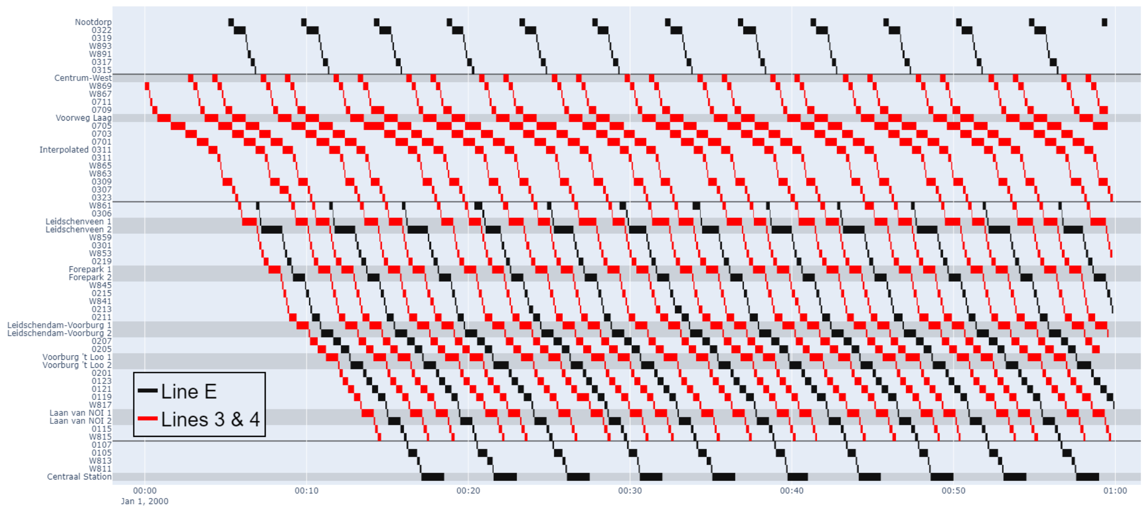

Figure 15: A simulation of the Current timetable for a spread of 90s

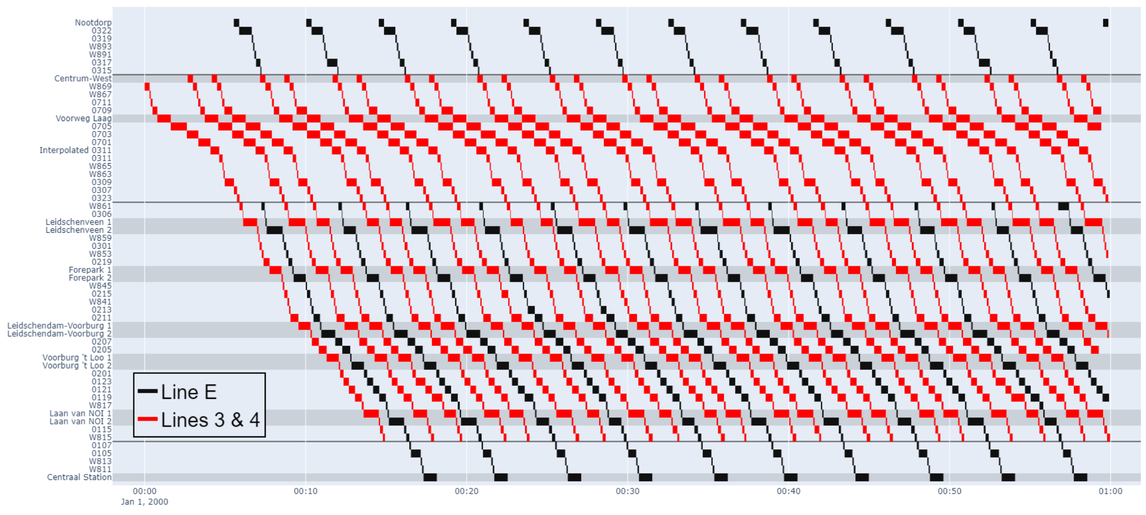

Figure 16: A simulation of the Average Data timetable for a spread of 90s

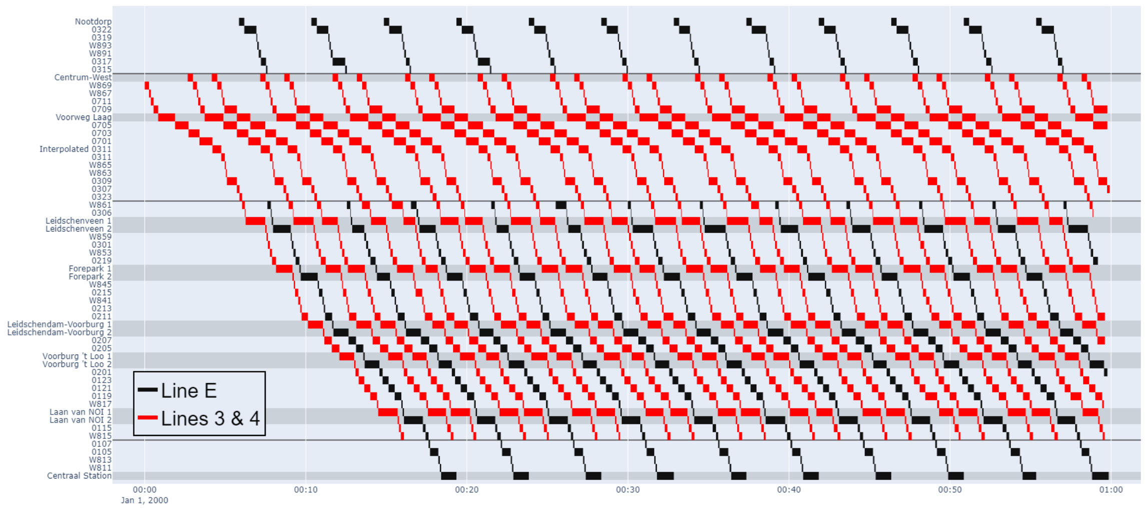

