UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Minimizing expected passenger travel time by optimal buffer allocation in train networks

Henri Pieter Arwyn Goos

M.Sc. Thesis
August 2017

Supervisors:
dr. J. Goseling (UT)
J. van t Wout (NS)
Prof. dr. Dennis Huisman (NS)

Stochastic Operations Research
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
The Netherlands

Abstract

This project considers railway timetable development where the expected passenger travel time is minimized. Buffers are placed in the network to absorb delays and an optimization model is formulated to choose these buffers optimally to minimize expected passenger travel time. A new way of modeling so called ”excess journey time” has been proposed. Given this optimization model, a piecewise-linear approximation of the problem is formulated, where error bounds in terms of the objective function are given. In the model and literature regarding this subject a simplification of reality is made, by dividing the train network into independent parts, to get analytical expressions for the goal function. A model where this simplification is not made is developed and it is found that the simplification causes a severe decrease in solution quality. However, in this research a heuristic is proposed which results in solutions that are very close to optimality. The model is applied to the Dutch intercity network where it is concluded that the model is able to generate timetables meeting the demands necessary for the Dutch network.

Keywords Railway timetabling, expected passenger travel time, piecewise-linearization, mixed integer programming.

Preface

This report is the result of the graduation project I have been working on for the past couple of months. I really enjoyed the combination of both theoretical mathematical analysis and practical applicability of such an import part every everybody's life, the railway network. I would like to thank a few people in particular for their help and support during the project.

First of all, I would like to thank Dr. Jasper Goseling for his supervision of the project. I really enjoyed our collaboration and especially that he made me look at the broader mathematical picture at hand, of the sometimes quite practical problems of railway timetabling. Furthermore, he gave the right amount of guidance when needed, while at the same time giving me the opportunity to explore my own ideas.

Next I would like to thank Joël van t Wout for my supervision at NS. He especially gave me insight into the complications, and beauty of railway timetabling problems. Furthermore, he really helped me and gave good advice on the programming implementations of the models developed.

I would like to thank Prof. Dr. Dennis Huisman for allowing me to do my graduation thesis at NS and the guidance he provided during the formulation of the research goals. I would also like to thank the rest of the department PI for their discussions and insights.

Thanks Tjeerd, Leon, Valerie, Manon and Rory for the great time we had as the board of the internship association at NS. The events organized really gave insight into the diverse activities that NS is engaged in. The trip to the Operational Control Center Rail (OCCR) showed me the relevance of this project as the importance of robust, though speedy timetables was strongly emphasized by OCCR employees. Furthermore, the trip to Frankfurt we organized was an absolute highlight.

Last, thanks to all my friends and family for their support during the time I was working on this project. I hope you enjoy reading this thesis.

Arwyn Goos,
Juli 2017, Utrecht

Contents

1 Introduction

1.1 Company background

Netherlands Railways (NS) is the largest railway transport provider of The Netherlands. Founded in 1937, as a result of a merge of several different railway providers, NS has operated the entire Dutch network until the 90's. From then on, a few small providers have made their appearance on mostly small lines. The Dutch railway network is used every day by on average 1.1 million passengers. Next to passenger transport, there is freight train transport, which uses the same infrastructure as passenger trains. I have been doing research at the department PI (Process quality and Innovation). PI is mainly a research department which focuses on developing decision support tools which are being used in other departments of NS. Operations Research is used to give a better understanding and improve decision making in various areas such as Crew Scheduling, Rolling Stock Scheduling and Train Timetabling. In 2008, the department won the Franz Edelman Award for Achievement in Operations Research and the Management Sciences awarded by the Institute for Operations Research and the Management Sciences with their paper, [1] on the use of Operations Research in developing timetables, rolling stock schedules and crew schedules for the Dutch network.

1.2 Planning process and requirements

This section will provide insight in the necessary decisions to be made before a railway network can be taken into operation. In general, there are 5 consecutive phases of decision making when developing a railway system:

1. *Passenger Estimation.* In order to generate an operating system, it is of great importance to know, or have an estimation of, the amount of passengers traveling from a specific origin to destination. Depending on those numbers decisions can be made about the quality, frequency and size of trains in specific sections. For every Origin-Destination pair, the amount of passengers traveling from that specific Origin to the Destination is estimated, resulting in the so called OD-Matrix.

2. *Line Planning.* After the OD-matrix is determined, a line plan should be determined. The line plan is the collection of all train lines. Each line has an origin and destination; a frequency and a certain stopping pattern, indicating the stations where the train halts. The line plan problem determines how to cover the railway network with lines best, when considering traffic demand, operating costs and many more.

3. *Timetabling.* After the line plan is determined, a train timetable can be constructed. For every train it is known at which station it halts. The timetabling part assigns specific moments for the arrival time and departure time of each train at every station. Here various factors are taken into consideration. The minimum ride time between stations should be taken into account. The time between specific trains should not be too big, making passengers wait long at a transfer station. Also there are a lot of safety constraints. There are specific requirements on for example the time between trains using the same piece of infrastructure. Usually a minimum time between such trains is demanded to avoid collisions. So a lot of factors play a role in timetabling and these factors possibly influence each other. Furthermore, the timetable is commonly cyclic, meaning that the timetable repeats itself after a certain amount of time. This is done with the aim of customer clarity.

4. *Rolling Stock Planning.* After the timetable is constructed, the next problem is assigning train units to train lines in the timetable. NS owns a lot of different type of rolling stock, for example single deck and double deck trains. For every train line it should be determined which rolling stock is operated. Note that rolling stock does not need to be assigned to a train line for the entire day but can be used on different train lines throughout. Also during peak hours, usually more rolling stock is used compared to off-peak hours. The rolling stock plan describes for every piece of rolling stock when and where it is used throughout the day.

5. *Crew Planning.* In order to operate the system, drivers and conductors are necessary on trains. There are some challenging aspects that need to be taken into account here. First of all, crew should start and end the day at their home station, as it is undesired that a employee finishes it's day on the other side of the country. Furthermore, there are some specific demands on the hours of employees. There are constraints on the amount of time an employee is allowed to work uninterrupted. Also a big issue is variation in the work of drivers and conductors. It is undesired by drivers and conductors to work on the same line during the entire day. They want to be employed on a diverse set of train lines. All these factors and many more need to be taken into account when making a crew planning, specifying for every employee on which rolling stock it should be at a specific moment during the day.

The phases described above are executed consecutively. Note however that optimal decision making in a certain phase is dependent on the decision making in a later phase. For example during passenger estimation, passenger flows are distributed over the network. But in practice how passengers move is dependent on the timetable, which is determined in a later phase. So the distribution of passenger flows is dependent on the timetable, which is in turn dependent on the passenger flow. In theory all the phases should be executed at the same time. Though there is a lot of research going on on how to integrate these different phases, for example [2], [3], [4], at the moment this is too hard a problem. This report will focus on the timetabling part, assuming the line plan development and passenger estimations have already been performed. The rolling stock and crew planning problem will not be considered.

1.3 Problem motivation

There is a great need to provide fast and quality service for passengers. The railway network is owned by the Dutch government, so the government decides which companies can provide service on which pieces of infrastructure. The NS has the right to continue operating at least until 2025 on the current net. However, the Dutch government has demanded that the NS meets a set of performance measures, otherwise they will consider giving other railway companies the opportunity to operate on the net currently used by NS. These performance measures are called KPI's (Key Performance Indicator), which are among others quality of transfers, transport capacity during rush hour and punctuality.

To meet these demands it is of vital importance that the railway timetable is of best quality possible from the viewpoint of passengers. After all, the motto of NS is "The passenger on 1,2, and 3". The timetable should be such that it performs good when operations are going as planned, but also that it is robust against unexpected events. A tool used by NS to generate timetables is the module CADANS. CADANS is a solver which finds a solution to the Periodic Event Scheduling Problem (PESP). PESP will be discussed later on in this report. CADANS finds a feasible schedule, meaning the schedule meets all the safety and minimum ride time constraints. However, as it is just a feasibility problem, it gives no judgment about the quality of the timetable. Ideally a goal function should be added to get the best quality schedule possible, for a certain definition of quality. For this research, the quality of a schedule will be determined by the expected passenger travel time. Here expected refers to the possibility of delays occurring in the network, implicating the travel time of a passenger is a stochastic variable. By taking the expected travel time as goal function we will take into account both efficiency and robustness, as a trade off is being made by the two in the goal function.

1.4 Research goal

The goal of this research is to develop a way to construct feasible timetables which A) minimize expected travel time and B) can be applied to large networks. There has been previous research, [5], which developed expressions for expected travel time and used this as a goal function to develop train schedules. Here expected travel time was minimized by placing time buffers in the network, allowing to absorb delays occurring during operation. In this research a similar approach will be used where several extensions will be made to meet the specific demands of the Dutch network. Very generally speaking we have the problem of minimizing the expectation of a certain function, where the buffers D are the decision variables. The goal function is dependent on the decision variable D and a set of stochastic variables X, which represents the delays. There are also constraints on the decision variables D, resulting in a feasible set P. So the problem will be of the following form:

$$\min_{D \in P} \mathbb{E}[f(D, X)].$$

One of the main considerations when constructing a train schedule is to spread alternative train routes. Different train routes between two stations should be spread out as much as possible to minimize the waiting time until the first mode of transport for passengers arriving randomly at the station. This also gives passengers flexibility to choose their moment of departure. Previous research suggested it is computationally very hard to take this spreading into account. For the Dutch network however, it is of utmost importance for trains to be spread as good as possible. The goal is to examine if there are ways to take into account spreading which is still computationally feasible.

Based on previous research it will most likely be necessary to approximate the goal function by a piecewise linear approximation because of computation time considerations. It is desired that this approximation approximates the original problem as good as possible. We will determine if error bounds between the approximated and original problem, in terms of the goal function values of the solutions found, can be formulated.

Last, in the literature regarding minimizing expected passenger travel time a simplification of the problem is made by dividing the network into independent parts. This will be explained later on in the report. In doing so, analytical expressions for the goal function can be found, which is not possible if this simplification is not made. However, the effect of the simplification on solution quality has not yet been examined in literature. We will develop a model which does not divide the network into independent parts and compare it with the model developed in literature to see if the simplification made is justified.

Summarizing, the main research goals are the following:

1. Develop expressions for the expected passenger travel time depending on the delay characteristics of the Dutch network.
2. Construct a new way of modeling the spreading of trains which is numerically applicable to large train networks.
3. Determine error bounds between the original and piecewise approximated problem in terms of goal function for the solutions found.

4. Develop a model which does not divide the network into independent parts to conclude if the simplification made in literature is justified.

1.5 Outline

The structure of the report is as follows. Chapter 2 gives a literature review of several subjects used in this research. In Chapter 3, the optimization problem will be formulated, where the goal function and constraints will be defined. In Chapter 4 piecewise linearization is discussed. Algorithms are described and performance bounds are given. In Chapter 5 a model will be developed in order to check simplifications made in the model of Chapter 3. Chapter 6 describes the numerical results of the models. Chapter 7 closes with a conclusion and recommendations for future research.

2 Literature review

This chapter gives an overview of research already performed on subjects relevant for this project. First, a mathematical formulation of finding timetables will be examined. Then, the distribution of delays based on real life data will be considered. Afterwards research regarding buffer allocation will be discussed. Last, mathematical techniques to solve stochastic programs are examined.

2.1 Train scheduling

Graph representation

We will now focus on how to convert the problem of finding a timetable into a mathematical problem. In order to do so, we will first show how a train network can be represented as a graph, based on for example [6], [5] and in particular for the Dutch network, [7]. Using this graph, a mathematical formulation of the problem will be given.

A train network can be represented by a directed graph $G(V, E)$ where V is the set of vertices and E is the set of edges. The vertices represent stations and the edges represent actions. At a station i, a train j has to arrive and depart. Both are modeled as a vertex $V_{Arr}^{i,j}$ and $V_{Dep}^{i,j}$. The total vertex set is the union of all the trains and all the stations it traverses, so $V = \bigcup_i \bigcup_j \left[V_{Arr}^{i,j} \cup V_{Dep}^{i,j}\right]$. Furthermore the edges represent interactions between nodes. There are different kind of edges connecting nodes:

1. *Ride edge.* A train j can make a ride action, where it is riding from a station k to station l. This is modeled as an edge between $V_{Dep}^{k,j}$ and $V_{Arr}^{l,j}$.
2. *Dwell edge.* For every train j, halting at station i, there is a dwell edge, where the train is waiting to allow passengers to enter the train, between vertices $V_{Arr}^{i,j}$ and $V_{Dep}^{i,j}$.
3. *Transfer edge.* This is an action from the viewpoint of the passenger. If a passenger needs to make a transfer at a station k, transferring from train i to train j, we construct an edge between the corresponding nodes. An edge will be constructed between $V_{arr}^{k,i}$ and $V_{Dep}^{k,j}$.
4. *Headway edge.* A headway edge models the relation between trains who should have a certain minimum headway from eachother for safety considerations. This means that two trains at a certain station should be scheduled a minimum time apart. Commonly, such constraints occur when the trains use the exact same piece of infrastructure. Suppose we have a headway constraint between two trains i and j at a station k. These constraints can either be on the arrival part or the departing part. This interaction is modeled as an edge between $V_{Arr}^{k,i}$ and $V_{arr}^{k,j}$ or $V_{Dep}^{k,i}$ and $V_{Dep}^{k,j}$.