Figure 17: A simulation of the Buffer timetable for a spread of 90s

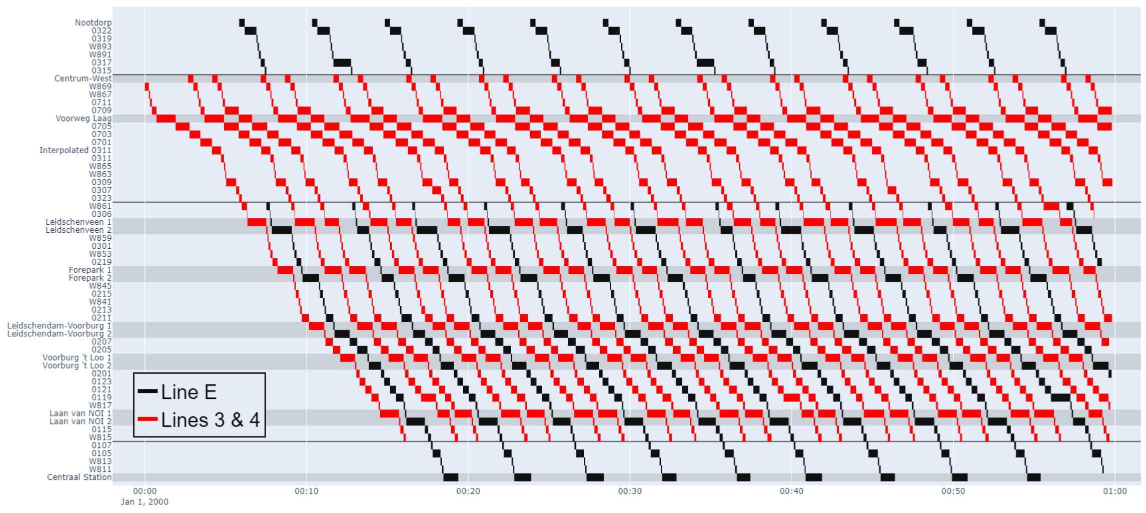

Figure 18: A simulation of the Average Track timetable for a spread of 90s

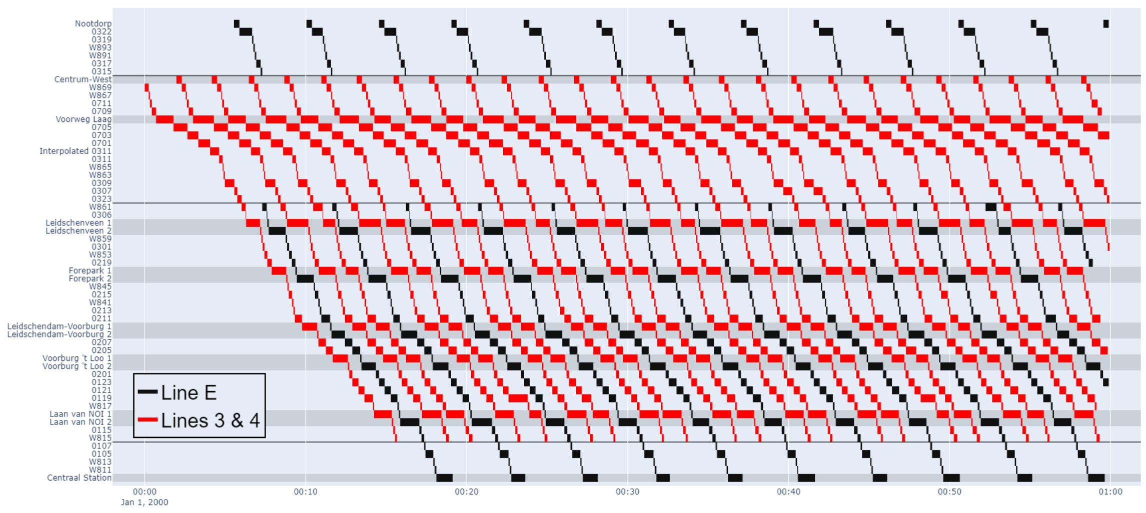

Figure 19: A simulation of the Average Train timetable for a spread of 90s

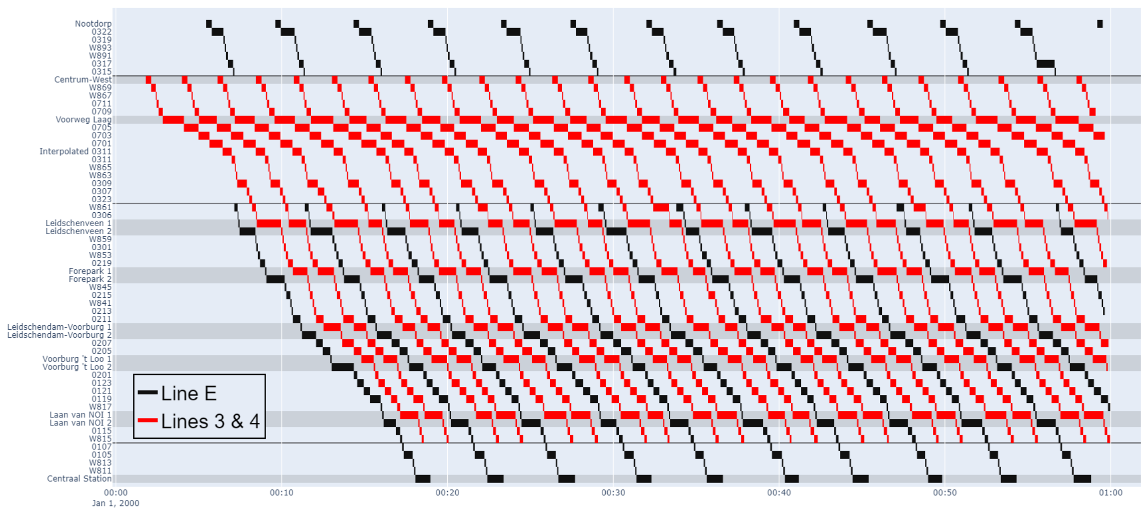

Figure 20: A simulation of the Average Timetable timetable for a spread of 90s

A.2 200 second spread

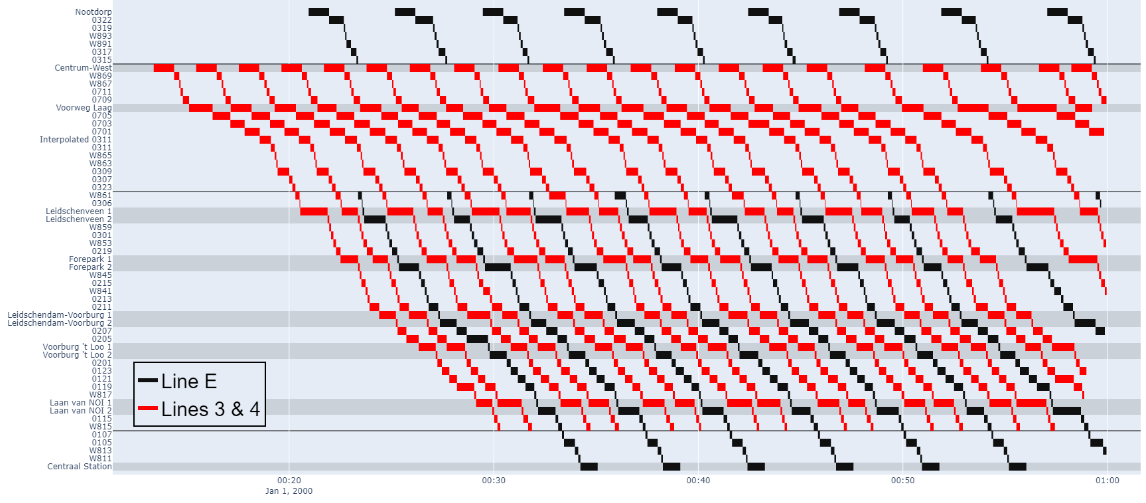

Figure 21: One of the solutions of the quadratic programming for a spread of 200s