These are all the interactions that happen between nodes. So to represent a network, the graph $G(V, E)$ defined as above provides us with all the information that is required to translate the problem into a mathematical formulation, which will be done in the following paragraph. An example of a graph representation of a train network is given in the following picture:

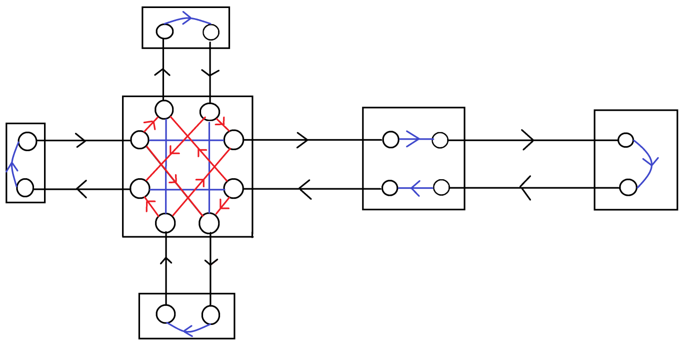


Figure 1: Graph representation of a train network

Here a black edge is a ride activity of a train, a blue edge is a dwell activity of a train and the red edges are passenger transfer actions. The black boxes are train stations. A headway edge looks like the following:

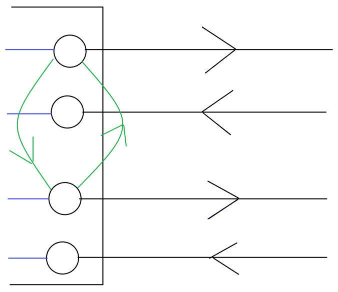


Figure 2: Example of headway edges

Here the two trains use the same piece of infrastructure after leaving the station, so a headway constraint is necessary. The green edges are the headway edges.

Periodic Event Scheduling Problem

The Periodic Event Scheduling Problem (PESP), introduced in [8], deals with finding a feasible solution to an Periodic Event Scheduling Problem, or to conclude that such a solution does not exist. A periodic timetable means that the timetable repeats itself after the timetable period T. So an event taking place at time b, will also take place at time $b + nT$, with $n \in \mathbb{Z}$. A periodic timetable is desired to give passengers more ease to remember the train schedule and get used to it. Usually, a timetable of period $T = 60$ minutes is desired. From a more general perspective, PESP is a framework to assign time values to periodic events, when there are constraints on the time between specific events. Applied to train scheduling, PESP needs, among others, the graph representation of a train network, described in the previous paragraph. PESP assumes that for every edge $e = (i, j) \in E$, connecting the vertices i and j, there is a minimum time l_e and maximum time u_e allowed for this specific action. PESP answers the following question:

Question: Given a directed graph $G = (V, E)$, the vectors l,u and an integer T. Does there exist a vector b such that the following holds:

$$(b_j - b_i - l_e) \bmod T \leq u_e - l_e, \ \forall e = (i,j) \in E.$$

Note that every b_k is the time of a specific node in the graph formulation. So for every $v \in V = \bigcup_i \bigcup_j \left[V^{i,j}_{Arr} \cup V^{i,j}_{Dep} \right]$ a time value b_k is assigned, for some k. In essence, PESP is a feasibility problem. Another way of representing PESP is the following by [6]:

$$\begin{array}{lll} \text{Find} & (b,n) & \\ \text{s.t.} & l_e \leq b_j - b_i + n_e T \leq u_e & \forall e = (i,j) \in E \\ & 0 \leq b_i < T & \forall i \in \mathbb{N} \\ & n_e \in \mathbb{Z}, & \forall e = (i,j) \in E \end{array}$$

Notice that PESP is a Mixed Integer Linear Program (MILP). A possible extension to the model is to add the constraint that the time an event takes place is a natural number, i.e. $b_i \in \mathbb{N} \ \forall i$, as timetables communicated to passengers usually have a precision in minutes. This would make the problem an Integer Linear Program (ILP). By [9] it can be shown that if l, u are integer, and there exists an solution to the problem, then also an integer solution exists to the problem. Furthermore, [9] showed that for $T \geq 3$, PESP is a NP-hard problem, by making a reduction from the k-Vertex Colorability Problem.

In [10] a description of how PESP is solved at NS is given. The solver is called CADANS, which was developed in collaboration with ORTEC, and has several modules. First a solution to the PESP problem is found. In order to do so, a lot of preprocessing is used to remove redundant constraints. After a feasible schedule is found, the next module does post optimization. Event times are moved around such that for example trains driving with a frequency bigger than two are spread as evenly as possible over the cycle period. If no solution is found, CADANS outputs the set of constraints causing the infeasibility and gives suggestions on how to adjust the constraints such that a feasible schedule is possible.

Another recent promising approach is to reformulate PESP as a SAT problem and solve the problem using SAT solvers, described in [11].

2.2 Delay distributions

PESP does not have a goal function. This means any solution is of equal quality in the eyes of PESP. However, in practice this might not be the case. In this research a goal function will be added to the PESP problem. We will develop expressions for the expected travel time and use this as an objective in the model. The travel time of a passenger is a stochastic variable, because there are stochastic delays occurring during operation. To develop expressions for expected travel time, we first need to determine how delays are distributed. There have been several researches on the distribution of delays, which will be analyzed and conclusions about the delay distributions to be used are made.

The research of [12] considers data in the Eindhoven area regarding train delays. An entire week, consisting of 1846 trains were considered. It was concluded that departure delays at stations are best modeled by a negative exponential distribution.

In [13], delay data was analyzed which was gathered during 9 months at 24 UK train stations, resulting in nearly one million recorded delay times. To fit the data, an exponential and q-exponential distribution were considered. It was concluded that the q-exponential distribution fits the data best, however the exponential distribution is also a good fit.

The Dutch The Hague station was examined by [14]. It was concluded that the Weibull distribution fits the data best, where the shape parameter of the Weibull distribution tends to be smaller than 1 for arrival and departure actions and bigger than 1 for dwell actions.

Lastly, [15] considered the Indian railway network. Over a period of 30 days, delays were recorded at 26 stations resulting on around 10000 delay recordings. It was concluded that a Weibull distribution fits the data best for arrival delays and a log-normal distribution is best to model departure delays.

The general trend in the researches seems to be that a Weibull distribution or exponential distribution is a good choice for modeling delay distributions. Since the exponential distribution is a specific case of the Weibull distribution, it is convenient to consider Weibull distributions. NS also has internal data regarding train delays. The general conclusion was that either an exponential or a gamma distribution were best fits, though it is not clear whether Weibull distributions were also considered. The advantage of using a Weibull distribution comes from the fact that a Weibull distribution is more flexible in terms of the shapes it can take than the exponential distribution. The exponential distribution is a strictly decreasing function, whereas the Weibull distribution is not monotonically increasing or monotonically decreasing. Consider the following figure:

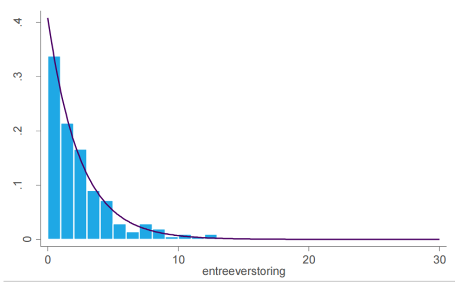


Figure 3: Dataset 1

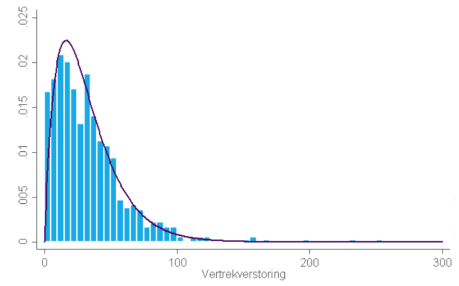


Figure 4: Dataset 2

These are datasets recorded by the NS at a specific station. For the first dataset an exponential distribution would be a good fit. For the second dataset however an exponential distribution would not be a good fit, as it can not model the increasing trend shown at the beginning of the dataset. A Weibull distribution would be suited to model the second dataset properly. In this research we will consider both exponential and Weibull distributed delays.

Parameter estimation

After deciding the exponential distribution and the Weibull distribution are the distributions to be used in this research, we need to determine a way to estimate distribution parameters, given a set of data. The estimator attempts to approximate the parameters of certain measurements. Various criteria and corresponding algorithms have been developed which all have a different criterium for being a "good fit". The most famous techniques are Maximum Likelihood Estimation, Method of Moments and Minimum Squared Error Estimation, as described in for example [16]. Maximum Likelihood Estimation is generally the most used technique. This is the method we will explain here.

The principle of Maximum Likelihood Estimation is that the parameters should be chosen in such a way, that the likelihood of the observations that were made is maximal. Suppose there are n independent and identically distributed observations $x_1, ..., x_n$, coming from a distribution function $f(x, \theta)$. Here θ is a vector of parameters. As the samples were independent, the joint density is just the product of the individual density functions. Let $\mathcal{L}(\theta; x_1, ..., x_n)$ be the likelihood function. It follows that the likelihood of doing the observations that we made is the following:

$$\mathcal{L}(\theta; x_1, ..., x_n) = f(x_1, ..., x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta).$$

This function should be maximized with respect to the parameter vector θ to find the maximum likelihood estimator for the observations. Now we will see what this means in practice, when using an exponential and Weibull distribution.

Exponential distribution

Suppose there is a set of delay observations $x_1, ..., x_n$ on which an exponential distribution should be fitted. The exponential distribution has $f(x) = \lambda e^{-\lambda x}$ as distribution function. The likelihood function for the exponential distribution is the following:

$$\mathcal{L}(\lambda; x_1, ..., x_n) = \prod_{i=1}^{n} \lambda e^{-\lambda x_i} = \lambda^n e^{-\lambda \sum_{i=1}^{n} x_i}.$$

This expressions should be maximized over λ i.e. we should solve $\frac{d}{d\lambda}\mathcal{L}(\lambda; x_1, ..., x_n) = 0$. Because $\ln(x)$ is a monotonically increasing function, this is equivalent with solving $\frac{d}{d\lambda}\ln(\mathcal{L}(\lambda; x_1, ..., x_n)) = 0$. We get:

$$\begin{aligned}\frac{d}{d\lambda}\ln(\mathcal{L}(\lambda; x_1, ..., x_n)) &= \frac{d}{d\lambda}\left[\ln(\lambda^n) + \ln\left(e^{-\lambda \sum_{i=1}^{n} x_i}\right)\right] \\ &= \frac{d}{d\lambda}\left[n\ln(\lambda) - \lambda \sum_{i=1}^{n} x_i\right] \\ &= \frac{n}{\lambda} - \sum_{i=1}^{n} x_i = 0.\end{aligned}$$

This results in $\lambda = \frac{n}{\sum_{i=1}^{n} x_i}$.