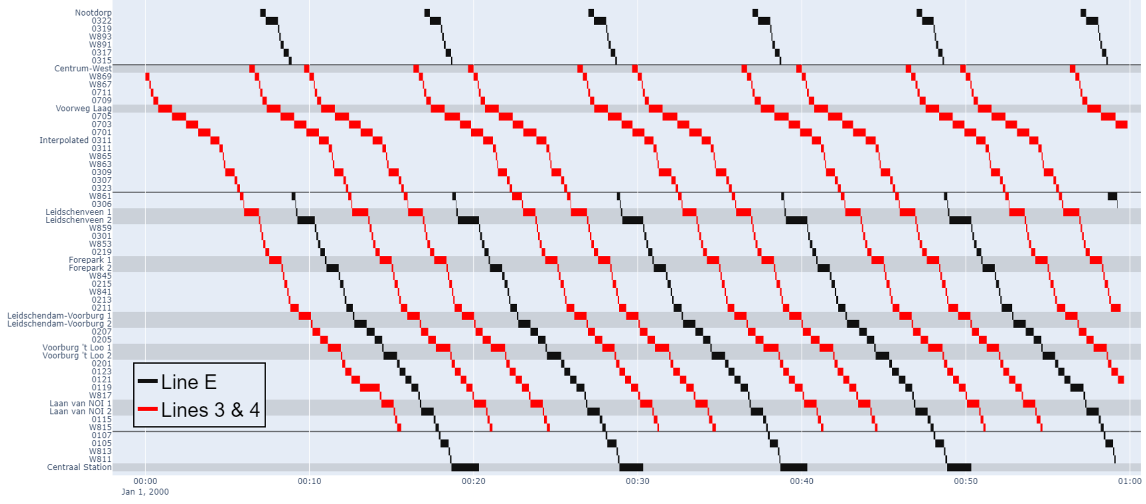

Figure 22: A simulation of the Current timetable for a spread of 200s

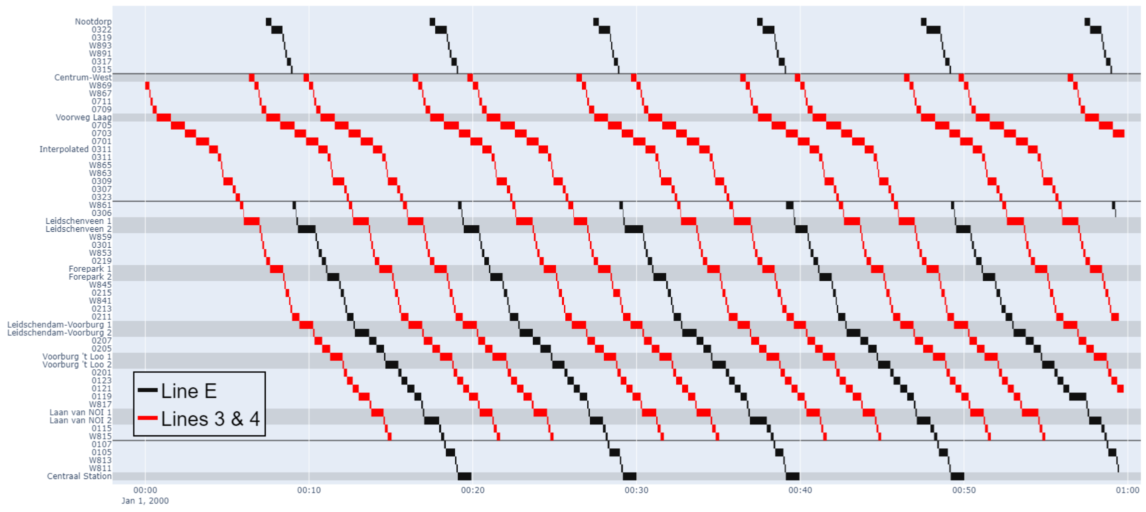

Figure 23: A simulation of the Average Data timetable for a spread of 200s

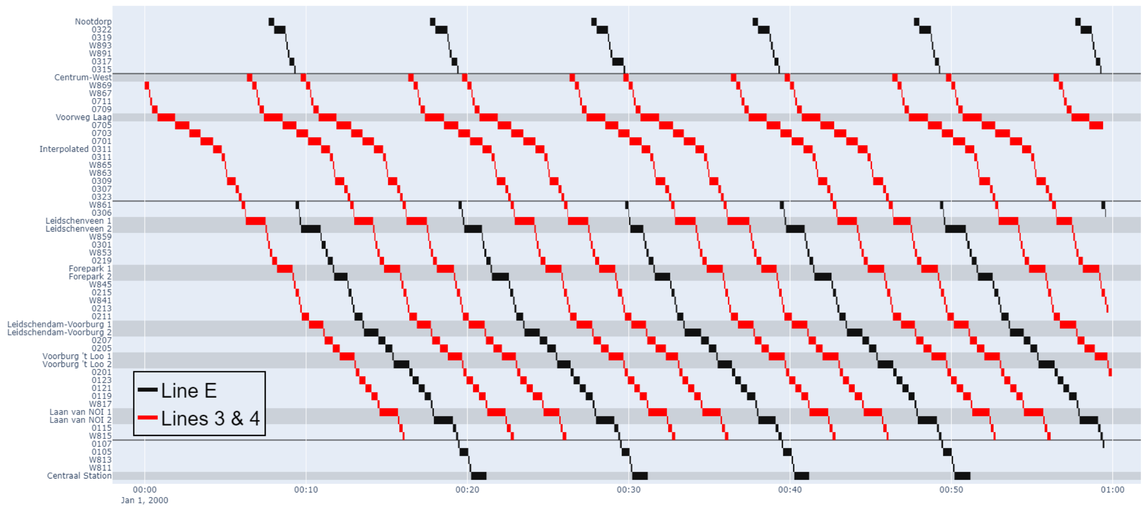

Figure 24: A simulation of the Buffer timetable for a spread of 200s

A.2 200 second spread

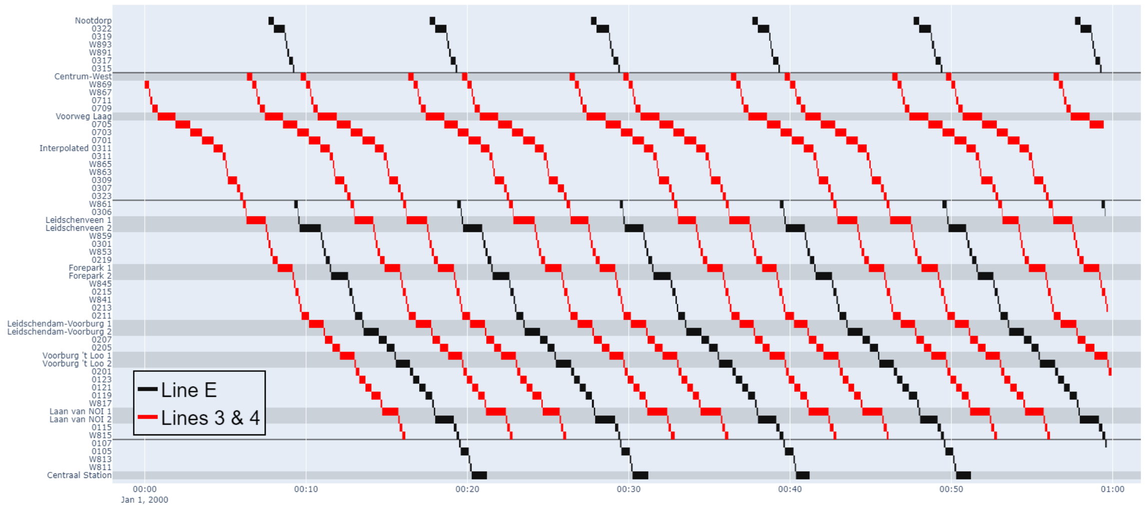

Figure 25: A simulation of the Average Track timetable for a spread of 200s

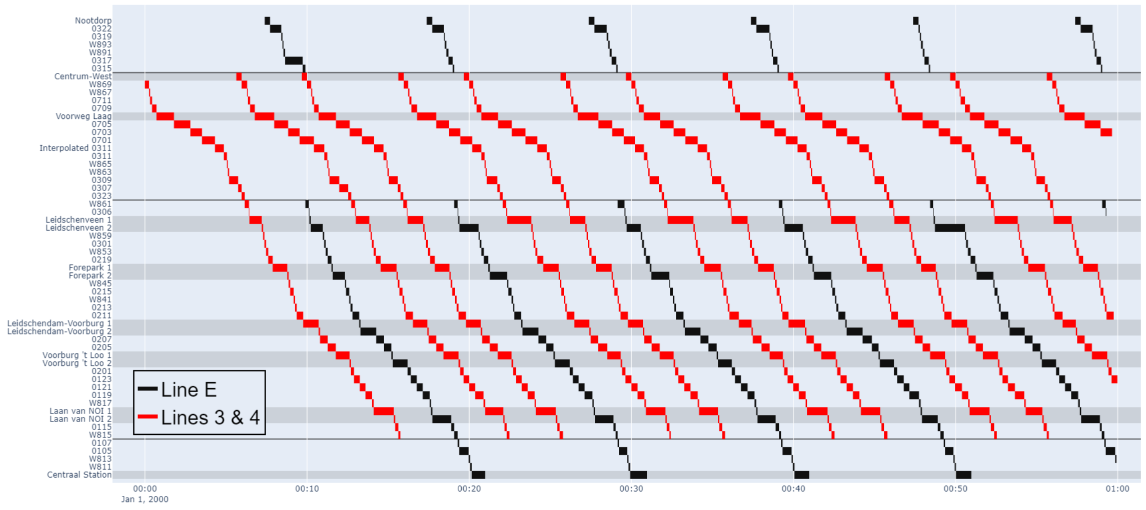

Figure 26: A simulation of the Average Train timetable for a spread of 200s

A.2 200 second spread

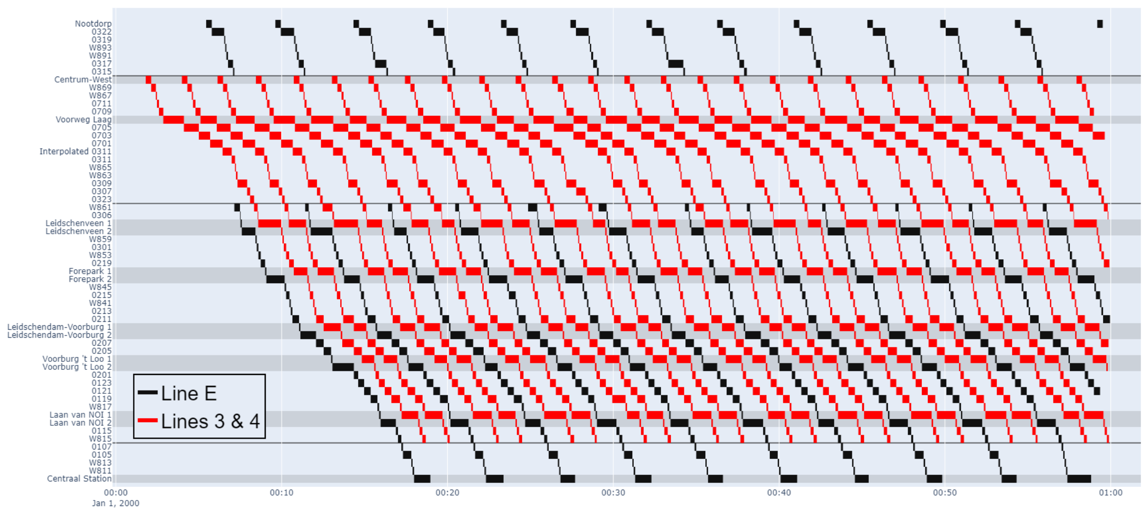

Figure 27: A simulation of the Average Timetable timetable for a spread of 200s

References

[1] Halfjaarcijfers ns 2019. `https://nieuws.ns.nl/halfjaarcijfers-ns-reizigersgroei-zet-door-spoorplafond-in-zicht/`. Accessed: 2019-10-14.

[2] Htm timetable. `https://www.htm.nl/dienstregeling/tram-3`. Accessed: 2020-06-16.

[3] Htm wiki departure punctuality kpi's. `http://htmwiki.nl/#!hackathon/bigdata.md`. Accessed: 2020-02-21.

[4] Onderzoek prestaties randstadrail. `https://nieuws.ns.nl/halfjaarcijfers-ns-reizigersgroei-zet-door-spoorplafond-in-zicht/`. Accessed: 2019-10-14.

[5] Ret overcapacity warning. `https://www.ovpro.nl/metro/2015/07/28/populariteit-randstadrail-maakt-uitbreiding-noodzakelijk/?gdpr=accept`. Accessed: 2020-06-16.

[6] Ret timetable. `https://www.ret.nl/home/reizen/dienstregeling/metro-e.html`. Accessed: 2020-06-16.

[7] Michael R Bussieck, Thomas Winter, and Uwe T Zimmermann. Discrete optimization in public rail transport. *Mathematical programming*, 79(1-3):415–444, 1997.

[8] Gabrio Caimi, Leo Kroon, and Christian Liebchen. Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, 6(4):285–312, 2017.

[9] Malachy Carey. Optimizing scheduled times, allowing for behavioural response. *Transportation Research Part B: Methodological*, 32(5):329–342, 1998.

[10] Matteo Fischetti and Michele Monaci. Light robustness. In *Robust and online large-scale optimization*, pages 61–84. Springer, 2009.