Weibull distribution

The Weibull distribution function is $f(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k}$. The likelihood function for the Weibull distribution given the observations $x_1, ..., x_n$ is the following:

$$\mathcal{L}(k, \lambda; x_1, ..., x_n) = \prod_{i=1}^{n} \frac{k}{\lambda}\left(\frac{x_i}{\lambda}\right)^{k-1} e^{-\left(\frac{x_i}{\lambda}\right)^k}.$$

To find the best k and λ we should find the maximum of the likelihood function. It is again convenient to take the logarithm of the likelihood function. This results in:

$$\begin{aligned}\ln(\mathcal{L}(k, \lambda; x_1, ..., x_n)) &= \ln\left(\left(\frac{k}{\lambda^k}\right)^n\right) + \ln\left(\prod_{i=1}^{n} x_i^{k-1}\right) + \ln\left(\prod_{i=1}^{n} e^{-\left(\frac{x_i}{\lambda}\right)^k}\right) \\ &= n\ln(k) - nk\ln(\lambda) + (k-1)\sum_{i=1}^{n} \ln(x_i) - \sum_{i=1}^{n}\left(\frac{x_i}{\lambda}\right)^k.\end{aligned}$$

Now the following system of equation should be solved:

$$\begin{aligned}
\frac{d}{d\lambda}\ln(\mathcal{L}(k,\lambda;x_1,...,x_n)) &= -nk\frac{1}{\lambda}+k\sum_{i=1}^{n}x_i^k\frac{1}{\lambda^{k+1}} && =0\\
\frac{d}{dk}\ln(\mathcal{L}(k,\lambda;x_1,...,x_n)) &= \frac{n}{k}-n\ln(\lambda)+\sum_{i=1}^{n}x_i-\sum_{i=1}^{n}\ln\left(\frac{x_i}{\lambda}\right)e^{k\ln\left(\frac{x_i}{\lambda}\right)} && =0
\end{aligned}$$

When solving this system of equations, we get by [17] the following expressions for λ and k.

$$\begin{aligned}
k &= \left[\frac{\sum_{i=1}^{N}x_i^k\ln(x_i)}{\sum_{i=1}^{N}x_i^k}-\frac{1}{N}\sum_{i=1}^{N}\ln(x_i)\right]^{-1}\\
\lambda &= \left[\frac{1}{N}\sum_{i=1}^{N}x_i^k\right]^{\frac{1}{k}}
\end{aligned}$$

Note that the expression for k is not explicit, but can easily be determined with numerical methods such as the Newton-Raphson method [18].

2.3 Buffer allocation

We have seen how train scheduling can be translated into a mathematical problem and how train delays are distributed. The main focus of this research is on buffer allocation. Buffers can be placed in the train network to absorb delays occurring during operation. We will now consider what is already known in literature about buffer allocation in train networks.

The research of [19] considers a train network for which an expression for expected travel time is developed. Depending on the delay distributions, expressions for different kind of actions that passengers can take are developed. The model is applied to a small subnetwork of the Belgian network. When comparing the solution of the timetable determined by the model with the timetable in use at that time in the Belgian railway network, a significant improvement was made.

The model of [19] was expanded in [5]. In [19] there were quite some simplifications made on the objective terms. [5] considers the same concept but with less simplifications. A model is developed to determine passenger flows, in order to properly weigh different kind of actions and thus making an analysis of the importance of specific actions/segments. The model is applied to all the Belgium hourly train network and has shown to be able to find feasible solutions, while also taking into account all kinds of safety constraints. A piecewise linearization using two linear segments of the goal function was made.

The model in [20] was developed by NS and the so-called SOM module used at NS resulted from this research. The model considers train lines and aims, by allocation of buffers, to minimize the delay that a train experiences. This is a significant difference with [5] and [19], as in those researches a minimization of passenger travel time is sought after. A further difference is that in [20], the train network is not divided into independent parts, which does happen in [5] and [19]. In this problem, a finite predetermined amount of buffer should be distributed over the different sections of the line. The results were put in practice at a, at the time, problematic line in terms of delays. A significant increase in performance was achieved afterwards.

2.4 Stochastic Programming

Later on in the report we will have a specific kind of minimization problem, for which the theoretical groundwork is explained here. The general form of the problem, using the notation of literature used, is $\min_{x \in X} \mathbb{E}[f(x, \xi)]$ with ξ being a stochastic variable. For the problem later on in the report specifically, an expression for the expectation of f is not possible, but f itself is. In this section, literature and techniques useful for solving this kind of problem will be elaborated on.

Stochastic Programming is a framework for modeling optimization problems where uncertainty is involved. Uncertainty can be apparent in different aspects, in the constraints, goal function and there are several criteria possible for when a solution is found. Stochastic Programs can be "Multi-Stage", meaning that the solution of the first stage program is dependent on the solution of the second stage problem. The two stage program is most commonly studied in stochastic programming literature, for example in [21] and has the following general form:

$$\begin{aligned} \min_{x \in X} \ & c^T x + \mathbb{E}[f(x, \xi)] \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned}$$

Here ξ is random, but follows a known probability distribution. $f(x, \xi)$ again is defined as follows:

$$\begin{aligned} f(x, \xi) = & \min_y q(\xi)^T y \\ & \text{s.t. } T(\xi)x + W(\xi)y \leq h(\xi) \\ & y \geq 0 \end{aligned}$$

In this formulation $x \in \mathbb{R}$ can be seen as a first-stage decision variable and $y \in \mathbb{R}$ as the second stage decision variable.

For our problem, we do not need to take a second decision based on the realizations of ξ (modeled by the second stage program), just on the expected travel time function. Also, as will be shown when the model for this project is actually developed, there are no constraints on the uncertain realizations of ξ. So for this problem, a first stage problem of the following form is what is desired:

$$\begin{aligned} \min \ & \mathbb{E}[f(x, \xi)] \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned}$$

In the following sections, we will consider two techniques to solve these kind of problems, when $\mathbb{E}[f(x, \xi)]$ can not be analytically determined, but where $f(x, \xi)$ is known.

2.4.1 Scenario construction

Suppose that the probability distribution of the stochastic vector ξ can be well approximated by a discrete distribution with known probability p_i for every scenario ξ^i. Then [22] and [23] write that the Stochastic Program can be transformed into a equivalent linear program of the following form:

$$\begin{aligned} \min & \sum_{i=1}^{N} p_i f(x, \xi^i) \\ \text{s.t. } & Ax = b \\ & x \geq 0 \end{aligned}$$

This program can be solved by standard methods, as it is just a deterministic program. However, the size of the linear program can become very big very fast. This makes the problem hard to solve numerically. Let the vector ξ be of length n. Furthermore, suppose that for every element ξ_j in ξ we allow m possibilities with corresponding probabilities $p_1, ..., p_m$. Then the total outcome space is of size m^n.

2.4.2 Sample Average Approximation

As we saw, we can expect quite some computational issues when we use scenario construction to solve our stochastic program. Sample Average Approximation (SAA) has a similar approach, but does not have the same outcome space problems.

Let x^* be the optimal solution to our stochastic program, and let z^* be it's corresponding function value, i.e. $z^* = \min_{x \in X} \mathbb{E}[f(x, \xi)] = \mathbb{E}[f(x^*, \xi)]$. SAA approximates the problem by sampling n samples $\xi^1, ..., \xi^n$, drawn from the distribution of ξ. Using these samples an approximation x_n^* of x^* and an approximation z_n^* of z^* is sought. These approximations are determined by considering the following problem:

$$\begin{aligned} \min \ & \frac{1}{n} \sum_{j=1}^{n} f(x, \xi^j) \\ \text{s.t.} \ & Ax = b \\ & x \geq 0 \end{aligned}$$

Here x_n^* is the optimal solution to the above problem and z_n^* is its corresponding objective value, i.e.:

$$z_n^* = \min_{x \in X} \left[\frac{1}{n} \sum_{j=1}^{n} f(x, \xi^j) \right] = \frac{1}{n} \sum_{j=1}^{n} f(x_n^*, \xi^j).$$

In essence, a Monte-Carlo approximation of the goal function is minimized when using SAA. In the following some properties of SAA regarding bias and convergence of z_n^* to z^* will be shown, by [24].

Bias

It will now be shown whether or not the estimator z_n^* is biased. z_n^* is unbiased if $\mathbb{E}\left(z_n^*\right) = z^*$. We get:

$$\mathbb{E}(z_n^*) = \mathbb{E} \left[\min_{x \in X} \frac{1}{n} \sum_{j=1}^{n} f(x, \xi^j) \right] \leq \min_{x \in X} \mathbb{E} \left[\frac{1}{n} \sum_{j=1}^{n} f(x, \xi^j) \right] = \min_{x \in X} \mathbb{E}[f(x, \xi)] = z.$$

Here the inequality follows because the minimum of an expectation in bigger than the expectation of the minimum. The estimator is a negatively biased estimator. However, this does not mean that z_n^* does not converge to z^*. In fact it does. We will show the estimator is a consistent estimator in the following section.

Consistency

We know $f_n(x_n^*) \leq f_n(x^*)$, as x_n^* is the optimal value when these specific n samples are considered. So for n samples, x^* is not necessarily the optimal solution, but x_n^* is. On the other hand we have $\mathbb{E}[f(x^*, \xi)] \leq \mathbb{E}[f(x_n^*, \xi)]$, as x^* is the optimal solution to the overall problem. Using these inequalities we get:

$$\begin{aligned}
|z_n^* - z^*| &= |f_n(x_n^*) - \mathbb{E}[f(x^*, \xi)]| \\
&= \max\{f_n(x_n^*) - \mathbb{E}[f(x^*, \xi)], \mathbb{E}[f(x^*, \xi)] - f_n(x_n^*)\} \\
&\leq \max\{f_n(x^*) - \mathbb{E}[f(x^*, \xi)], \mathbb{E}[f(x_n^*, \xi)] - f_n(x_n^*)\} \\
&\leq \max\{|f_n(x^*) - \mathbb{E}[f(x^*, \xi)]|, |\mathbb{E}[f(x_n^*, \xi)] - f_n(x_n^*)|\} \\
&\leq \sup_{x \in X} |f_n(x) - \mathbb{E}[f(x, \xi)]| \,.
\end{aligned}$$

Because $\lim_{n\to\infty} |f_n(x) - \mathbb{E}[f(x, \xi)]| = 0$ with probability one, we get that z_n^* converges to z^* with probability one.

3 Expected Passenger Travel Time

The following chapter will develop a framework for minimizing expected passenger travel time. The same approach as in [5] is used. However, there are several new contributions. First of all, the expressions developed in [5] and [19] were not very formal, meaning no precise expressions for the stochastic variables were stated. By reasoning and analyzing cases expressions were found. We will give a formal description of the problem. Furthermore, we allow to use Weibull distributed delays as opposed to just exponential delays and give expressions for the goal function. Lastly a new way of modeling excess journey time is developed.

As stated before, buffers are placed in the network to absorb delays. These buffers should not be too small, such that they cannot absorb any practical delay, nor should they be too big, causing passengers to wait unnecessarily long. A trade-off should be made between the two and this is done by considering the expected passenger travel time. Note that the model is not aimed to be robust against any delay. We consider relatively small high frequency delays and not low frequency big delays as a consequence of for example a train having a collision with a pedestrian. High frequency small delays are for example delays as a consequence of leaf on the rail, crowded trains during boarding etc. We allow buffers to be placed after dwell or transfer actions, making them absorb the delay which has occurred starting from the previous buffer in the system. Making the buffers absorb only the delays that have occurred since the last buffer is a simplification, as there might still be delay left in the network not entirely absorbed by the previous buffer. This will be treated in Chapter 5. The following picture shows the idea on a subnetwork at a station:

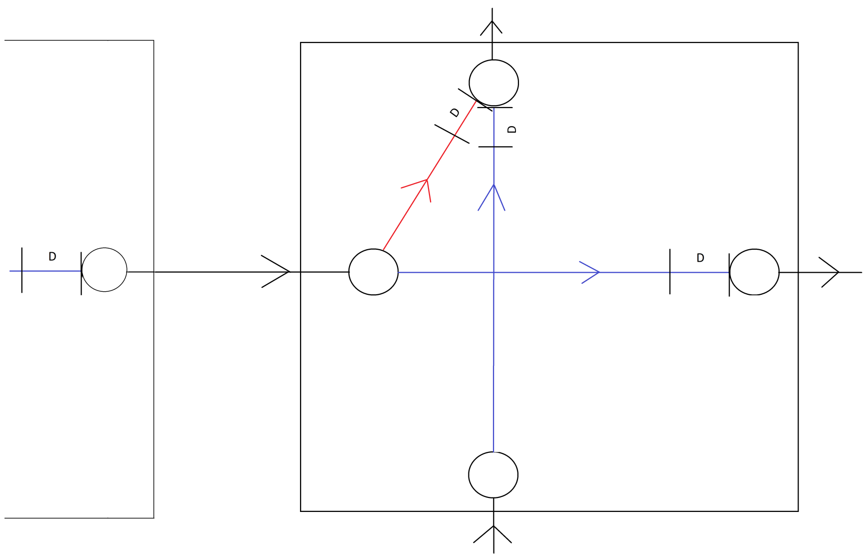


Figure 5: Example of buffer placement

We use the notation of [5], where D is the size of the buffer, not to be confused with delay. Delays occur on the black, blue and red edges according to some known distribution function. The big question is what the size of these buffers should be and that is exactly the model we will build to answer this question.

We are looking for an expression for the total expected passenger travel time. This will be done by first determining the expected travel time for a route of a passenger. To get the expected passenger travel time for a certain route, we will decompose the route a passenger takes into different actions. For these actions, expressions for the expected travel time will be developed, after which the total expression for the travel time of that route can be determined, by simply summing over the travel cost of the actions used. An action consists of all the activities between two buffers that a passenger comes by when traversing the network. The possible actions will now be explained.

First a passenger has to take a departing action to board the train. When the passenger has boarded the train and arrives at the next station, it can either stay in the train, which is a dwell action, or it can make transfer to an other train, which is a transfer action. Every destination can be reached by a finite amount of these actions. Because of the structure of the network and allowed buffer places, an action will always consist of a ride together with a dwell or transfer edge. Depending on the distribution of the delays at these edges and the buffers chosen, we want to know the expected passenger travel time. This will be done for every action possible.

The travel time in practice consists of the minimum travel time, which is the travel time when no delays were to occur, plus the travel time as a consequence of these delays. The first part however is a constant. There are certain minimum travel times for every action, independent from the timetable chosen. We will leave this out of the model, as there is no use in optimizing this. We will focus on minimizing the journey time on top of the technical minimum journey time.

3.1 Action expressions

For every possible action, the expected travel time will be determined in the following parts.

3.1.1 Depart action

When travelling from an origin to a destination, a passenger first has to depart at the station. Consider the following illustration:

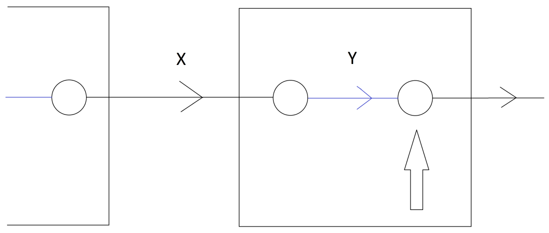


Figure 6: Depart action

Departing passengers arrive at the node the big black arrow is pointing to. After the blue edge, a buffer is placed that can absorb the delays that have occurred since the last buffer, a delay X during the ride activity and a delay Y during the dwell activity. If the cumulative delay is bigger than the buffer, the train will arrive late and the passenger has to wait. In this case, the passenger has to wait for a time $X + Y - D$, where D is the size of the buffer. In case the cumulative delay of the previous part is smaller then the buffer, a departing passenger does not have to wait, as the delay is entirely absorbed by the buffer and thus the train arrives on time. So the stochastic variable C^{Depart}, representing the travel time for a depart action, is the following:

$$C^{Depart}(D) = \max\{0, X + Y - D\}.$$

As we are interested in the expectation, let f^{Depart} be defined as $f^{Depart} = \mathbb{E}\left(C^{Depart}(D)\right)$. By definition, f^{Depart} is determined as follows:

$$\begin{aligned} f^{Depart} &= \mathbb{E}\left(C^{Depart}(D)\right) \\ &= \mathbb{E}\left(\max\{0, X+Y-D\}\right) \\ &= \int_0^\infty \int_0^\infty \max\{0, x+y-D\} f_x(x) f_y(y) dx dy. \end{aligned}$$

Here $f_x(x)$ and $f_y(y)$ are the distribution functions of X and Y. As we have seen that delays can be best modeled by exponential and Weibull distributed variables, we will apply this to the above expression.

Lemma 1. When using an exponential distribution, f^{Depart} is as follows:

$$f^{Depart}(D) = \frac{\lambda_x}{\lambda_y(\lambda_x - \lambda_y)}\left(e^{-\lambda_y D} - e^{-\lambda_x D}\right) + \frac{1}{\lambda_x} e^{-\lambda_x D} + \frac{1}{\lambda_y} e^{-\lambda_x D}.$$

Proof. See Appendix A.1.1 □

Lemma 2. When using a Weibull distribution, f^{Depart} is as follows:

$$\begin{aligned} f^{Depart}(D) = &\left(\lambda_y \Gamma\left(1 + \frac{1}{k_y}\right) - D\right) e^{-\left(\frac{D}{\lambda_x}\right)^{k_x}} + \lambda_x \Gamma\left(1 + \frac{1}{k_x}, \left(\frac{D}{\lambda_x}\right)^{k_x}\right) \\ &+ \int_0^D \left[(x-D) e^{-\left(\frac{D-x}{\lambda_y}\right)^{k_y}} + \lambda_y \Gamma\left(1 + \frac{1}{k_y}, \left(\frac{D-x}{\lambda_y}\right)^{k_y}\right)\right] f_x(x) dx. \end{aligned}$$

Proof. See Appendix A.1.2 □

Theorem 1. For an exponential distribution, f^{Depart} is a convex function in D

Proof. See Appendix B.1 □

3.1.2 Dwell action

Suppose that a passenger has already boarded the train and it does not leave or transfer at the next station. So the passenger is using a dwell action. What are the travel costs when a buffer D is placed for this passenger? In any case, the passenger needs to wait a time D. However, if the cumulative delays occurring during riding X and dwelling Y are bigger than this buffer, the passenger needs to wait for an additional time of $X + Y - D$. Concluding, the dwell travel time is given by the following stochastic variable:

$$C^{Dwell}(D) = D + \max\{0, X+Y-D\}.$$

We are interested in the expectation of this variable, so define f^{Dwell} as $f^{Dwell} = \mathbb{E}\left(C^{Dwell}(D)\right)$. By taking the expectation, we get the following expression:

$$\begin{aligned} f^{Dwell}(D) &= \mathbb{E}\left(C^{Dwell}(D)\right) \\ &= D + \mathbb{E}\left(\max\{0, X+Y-D\}\right) \\ &= D + \int_0^\infty \int_0^\infty \max\{0, x+y-D\} f_x(x) f_y(y) dx dy. \end{aligned}$$

Lemma 3. When using an exponential distribution, f^{Depart} is as follows:

$$f^{Dwell}(D) = D + \frac{\lambda_x}{\lambda_y(\lambda_x - \lambda_y)}\left(e^{-\lambda_y D} - e^{-\lambda_x D}\right) + \frac{1}{\lambda_x} e^{-\lambda_x D} + \frac{1}{\lambda_y} e^{-\lambda_x D}.$$

Proof. Since $f^{Dwell}(D) = D + f^{Depart}(D)$ and f^{Depart} was shown above, the result follows immediately. $\square$

Lemma 4. When using a Weibull distribution, f^{Depart} is as follows:

$$\begin{aligned} f^{Dwell}(D) = D &+ \left(\lambda_y \Gamma\left(1 + \frac{1}{k_y}\right) - D\right) e^{-\left(\frac{D}{\lambda_x}\right)^{k_x}} + \lambda_x \Gamma\left(1 + \frac{1}{k_x}, \left(\frac{D}{\lambda_x}\right)^{k_x}\right) \\ &+ \int_0^D \left[(x - D)e^{-\left(\frac{D-x}{\lambda_y}\right)^{k_y}} + \lambda_y \Gamma\left(1 + \frac{1}{k_y}, \left(\frac{D-x}{\lambda_y}\right)^{k_y}\right)\right] f_x(x)dx. \end{aligned}$$

Proof. Since $f^{Dwell}(D) = D + f^{Depart}(D)$ and f^{Depart} was shown above, the result follows immediately. $\square$

Theorem 2. For an exponential distribution, f^{Dwell} is a convex function in D

Proof. The sum of two convex functions is convex. Since $f^{Dwell}(D) = D + f^{Depart}(D)$ and by Theorem 1, f^{Depart} is a convex function and D is also a convex function, the result follows immediately. $\square$

Note that for every buffer placed after a dwell edge, we have an $f^{Depart}(D)$ function as well as an $f^{Dwell}(D)$ function depending on the same buffer D.

3.1.3 Transfer action

Now suppose that a passenger is making a transfer action. For a transfer action, we model expected transfer time as a consequence of the possibility of missing the train the passenger is transferring to. A passenger misses its transfer when he arrives later than the scheduled time of departure of the train it is transferring to. So it is assumed that the train that is being transferred to leaves on time. This is a simplification of reality. If we use this definition, a passenger misses its train if the sum of the delays acquired during the ride and transfer action is bigger than the buffer D. If a passenger misses its train, we assume that a passenger takes the exact same train, but one frequency later. This is a conservative assumption, because there might be other travel routes to the passenger's destination that make use of other lines, which arrive earlier than the next frequency of the train that is missed. If we let the passenger take the next frequency train, the waiting cost for missing a transfer is $\frac{T}{m}$, where T is the timetable period and m the frequency of the train transferring to. In either case, a passenger needs to wait for a time D. This gives us the following stochastic variable for the travel time:

$$C^{Transfer}(D) = D + \mathbb{1}\{X + Y > D\}\frac{T}{m}.$$

Here Y is the stochastic delay a passenger experiences when walking from a train to another train in a station. Furthermore, $\mathbb{1}\{X+Y > D\}$ is an indicator variable, taking value 1 when $X+Y > D$

and taking value 0 if $X + Y \leq D$:

$$\mathbb{1}\{X + Y > D\} = \begin{cases} 1 & \text{if } X + Y > D \\ 0 & \text{if } X + Y \leq D \end{cases}$$

We are interested in taking the expectation. This gives us the following expression for f^{Depart}:

$$\begin{aligned} f^{Trans}(D) &= \mathbb{E}\left(C^{Trans}(D)\right) \\ &= D + \mathbb{E}\left(\mathbb{1}\{X + Y > D\}\frac{T}{m}\right) \\ &= D + \int_0^\infty \int_0^\infty \mathbb{1}\{X + Y > D\}\frac{T}{m} f_x(x) f_y(y) dx dy. \end{aligned}$$

Lemma 5. For an exponential distribution, f^{Trans} is as follows:

$$f^{Trans} = D + \frac{T}{m}\frac{\lambda_y e^{-D\lambda_x} - \lambda_x e^{-D\lambda_y}}{\lambda_y - \lambda_x}.$$

Proof. See Appendix A.3.1 □

Lemma 6. For an Weibull distribution, f^{Trans} is as follows:

$$f^{Trans} = D + \frac{T}{m}\int_0^\infty f_x(x) e^{-\left(\frac{D-x}{\lambda_y}\right)^{k_y}} dx + e^{-\left(\frac{D}{\lambda_x}\right)^{k_x}}.$$

Proof. See Appendix A.3.1 □

The transfer objective function is not convex, as the following picture illustrates for $\lambda_x = 1.9$ and $\lambda_y = 0.6$.

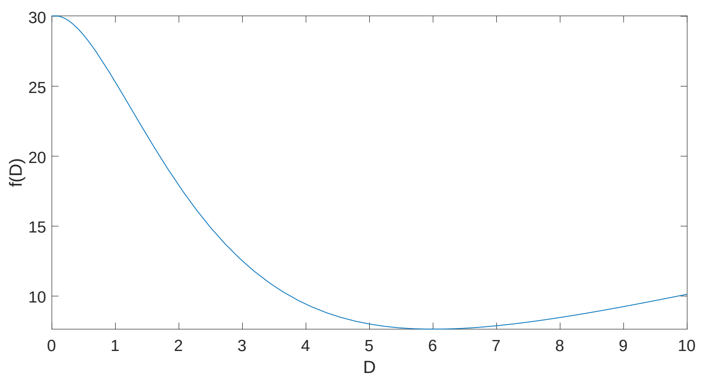
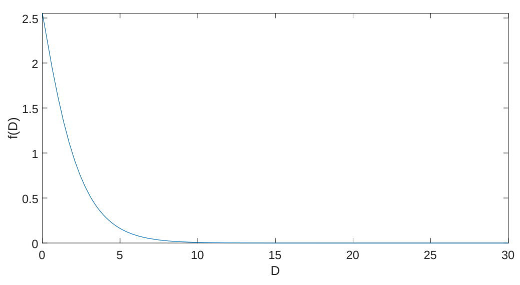
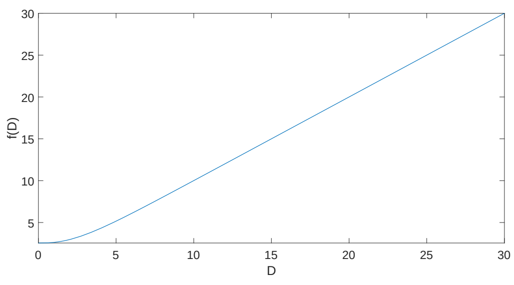


Figure 7: Transfer goal function with $\lambda_x = 1.9$ and $\lambda_y = 0.6$

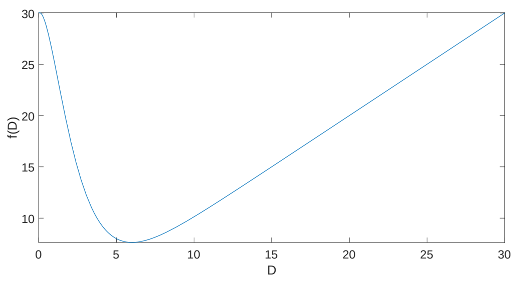
The goal functions of the three actions will be shown and analyzed on their behavior. For $\lambda_x = 0.64$ and $\lambda_y = 1$, the goal function for the actions looks as follows:



Depart action, $\lambda_x = 0.64$ and $\lambda_y = 1$



Dwell action, $\lambda_x = 0.64$ and $\lambda_y = 1$



Transfer action, $\lambda_x = 0.64$ and $\lambda_y = 1$

The departure action is a strictly decreasing function. The bigger the buffer, the lower expected travel time for a departing passenger. This is to be expected. A leaving passenger is basically minimizing the expected travel time as a consequence of the train arriving late. A buffer as big as possible is desired to make sure the train is on time. For the parameters used, a buffer bigger than 10 does not have a significant effect anymore, the train is basically guaranteed to arrive on time, so no additional travel time is to be expected.