[11] Matteo Fischetti, Domenico Salvagnin, and Arrigo Zanette. Fast approaches to improve the robustness of a railway timetable. *Transportation Science*, 43(3):321–335, 2009.

[12] Ove Frank. Two-way traffic on a single line of railway. *Operations Research*, 14(5):801–811, 1966.

[13] Thom Glastra. Enabling glosa for on-street operating traffic light controllers. *http://resolver.tudelft.nl/uuid:63a21739-a200-4375-8f19-641352e504b2*, 2020.

[14] A González-Gil, R Palacin, and P Batty. Optimal energy management of urban rail systems: Key performance indicators. *Energy Conversion and Management*, 90:282–291, 2015.

[15] HTM Personenvervoer N.V. Htm 2019 annual report. `https://assets.production.htm.redkiwi.io/Jaarverslagen/Jaarverslag_2019.pdf`, April 2020.

[16] Leo Kroon, Gábor Maróti, Mathijn Retel Helmrich, Michiel Vromans, and Rommert Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, 42(6):553–570, 2008.

[17] Leo G Kroon, Rommert Dekker, and Michiel JCM Vromans. Cyclic railway timetabling: a stochastic optimization approach. In *Algorithmic methods for railway optimization*, pages 41–66. Springer, 2007.

[18] Christian Liebchen, Marco Lübbecke, Rolf Möhring, and Sebastian Stiller. The concept of recoverable robustness, linear programming recovery, and railway applications. In *Robust and online large-scale optimization*, pages 1–27. Springer, 2009.

[19] Christian Liebchen and Sebastian Stiller. Delay resistant timetabling. *Public transport*, 1(1):55–72, 2009.

[20] Ludolf E Meester and Sander Muns. Stochastic delay propagation in railway networks and phase-type distributions. *Transportation Research Part B: Methodological*, 41(2):218–230, 2007.

[21] Ministerie van Infrastructuur en Milieu. Hoofdrapport nmca, 2017.

[22] Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.

[23] Yanbo Zhao and Petros Ioannou. Positive train control with dynamic headway based on an active communication system. *IEEE Transactions on intelligent transportation systems*, 16(6):3095–3103, 2015.