For a dwell action, the graph is a strictly increasing function, so a buffer as small as possible is desirable. A passenger already in a train has no advantage of buffer placement. Introducing buffers only increases the probability that the train has to wait unnecessary, for example when the previous delays are smaller than the buffer. This is why for a dwelling passenger, a buffer of zero is optimal.

For a transfer action, we see the function has an obvious minimum. If the buffer chosen is too small, the train to which the passenger is traveling to will most likely be missed. When the buffer is chosen too big, the transfer to the next train will almost certainly never be missed, however the passenger does need to wait for an unnecessary long time. The optimal solution is somewhere in between.

As was stated before, for every buffer placed after a dwell edge, both the f^{Depart} and the f^{Dwell} function use the same value of the buffer D as input. So actually the travel time as a consequence of a buffer D equals $f^{Depart}(D) + f^{Dwell}(D)$, when we consider one passenger departing and one passenger departing. The sum of both functions is convex and as a consequence has a unique minimum, as can be seen in the following figure:

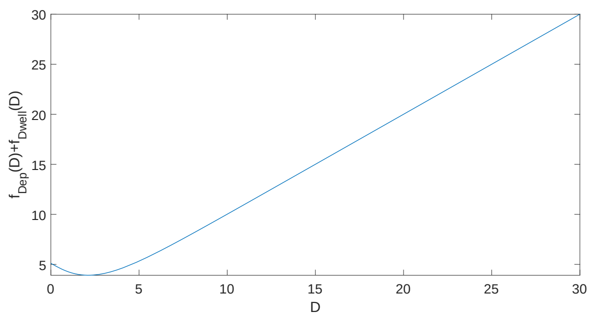


Figure 11: $f^{Depart} + f^{Dwell}$

3.2 Excess journey time

The objective functions developed in the previous sections described the travel time of a passenger starting from the moment that the train the passenger takes should arrive, until the passenger leaves the network. This objective function, however, does not describe the time a passenger waits for its transport when the passenger arrives randomly at the station. Those are passengers that arrive at the station independently of the actual train timetable. Think for example of passengers who go the the station immediately after their last appointment at work is done, which might be any time. The time that passengers have to wait until the scheduled moment of their first form of transport is called excess journey time. This excess journey time will be added to the objective function. By adding this to the objective function, as a result, alternative routes between origin destination pairs will be more spread in time. Of course not every passenger arrives randomly or independent of the timetable. For a certain fraction of passengers, the excess journey time will be added.

We do enforce that trains with a frequency bigger than 1 in the timetable period T are spread perfectly. Meaning that if a train has a frequency $m > 1$, then the inter arrival times of those trains are enforced to be $\frac{T}{m}$. This is because of customer clarity. By adding the objective function described above we will spread trains of different series as good as possible.

Previous research, [5], has also researched adding the excess journey time to the objective function. Although expressions for the excess journey time are easy to achieve, the numerical implementation got quite problematic, as massive computation times prohibited solutions with a satisfactional optimality gap. This was caused by the unknown order of trains during scheduling, to be explained in the following sections, where a lot of constraints had to be added to model this unknown order.

In this section first we will derive expressions for the excess journey time. Afterwards, we will discuss how the problem was modeled in the research of [5]. Last we will introduce a new way of taking into account excess journey time, which is expected to have better computation time.

3.2.1 Deriving excess journey time expressions

We are interested in finding the expected waiting time when a passenger arrives randomly at the station. The expression for excess journey time has been well documented in the literature. In the literature however a stochastic inter arrival time is assumed with known distribution function. For our case the interarrival times are not stochastic as we are considering the time between *planned* arrivals of trains. However we will see that an analogy can be made by considering the approach in literature where stochastic inter arrival times are assumed, and our approach where inter arrival times are deterministic. First we will derive the expected waiting time when considering the deterministic arrival times, afterwards we will derive the way shown in literature and show how the two are connected.

Suppose we have a set of N trains serving the same origin and destination. Suppose that these trains are scheduled at times $b_1, ..., b_N$, with $b_i \leq b_{i+1}\ \forall i$ and $0 \leq b_i \leq T\ \forall i$. Now define s_i as the inter arrival time between b_i and b_{i+1}, so $s_i = b_{i+1} - b_i\ \forall i = 1...N-1$ and $s_N = b_0 - b_N + T$. Given these headway times, the waiting time when arriving at a time t, $0 \leq t \leq T$ is characterized by a sawtooth function $Y(t)$. The following figure expresses this:

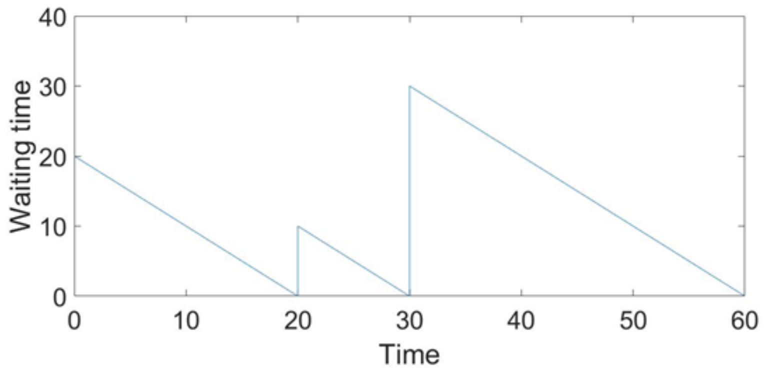


Figure 12: Sawtooth function $Y(t)$ characterizing excess journey time

In this example, the interarrival times are 20, 10 and 30 minutes respectively. The sawtooth function $Y(t)$ states for a given t the waiting time if a passenger were to arrive at time t. Let $f(t)$ be the distribution function of the arrival time of a passenger. Then the expected waiting time is given as $\mathbb{E}(w) = \int_0^T Y(t)f(t)dt$. We assume that passengers arrive uniformly over the interval, so we have $f(t) = \frac{1}{T}$ as distribution function. The expected waiting time now is given by the following expression: $\mathbb{E}(w) = \frac{\int_0^T Y(t)dt}{T}$. Let s_i be the distance between the zero's of the sawtooth function. Naturally, the sum of interarrival times sum up to T. This gives us the following:

$$\mathbb{E}(w) = \frac{\int_0^T Y(t)dt}{T} = \frac{\sum_{i=1}^N \frac{1}{2}s_i^2}{T} = \frac{1}{2T}\sum_{i=1}^N s_i^2.$$

This is the expression that we are going to add in the objective function, for every OD pair where multiple transportation possibilities are possible. Now we will take a look at how expected waiting time is determined in the literature, for example [25]. The general approach is to get an expression for $\mathbb{E}(w)$ using the following:

$$\mathbb{E}(w) = \frac{\text{Expected total passenger waiting time per vehicle departure}}{\text{Expected passengers per vehicle departure}}.$$

It is assumed that the interarrival times between two consecutive forms of transport is stochastic with a distribution function $g(h)$. Furthermore let $n(h)$ be the amount of passengers arriving in a headway of length h and let $w(h)$ be the mean waiting time for passengers arriving in a headway of length h. Then $\mathbb{E}(w)$ can be expressed in the following way:

$$\mathbb{E}(w) = \frac{\int_0^\infty n(h)w(h)g(h)dh}{\int_0^\infty n(h)g(h)dh}.$$

Let the arrival rate of passengers be λ, then $n(h) = \lambda h$. Furthermore, the mean waiting time when arriving in a headway of length h is $w(h) = \frac{h}{2}$, if uniform arrival of passengers is assumed. Plugging in these expressions we get:

$$\mathbb{E}(w) = \frac{\int_0^\infty \lambda \frac{h^2}{2} g(h)dh}{\int_0^\infty \lambda h g(h)dh} = \frac{1}{2}\frac{\int_0^\infty h^2 g(h)dh}{\int_0^\infty hg(h)dh} = \frac{1}{2}\frac{\mathbb{E}(h^2)}{\mathbb{E}(h)}.$$

Note that this expression is the same as the expected residual service time in an $M/G/1$ queue. Also an analogy with the limit of the expectation of residual life in a renewal process can be made. The headway times in our model basically are a sequence of numbers, of which, according to the above, we want the second and first moment. Let the second moment of $[s_1, ..., s_N]$ be defined as $\frac{1}{N}\sum_{i=1}^N s_i^2$ and the first moment as $\frac{1}{N}\sum_{i=1}^N s_i$. Then we get the following:

$$\mathbb{E}(w) = \frac{1}{2}\frac{\frac{1}{N}\sum_{i=1}^N s_i^2}{\frac{1}{N}\sum_{i=1}^N s_i}.$$

Because we know $\sum_{i=1}^N s_i = T$, we get $\mathbb{E}(w)\frac{1}{2T}\sum_{i=1}^N s_i^2$, which is exactly the same expression as already determined before.

The problem with implementing the expressions developed above is the yet unknown order of trains during optimization. The s_i used in the goal function are determined by taking the time difference between two consecutive trains. However, since a timetable is still being constructed, it is unknown which trains are consecutive trains at the station considered. As a consequence, it is unclear how the s_i used in the goal function should be defined. This is the core of the problem. First the modeling of the unknown order problem in [5] will be explained, afterwards a new way of modeling the unknown order of trains is introduced.

3.2.2 Modeling in previous research

Consider a randomly arriving passenger at a station and suppose there are N possible trains the passenger can take in the cycle period to get to its destination. Let $b_1, ..., b_N$ be the time of arrival of trains $1, .., N$. In order to determine the subsequent order of $b_1, ..., b_N$, which are not yet ordered, a new set of variables $\hat{b}_i$, using the same values as b_i are introduced. Here the modulo T values of the starting times, such that $0 \leq b_i \leq T$, are taken. To enforce ordering, $\hat{b}_i \leq \hat{b}_{i+1}$ is added as a constraint. Since the values used in $\hat{b}_i$ are the same as in b_i, a permutation matrix $p \in \mathbb{N}^{N\times N}$ is introduced to enforce the increasing order. This is done by adding the following constraints:

$$\forall i : \begin{cases} \hat{b}_i & \leq \hat{b}_{i+1} \\ \hat{b}_i & = \sum_{j=1}^N p_{ij} b_j \\ p_{ij} & \in \{0,1\}\ \forall j \\ \sum_{j=1}^N p_{ij} & = 1 \\ \sum_{j=1}^N p_{ji} & = 1 \end{cases}$$

Now that $\hat{b}_i$ is of increasing order, s_i can be determined as follows:

$$\begin{aligned} s_i &= \hat{b}_{i+1} - \hat{b}_i && i = 1...N-1 \\ s_N &= \hat{b}_N - \hat{b}_1 + T \end{aligned}$$

Furthermore, because the terms $p_{ij}b_j$ are nonlinear, as both p_{ij} and b_j are undetermined variables, all the terms $p_{ij}b_j$ are linearized using an big-M method to get a linear model. The expression

$\hat{b}_i = \sum_{j=1}^{N} p_{ij} b_j$ in the constraints is replaced by $\hat{b}_i = \sum_{j=1}^{N} h_{ij}$, where the following constraints on h_{ij} are added:

$$\begin{aligned} -T(1-p_{ij}) \leq h_{ij} - b_j &\leq T(1-p_{ij}) \\ 0 \leq h_{ij} &\leq T p_{ij} \end{aligned}$$

Using these constraints has the effect that $h_{ij} = p_{ij} b_j \ \forall i, j$. So if $p_{ij} = 0$ then $h_{ij} = 0$ and likewise if $p_{ij} = 1$ then $h_{ij} = b_j$. However we now only use linear constraints.

As we can see a lot of constraints and binary variables have to be introduced to model the unknown order of trains. This is one of the reasons of the significant increase in computation time. The way of modeling is summarized below:

$$\begin{aligned}
\min \ & \frac{1}{2T} \sum_{i=1}^{N} s_i^2 \\
\text{s.t. } & \hat{b}_i \leq \hat{b}_{i+1} && \forall i = 1, ..., N \\
& 0 \leq b_i \leq T && \forall i = 1, ..., N \\
& p_{i,j} \in \{0, 1\} && \forall i = 1, ..., N, j = 1, ..., N \\
& \sum_{j=1}^{N} p_{i,j} = 1 && \forall i = 1, ..., N \\
& \sum_{j=1}^{N} p_{j,i} = 1 && \forall i = 1, ..., N \\
& \hat{b}_i = \sum_{j=1}^{N} h_{ij} && \forall i = 1, ..., N \\
& s_i = \hat{b}_{i+1} - \hat{b}_i && \forall i = 1, ..., N \\
& -T(1-p_{ij}) \leq h_{ij} - b_j \leq T(1-p_{ij}) && \forall i = 1, ..., N, j = 1, ..., N \\
& 0 \leq h_{ij} \ \leq T p_{ij} && \forall i = 1, ..., N, j = 1, ..., N
\end{aligned}$$

3.2.3 New way of modeling

After taking a considerate look at the modeling of excess journey time in [5], the main problem is expected to be caused by introducing the permutation matrix p. If there are N trains, N^2 binary variables have to be introduced. Furthermore, if N trains are considered, $2N$ constraints on the binary variables introduced in the permutation matrix have to be introduced to make sure that the sum over the rows and columns equals 1, making p an actual permutation matrix. Lastly, for every element p_{ij} , linearization constraints have to be added to linearize the expression $\hat{b}_i = \sum_{j=1}^{N} p_{ij} b_j$. For every p_{ij} , four constraints are needed, resulting in $4N^2$ binary constraints. So in total $4N^2 + 2N$ binary constraints are needed. As can be seen, even for relatively low N, already a lot of extra variables and constraints have to be introduced. This is probably the main cause of the excessive computation times in [5].

The newly proposed method uses the so called cyclic order of trains. [6] defines the cyclic order of a set of trains as the following:

Definition 1. The events 1,...,k, scheduled at times $b_1, ..., b_k$ are said to be cyclically sequenced in the order $1 \rightarrow ... \rightarrow k$ if:

$$0 \leq (b_2 - b_1) \text{mod } T \leq ... \leq (b_k - b_1) \text{mod } T.$$

By considering the cyclic order of trains, we take the periodic nature of the timetable into account. For example, in a cyclic timetable, the order of trains 1,2,3 is exactly the same as the order 2,3,1 and 3,1,2 because the timetable repeats itself. The cyclic order basically stores all these possible equivalent orders into 1 order, namely the cyclic order $1 \rightarrow 2 \rightarrow 3$.

We will now take a look at the amount of cyclic orders possible when there are N trains. This will be necessary in the new way of modeling the unknown order of trains.

Theorem 3. If there are N trains, then there are $(N-1)!$ cyclic orders of trains.

Proof. Proof by induction.

Basis. Suppose we have two trains, then the sequence 1,2 is obviously equivalent with the sequence 2,1, leaving 1 cyclic order. So the base case holds.

Induction step. Assume the induction hypothesis holds for $N = k$, i.e. k trains have $(k-1)!$ different cyclic orders. Now we have to show that $k+1$ trains can be scheduled in $k!$ ways. Consider the k trains, which can be scheduled in $(k-1)!$ ways. Now we have to determine in how many ways we can add the last $k+1^{th}$ train. The $k+1^{th}$ train can be placed before the $1^{st}, 2^{nd}, ..., k^{th}$ train and after the k^{th} train. Leaving us $k+1$ possibilities. However, since placing the train before the 1^{st} train is equivalent with placing the train after the k^{th} train, we actually have k possibilities to place the last train. In total we have $(k-1)!\,k = k!$ possibilities, as was to be shown.

It can be concluded by induction that N trains can be scheduled in $(N-1)!$ different cyclic orders. □

The new way of modeling is based on the observation that if there are N trains, we basically have to choose between $(N-1)!$ cyclic orders. When a cyclic order is chosen, this determines how the s_i used in the goal function are determined. Our decision problem will now consist of which order, out of the $(N-1)!$ possibilities is chosen. We introduce $(N-1)!$ binary variables $w_1, ..., w_{(N-1)!}$, where every binary variable indicates whether or not that specific cyclic order is chosen. As we should select exactly 1 order, we have the following constraint:

$$\sum_{i=1}^{(N-1)!} w_i = 1, \text{ with } w_i \in \{0,1\} \; \forall i.$$

Now that a specific order of trains is enforced, we should make sure that the correct inter-arrival times are used in the goal function for the chosen order. Let the vector g_i represent the beginning times of the trains chosen in the i^{th} order. So $g^i = \{b_{1_i}, ..., b_{N_i}\}$. For example, when 3 trains need to be spreaded, according to theorem 3, there are 2 cyclic orders, namely the cyclic order $1 \rightarrow 2 \rightarrow 3$ and the cyclic order $1 \rightarrow 3 \rightarrow 2$. For this example we get $g^1 = \{b_1, b_2, b_3\}$ and $g^2 = \{b_1, b_3, b_2\}$. Now the interarrival times between trains for order i is the time difference between b^i_{j+1} and b^i_j $\forall j$. However, because the interarrival times are of course between 0 and T, the modulo of the time difference between b^i_{j+1} and b^i_j is taken by defining $g^i_{j,j+1} = b_{i+1} - b_i + nT$ with n an integer and $0 \leq g^i_{j,j+1} \leq T$. So just like in the previous research a modulo operation is needed. Let $s_1, ..., s_N$ be the variables used in the objective function. If the i^{th} order of trains is chosen, the s_j should be equal to the order chosen in g^i. So $s_1 = g^i_{1,2}, ..., s_j = g^i_{j,j+1}, ..., s_N = g^i_{N,1}$. However, this only has to hold in the case that we actually choose this order i. In total we get the following constraints:

$$\sum_{i=1}^{(N-1)!} w_i = 1, \qquad\qquad w_i \in \{0,1\}, \forall i$$
$$(s_k - g^i_{k,k+1})w_i = 0 \qquad\qquad k = 1...N, i = 1, ..., (N-1)!$$

By multiplying the expression $(s_k - g^i_{k,k+1})$ with w_i we make sure that only in case order i is chosen the variables s_k used in the goal function are determined by the order i, so the other possible orders have no influence in this case. We now have introduced $(N-1)!$ binary variables and $(N-1)!\,N$ binary constraints. To give a better understanding, an example will be treated below.

Example 1. Suppose we need to spread 3 trains. Then according to Theorem 3, there are 2 cyclic orders of these trains, namely the order $1 \to 2 \to 3$ and the order $1 \to 3 \to 2$. Let w_1 be the binary variable indicating whether or not cyclic order $1 \to 2 \to 3$ is chosen and let w_2 indicate if the cyclic order $1 \to 3 \to 2$ is chosen. Now g^i is the following: $g^1 = \{b_1, b_2, b_3\}$ and $g^2 = \{b_1, b_3, b_2\}$. It follows that $g^i_{j,j+1}$ is as follows:

$$\begin{aligned} g^1_{1,2} &= b_2 - b_1 + n_1 T & \qquad g^2_{1,2} &= b_3 - b_1 + n_4 T \\ g^1_{2,3} &= b_3 - b_2 + n_2 T & g^2_{1,2} &= b_2 - b_3 + n_5 T \\ g^1_{3,1} &= b_1 - b_3 + n_3 T & g^2_{1,2} &= b_1 - b_2 + n_6 T \end{aligned}$$

The constraints used are the following:

$$\begin{aligned} & w_1 + w_2 = 1 \\ & (s_1 - g^1_{1,2})w_1 = 0 & \qquad & (s_1 - g^2_{1,2})w_2 = 0 \\ & (s_2 - g^1_{2,3})w_1 = 0 & & (s_2 - g^2_{2,3})w_2 = 0 \\ & (s_3 - g^1_{3,1})w_1 = 0 & & (s_3 - g^2_{3,1})w_2 = 0 \end{aligned}$$

Note that if $w_1 = 1$, the values of s_i will be chosen according to the cyclic order g^1 and if $w_2 = 1$, the values of s_i will be chosen according to the cyclic order g^2.

Note that in the expression above, $(s_k - g^i_{k,k+1})w_i$ are nonlinear terms, as undetermined variables are multiplied. A linearization technique is used to replace these nonlinear terms by linear terms, by adding extra constraints.

To do so we are going to introduce a variable $z_{k,i}$. The goal is to replace the nonlinear constraint $(s_k - g^i_{k,k+1})w_i = 0$ by $z_{k,i} = 0$. However, $z_{k,i}$ should of course take the exact same values as $(s_k - g^i_{k,k+1})w_i$. This is enforced by adding the following constraints:

$$LB \cdot w_i \leq z_{k,i} \leq UB \cdot w_i$$
$$(s_i - g^i_{k,k+1}) - UB(1 - w_i) \leq z_{k,i} \leq (s_i - g^i_{k,k+1}) + (1 - w_i)LB$$

Here UB and LB are upper and lower bounds of $(s_k - g^i_{k,k+1})w_i$. As both we have $0 \leq s_k \leq T$ and $0 \leq g^i_{k,k+1} \leq T$, the lowerbound LB can be chosen as $-T$ and the upper bound as T. The constraints enforce that if $w_i = 1$, then $z_{k,i} = s_k - g^i_{k,k+1}$. On the other hand if $w_i = 0$, then $z_{k,i} = 0$.

The goal function and their constraints now look like the following, if we have to spread N trains:

$$\begin{aligned}
\min\ & \frac{1}{2T}\sum_{i=1}^{N} s_i^2 \\
\text{s.t.}\ & \sum_{i=1}^{(N-1)!} w_i = 1, \\
& w_i \in \{0,1\} && i = 1, ..., (N-1)! \\
& z_{k,i} = 0, && k = 1, ..., N, i = 1, ..., (N-1)! \\
& -Tw_i \leq z_{k,i} \leq Tw_i && k = 1, ..., N, i = 1, ..., (N-1)! \\
& (s_i - g^i_{k,k+1}) - T(1-w_i) \leq z_{k,i} \leq (s_i - g^i_{k,k+1}) - T(1-w_i) && k = 1, ..., N, i = 1, ..., (N-1)! \\
& 0 \leq g^i_{j,j+1} \leq T && j = 1, ..., N, i = 1, ..., (N-1)! \\
& g^i_{j,j+1} = b^i_{j+1} - b^i_j + nT && j = 1, ..., N, i = 1, ..., (N-1)!
\end{aligned}$$

For both models integer variables are needed to make sure the expressions are modulo T, so between 0 and T, using the same amount of constraints. The amount of binary variables and constraints needed in both models will now be compared, as the amount of binary variables and constraints are expected to have the most significant influence on computation time compared to continuous variables. As was shown before, the model of [5] needs to introduce N^2 binary variables and $4N^2 + 2N$ binary constraints. The newly proposed model needs $(N-1)!$ binary variables to consider the possible cyclic orders. $(N-1)!\,N = N!$ constraints are needed to make sure the right interarrival times is chosen, corresponding to the chosen order of trains. Every one of these constraints is linearized, using 4 inequality constraints, resulting in $4N!$ constraints. 1 additional constraint is necessary to make sure that exactly 1 order is chosen. This results in the following table:

		$N = 2$	$N = 3$	$N = 4$
Sels	Variables=N^2	4	9	16
	Constraints=$4N^2 + 2N$	20	42	72
Arwyn	Variables=$(N-1)!$	1	2	6
	Constraints=$4N!+1$	9	25	97

Table 1: Amount of binary variables and constraints for both models

For $N = 2$ and $N = 3$ the new model is expected to perform faster, it is unclear which model is expected to perform better for $N = 4$. The older model has a lot more binary variables, but needs less constraints. Note that needing less binary variables and constraints does not guarantee for better computation times. However, using the newly proposed model, the same solution space is described with less variables. This means there are less branches to be examined by a solver using a branch and bound approach, which is expected to have positive effects om computation times. A branch and bound algorithm is the standard solution methods for Mixed Integer Programs, which will be elaborated on in the results section.

3.3 Objective

Now all expressions needed for the goal function have been derived, so the total goal function can be composed. First an expression for the expected travel time without excess journey time will be explained. Afterwards the excess journey time will be added. Every passenger traverses a certain route when traveling from his origin to destination. Every routes consists of a set of consecutive actions, described in the previous section. The total travel time is the sum of the travel time on these actions. Let a route t be a set of actions, so $t = \{t_1, ..., t_n\}$, where every t_i is a dwell, transfer or depart action. For a passenger following a route t, its expected travel time is:

$$\mathbb{E}\left(R^t\right) = \sum_{i=1}^{|t|} f^{t_i}(D_{t_i}).$$

Here $|t|$ is the size of the vector t, so the amount of actions needed for route t. Suppose that an amount op w_t follow a route t. Then the total cost of route t is w_t multiplied by the above expression. Furthermore, we want to consider all possible routes and sum over them. Let RT be the set of all possible routes. This results in the following expression for the Total Traveltime TT:

$$\mathbb{E}(TT) = \sum_{t\in RT} w_t \mathbb{E}\left(R^t\right) = \sum_{t\in T}\sum_{i=1}^{|t|} w_t f^{t_i}(D_{t_i}).$$

The excess journey time has not been incorporated in the expression for TT yet. Let SP be the total set of train pairs that should be spreaded where an element $p = ([p_1, ..., p_N], w_p) \in SP$ is a tuple storing the specific trains, $p_1, ..., p_N$ that should be spreaded and the amount of passengers using these trains, w_p. Furthermore, we assume a fraction a, $0 \leq a \leq 1$, of all passengers arrive randomly, so for these passengers the excess journey time should be added to the goal function. A fraction $1-a$ of the passengers adjust their arrival time at the station according to the timetable, so no excess journey time should be modeled for these passengers. We get the following:

$$\mathbb{E}(TT) = \sum_{t\in RT}\sum_{i=1}^{|t|} w_t f^{t_i}(D_{t_i}) + a\sum_{p\in SP}\sum_{i=1}^{p_N} w_p \frac{1}{2T} s_i^2.$$

3.4 Constraints

Several constraints need to be taken into account to get an operational feasible schedule. In order to do so, we need to get expressions for the time that every event, or node in the graph formulation, takes place. Depending on the buffers D chosen, expressions for the time of the events can be determined. The constraints necessary to guarantee an operational feasible schedule are with regards to those event times.

Let us first consider edges regarding train activities. The length, or time, of an edge connecting two events or nodes, is the time difference between these events. Every edge or activity has a minimum time m_e, for example the technical minimum time that is necessary for a train to travel between two consecutive stations. On top that, for a dwell edge, there is a buffer D_e. Let b_e be the starting time of edge e and let e_e be the ending time of edge e. Then the following constraints are added to the model:

$$\begin{aligned} \forall e \in E_{Ride}: &\ b_e + m_e = e_e \\ \forall e \in E_{Dwell}: &\ b_e + m_e + D_e = e_e. \end{aligned}$$

Suppose we have an edge e_1 and an edge e_2, for which the ending node of e_1 is the same as the beginning node of e_2. Let the function $q(e)$ give the starting node of a directed edge e and the function $r(e)$ give the ending node of a directed edge e. So in the case described, we have $q(e_2) = r(e_1)$. For these cases it obviously holds that $e_{e_1} = b_{e_2}$. .

Now transfer edge constraints will be considered. A transfer edge is an edge giving a passenger the possibility to transfer from train A to train B. On these edges also a buffer is chosen. Since train A and train B also occur an arbitrary multiple of the timetable period, the following constraints are added:

$$\forall e \in E^{Transfer}: \; b_e + m_e + D_e + n_e T = e_e, \text{ with } n_e \in \mathbb{N}. \tag{1}$$

Also a very trivial constraint is that we enforce every D to be positive and smaller or equal to the timetable period:

$$\forall e : 0 \leq D_e \leq T.$$

Now let us consider the headway constraints. A headway constraint is a safety constraint, making sure that two trains are scheduled a minimum time apart from eachother if they use the same piece of infrastructure. This has the goal that there will be no collisions between trains. NS has a dataset indicating which pair of trains have headway constraints with eachother, at what station and what the corresponding minimum headway time is. Suppose that two trains have a headway constraint $h \in H$ between two nodes. Where H is the total set of headway pair constraints. Let b_{h_1} be the time of this node for train 1 and let b_{h_2} be the time of the node for train 2. Furthermore, let h_{Min} be the minimum time demanded between the two trains. We are going to construct two directed edges, one from node 1 to node 2, and one from node 2 to node 1. For both edges we model a supplement time s_{h_1} and s_{h_2}, which is the time between the events. Furthermore, because of the cyclic nature of the timetable, we also have to take into account periodicity. This results in the following constraints:

$$\forall h \in H : \begin{cases} b_{h_1} + s_{h_1} + n_{h_1} T = b_{h_2} \\ b_{h_2} + s_{h_2} + n_{h_2} T = b_{h_1} \\ h_{Min} \leq s_{h_1} \leq T - h_{Min} \\ h_{Min} \leq s_{h_2} \leq T - h_{Min} \\ n_{h_1}, n_{h_2} \in \mathbb{N} \end{cases}$$

When these constraints are met, the trains are scheduled at least a time h_{Min} apart from each other. The constraints explained so far are all mandatory to get a timetable which meets all demands. Furthermore, previous research has shown that by introducing additional constraints, computation time will decrease. This will be treated in the following.

[5] showed that using results from [9], we can narrow down the search space of the integer variables for transfer actions, so the integers used in equation 1. This is the case when two transfers together form an hourglass. An hourglass is defined as a set of dwell and transfer edges which together have the following form:

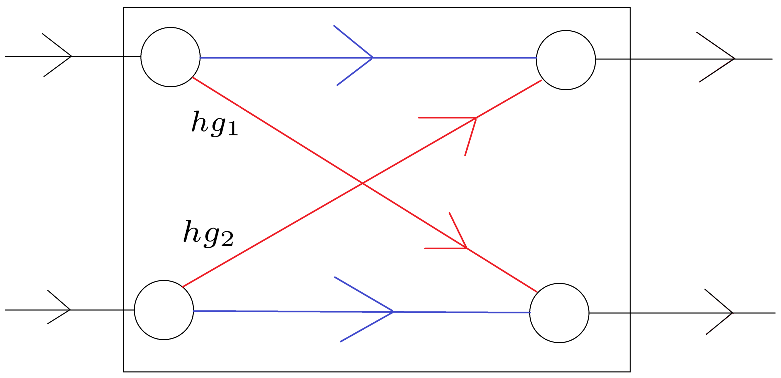


Figure 13: Example of an hourglass construction

It was shown that for those cases specific constraints on the integer variables can be added. It was shown that a scheduling problem has a solution if and only if these constraints hold. So adding these constraints does not reduce the solution space. Let HG be the set of hourglass cycles. Then the following constraints can be added:

$$\forall hg \in HG : -1 \leq n_{hg_1} + n_{hg_2} \leq 1.$$

Here n_{hg_1} is the integer variable modeling the periodicity of a transfer action of edge hg_1 and similarly for n_{hg_2}. Adding these constraints reduces the search space, hopefully positively influencing computation time, but does not affect the quality of the solution.

Concluding, we have the following mandatory constraints:

$$\begin{aligned}
&\forall e \in E_{Ride}: \; b_e + m_e = e_e \\
&\forall e \in E_{Dwell}: \; b_e + m_e + D_e = e_e \\
&\forall e \in E_{Transfer}: \; b_e + m_e + D_e + n_e T = e_e, \text{ with } n_e \in \mathbb{N} \\
&\forall h \in H : \begin{cases} b_{h_1} + s_{h_1} + n_{h_1} T = b_{h_2} \\ b_{h_2} + s_{h_2} + n_{h_2} T = b_{h_1} \\ h_{Min} \leq s_{h_1} \leq T - h_{Min} \\ h_{Min} \leq s_{h_2} \leq T - h_{Min} \\ n_{h_1}, n_{h_2} \in \mathbb{N} \end{cases} \\
&\forall e : 0 \leq D_e \leq T \\
&\forall e \in E : b_e \geq 0 \\
&\forall (e_1, e_2) \in E \times E \text{ such that } r(e_1) = q(e_2) : e_{e_1} = b_{e_2},
\end{aligned}$$

and the following optional constraints to speed up computation

$$\forall hg \in HG : -1 \leq n_{hg_1} + n_{hg_2} \leq 1.$$

Connection with PESP

In the beginning if this report, PESP was introduced as the common way to mathematically formulate timetabling problems. Though the constraints used in this model are quite similar, there are also some differences. First of all the time of an event is not forced to be between 0 and T, it can be anything bigger or equal to 0. In doing so, it is not necessary to model integer constraints in the form of $b_e + m_e + n_e T = e_e$ for edges concerning train activities i.e. dwell and ride edges, resulting in less integer constraints. Only for transfer edges integer variables are used to model periodicity. Furthermore, PESP assumes a certain given lower bound and an upper bound for the duration of every specific activity. For this model, we do have a lower bound, the technical minimum time, but the upper bound is not specific for every edge. This is because on an edge we allow the buffer to be maximally T. So for every edge the lower bound is m_e and the upper bound is $m_e + T$. Because of taking the expected passenger travel time as a goal function, the buffers used will not become very large and thus a tighter upper bound is not necessary. Of course if desired, tighter upper bounds could be added.

4 Piecewise linearization

In the previous chapter, the optimization model has been formulated. To be able to solve the model on large networks, the problem will be approximated by a piecewise linear problem, reducing computation time. First a piecewise linearization algorithm will be described. Afterwards it will be shown how the goal function when Weibull distributed delays are assumed can still be used in the optimization problem by making use of numerical integration. Last, error bounds of the approximated problem with respect to the original problem will be given.

4.1 Piecewise linearizing a function

In the previous sections we have explained what the total goal function is and what the required constraints are. Now we want to solve this optimization problem. There a several things to consider. First, we have a Mixed Integer Program (MIP) to be solved. MIP are known to be far more difficult to solve than regular Linear Programs (LP), as they are generally NP-hard problems. Furthermore, the goal function is nonlinear and the transfer action part of the objective function is non-convex. All these aspects together raises the question whether or not this problem is numerically solvable, and if so in acceptable time.

One common technique when solving problems which have a non-linear objective function is to make a piecewise-linear approximation of the goal function. This way, computation time is reduced as the problem can be formulated as a Mixed Integer Linear Program (MILP), but the approximated goal function can still resemble the actual goal function quite good.

Piecewise linearization is a technique that is used in a wide variety of fields. For example image processing, data fitting and economics. Several algorithms have been developed to determine a piecewise-linear approximation of a known non-linear function $f(x)$. Most of these algorithms use a criterium of the quality of the linearized function compared to the original function. Several criteria can be thought of:

1. Minimize the squared error between the original function and the approximation
2. Minimize the maximum difference between original function and the approximation
3. Minimize the integral of the difference between the original function and the approximation
4. Minimize the integral of the squared difference between the original function and the approximation
5. Equal distribution of curvature between line segments in the approximation

Commonly the optimality criterium used is dependent on the nature of the problem considered. As we are considering an optimization problem, we will consider if a logical criterium can be thought of when making a piecewise-linear approximation used for an optimization problem.

4.1.1 In the context of optimization

Suppose we have a non-linear smooth function $f(x)$ defined on the interval $[a, b]$, which needs to be piecewise-linearized to be used in linear programming solvers. A piecewise-linear function is a collection of linear functions, all defined at a subinterval of the interval $[a, b]$. A piecewise linear function consists of n linear segments with corresponding breakpoints $a_1, ..., a_{n+1}$. Let the i^{th} linear segment be given by the function $f_i(x)$. Mathematically speaking a piecewise-linear function f^{Lin} consisting of n linear segments can be defined as follows:

$$f^{Lin}(x) = \sum_{i=1}^{n} f_i(x)\mathbb{1}\{x \in [a_i, a_{i+1}]\}.$$

As we are using the piecewise linear approximation in the context of optimization, the definition of a "good" approximation should be chosen accordingly. We wish to choose a piecewise linear approximation for which the solution of the linearized problem is as close as possible to the actual solution of the problem. Suppose we have the following problem:

$$\begin{aligned} \min_x \ & f(x) \\ \text{s.t.} \ & Ax = b \\ & x \geq 0 \end{aligned}$$

This problem has some optimal solution $\hat{x}$. Suppose that we need to piecewise linearize $f(x)$ into some $f^{Lin}(x)$ for this problem. We should chose a piecewise linear approximation algorithm such that the norm of the difference between the value of the approximated and original problem is minimized. Suppose we have a set of linearization techniques $\{Lin_1, ..., Lin_n\}$. Then we wish to determine the linearization technique as follows:

$$\begin{aligned} \min_{Lin_i} \ & |f(\hat{x}) - \min_x f^{Lin_i}(x)| \\ \text{s.t.} \ & Ax = b \\ & x \geq 0 \end{aligned}$$

Now consider the following example, suppose we wish to solve the following problem:

$$\begin{aligned} \min_x \ & f(x) \\ \text{s.t.} \ & x \geq -2 \end{aligned}$$

Where $f(x)$ is plotted in the picture below.

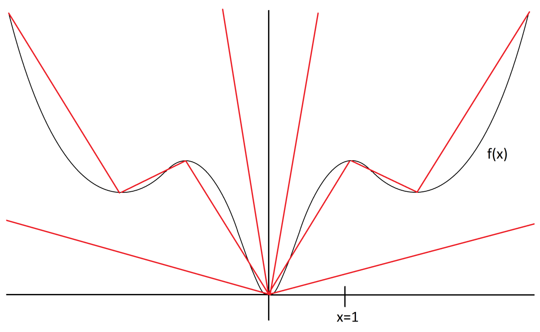


Figure 14: Example of linearization

Then according to definition, any of the red piecewise linear approximations are equally good and actually optimal. This is because the optimum is at $\hat{x} = 0$ for every piecewise linear approximation. However, for other constraints, for example $x \geq 1$, the quality of the approximations differ significantly between the different piecewise approximations.

Suppose for the train buffer allocating problem we have to chose n buffers to minimize the expected travel time. So we have a function $f(D^n) : \mathbb{R}^n \to \mathbb{R}$. As we have seen before, $f(D^n)$ is a separable function, meaning it can be written as $f(D^n) = \sum_{i=1}^n \hat{f}_i(D_i)$. However it is not necessarily true that *opt* $D^n = \cup_{i=1}^n \arg\min_{D_i} \hat{f}_i(D_i)$, because of the possible constraints on D. This means that it is important that the approximations $\hat{f}_i^{Lin}(D_i)$ of $\hat{f}_i(D_i)$ should also be "good" for $D_i \neq \arg\min \hat{f}(D_i)$. As it is not directly clear what the search space resulting from the constraints will be, the approximation should be good everywhere along the curve. We can conclude that we need an approximation technique which is as close as possible to f everywhere on $[a, b]$.

4.1.2 Algorithm

We will now consider an piecewise linearization algorithm described in [26]. The goal of the algorithm is to choose, for a given amount of linear segments, the breakpoints between segments such that the squared error between the original function and the piecewise-linear approximation is minimized.

We are going to represent $f(x)$ by a finite point set $P = \{p_1, ..., p_n\} = \{(x_1, y_1), ..., (x_n, y_n)\} = \{(x_1, f(x_1)), ..., (x_n, f(x_n))\}$. If the interval of f is $[a, b]$, then $(x_{i+1} - x_i) = \frac{b-a}{n}$, so the points are at regular distances in x direction from eachother. Given this pointset, we need to choose a set of $T < n$ lines to approximate the data points. The piecewise linearization is characterized by the breakpoints $\{l_1, ..., l_{T+1}\} \subset P$. We want to minimize the squared difference between the original pointset P and the piecewise approximation. Let the i^{th} linear segment be described by $f_i^{Lin}(x) = a_i x + b_i$, where $f_i(x)$ connects the points l_i with l_{i+1} in a piecewise linearization. Note that there can be points between l_i and l_{i+1}, namely $l_i, p_j, ..., p_{k,} l_{i+1}$ for some j and k. So the squared error for the i^{th} line segment connecting the points l_i and l_{i+1} is as follows:

$$\sum_{k=l_i}^{l_{i+1}} [f(x_k) - f_i^{Lin}(x_k)]^2 = \sum_{k=l_i}^{l_{i+1}} [f_i(x_k) - a_i x_k - b_i]^2.$$

As we want to choose the breakpoints such that the entire squared difference along all segments is minimized, we have the following problem

$$\min_{l_1, ..., l_{T+1}} \sum_{i=1}^{T} \sum_{k=l_i}^{l_{i+1}} [f(x_k) - f_i^{Lin}(x_k)]^2.$$

To solve this problem, a dynamic programming approach is used. To do so, let $F(j, t)$ be the minimum value of the expression described above to approximate the points $\{p_1, ..., p_j\}$ with exactly t line segments. Furthermore, let $E(i, j)$ be the minimum squared error of approximating $\{p_i, ..., p_j\}$ with exactly 1 segment. Now it can be shown that the following dynamic programming relation yields the minimum squared error approximation:

$$F(j, t) = \min_{t \leq i \leq j} F(i, t-1) + E(i, j).$$

A full description of the algorithm in pseudo code can be found in Appendix C.

Next we will show some results of what a piecewise-linear approximation looks like when using the algorithm described above when different amounts of segments are allowed to model a transfer action.

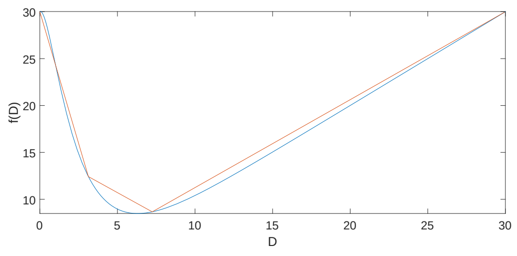


Figure 15: Using 3 segments

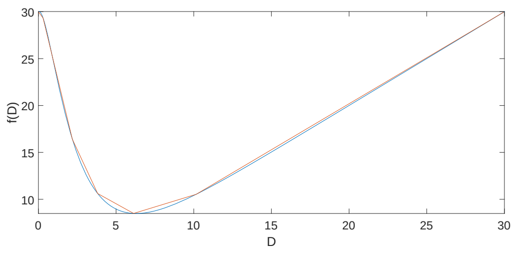


Figure 16: Using 6 segments

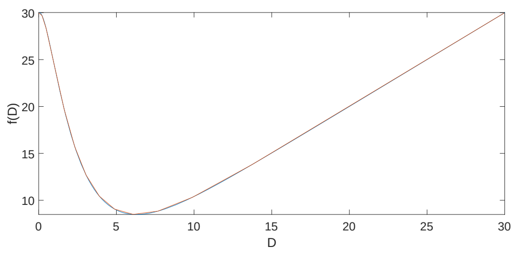


Figure 17: Using 12 segments

The more segments allowed, the better the approximation is.

4.2 Numerical integration

In the Chapter 3 we found expressions for the expected delay of certain actions where both exponential and Weibull distributed disturbances were considered. For the exponential distributions, closed form expressions were found. For the Weibull distribution however, we found expressions which were not analytically integrable, for example the depart action:

$$f^{Depart}(D) = \left(\lambda_y \Gamma\left(1 + \frac{1}{k_y}\right) - D\right) e^{-\left(\frac{D}{\lambda_x}\right)^{k_x}} + \lambda_x \Gamma\left(1 + \frac{1}{k_x}, \left(\frac{D}{\lambda_x}\right)^{k_x}\right)$$
$$+ \int_0^D \left[(x - D)e^{-\left(\frac{D-x}{\lambda_y}\right)^{k_y}} + \lambda_y \Gamma\left(1 + \frac{1}{k_y}, \left(\frac{D-x}{\lambda_y}\right)^{k_y}\right)\right] f_x(x)dx.$$

Using these expressions as a goal function in the current form is not possible. However, we can work our way around this problem. In the previous chapter it was explained that the goal function needs to be piecewise linearized, and to do so we need to know the value of $f(x)$ in a finite set of points $P = \{(x_1, y_1), ..., (x_n, y_n)\}$. In other words, it is not necessary to know $f(x)$ for every x, just for the set $x_1, ..., x_n$. This gives us the possibility to find, by numerical integration, expressions for $f(x_i)$, $i = 1, ..., n$ when the delays are Weibull distributed. This way, we can still use the Weibull goal function during optimization. Our goal is now to find algorithms to numerically integrate functions in order to determine $f(x_i)$.

4.2.1 Simpson's integration rule

One of the most common approaches to numerically integrate a function is to use Simpsons's rule. The idea of Simpsons rule is quite simple. We need to determine some integral $\int_a^b f(x)dx$. The interval $[a,b]$ is divided into n subintervals of length $\frac{b-a}{n}$, with n being an even number. Let $f(x_i)$ be the values of f at the bounds of these subintervals. Furthermore, for every subinterval, we also store its midpoint value i.e. $f(\frac{x_i+x_{i+1}}{2})$. Now for every subinterval, a 2^{nd} order polynomial is fitted on the points $f(x_i), f(\frac{x_i+x_{i+1}}{2})$ and $f(x_{i+1})$. For a 2^{nd} order polynomial it is very easy to determine it's integral. If we now take the sum of the integrals of these subintervals, we get an approximation of the interval over the interval $[a,b]$. This results in the following formula:

$$\int_a^b f(x)dx \approx \frac{h}{3}\left[f(x_0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n)\right].$$

The bigger the amount of different subintervals considered, the better the approximation will become. The absolute error for the composite Simpsons rule is bounded by $\frac{h^4}{180}(b-a)\max_{\xi\in[a,b]}\left|f^{(4)}(\xi)\right|$, see for example [27].

4.2.2 Monte-Carlo integration

An other common approach is Monte-Carlo integration. The idea of Monte-Carlo integration is that if we evaluate the to be integrated function at enough random points and taking the average, we will end up with the value of the integral. Again we want to determine an integral of the form $\int_a^b f(x)dx$. Now the Monte-Carlo estimate z of $\int_a^b f(x)dx$ is defined as follows:

$$z = \frac{b-a}{N}\sum_{i=1}^{N} f(x_i).$$

Here x_i are samples, drawn from the $Uniform(a,b)$ distribution. A $Uniform(a,b)$ distribution has $\frac{1}{b-a}$ as distribution function. We will now show that the expectation of z is $\int_a^b f(x)dx$:

$$\begin{aligned}\mathbb{E}(z) &= \mathbb{E}\left(\frac{b-a}{N}\sum_{i=1}^{N} f(x_i)\right)\\ &= \frac{b-a}{N}\sum_{i=1}^{N}\mathbb{E}(f(x_i))\\ &= \frac{b-a}{N}\sum_{i=1}\int_a^b f(x)\frac{1}{b-a}dx\\ &= \frac{1}{N}\sum_{i=1}^{N}\int_a^b f(x)dx\\ &= \int_a^b f(x)dx.\end{aligned}$$

As is shown, z is an consistent estimator of the integral. We can improve the above technique however. In the above, samples were drawn from a uniform distribution. It might be better take more samples from an area that is more "important" to get more detailed information about it. This results in a technique called Importance Sampling.

Importance Sampling

Now instead of sampling our samples x_i from a uniform distribution, we choose to sample the samples from a certain distribution $q(x)$, with $x \in [a,b]$. Now the Monte-Carlo estimate of the integral is as follows:

$$z = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{q(x_i)}.$$

We will again show that the expectation of z is equal to the integral:

$$\begin{aligned}
\mathbb{E}(z) &= \mathbb{E}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{q(x_i)}\right)\\
&= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left(\frac{f(x_i)}{q(x_i)}\right)\\
&= \frac{1}{N}\sum_{i=1}^{N}\int_a^b \frac{f(x)}{q(x)}q(x)dx\\
&= \frac{1}{N}\sum_{i=1}^{N}\int_a^b f(x)dx\\
&= \int_a^b f(x)dx.
\end{aligned}$$

So also the estimator when using importance sampling is consistent. Note that if we took $q(x)$ to be the uniform distribution function, we would end up with exactly the expression stated at the beginning of the Monte-Carlo integration section. A proper choice for the function $q(x)$ for our example would be to sample according to the given Weibull distribution. Importance Sampling has been proven to have lower variance between the actual value and the approximation than using uniform sampling.

Comparing Simpson's rule with Monte-Carlo integration

We will now use the Simpson algorithm and the Monte-Carlo importance sampling algorithm to use of the Weibull distributed delays f^{Depart} function. For the Monte-Carlo importance sampling, samples are generated by generating Weibull distributed samples. Both solutions are graphed in the following figure:

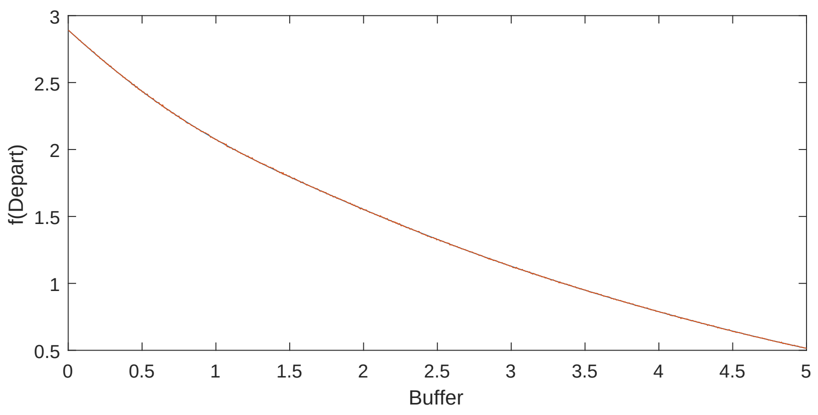


Figure 18: Monte-Carlo integration and integration using Simpson's rule with $\lambda_x = 2, \lambda_y = 1, k_x = 1.1, k_y = 1.7$

Both the Monte-Carlo integration and Simpsons integration give similar results and at first sight, there is no difference between the graph of the two. However, when zooming in on the solution, we see the following.

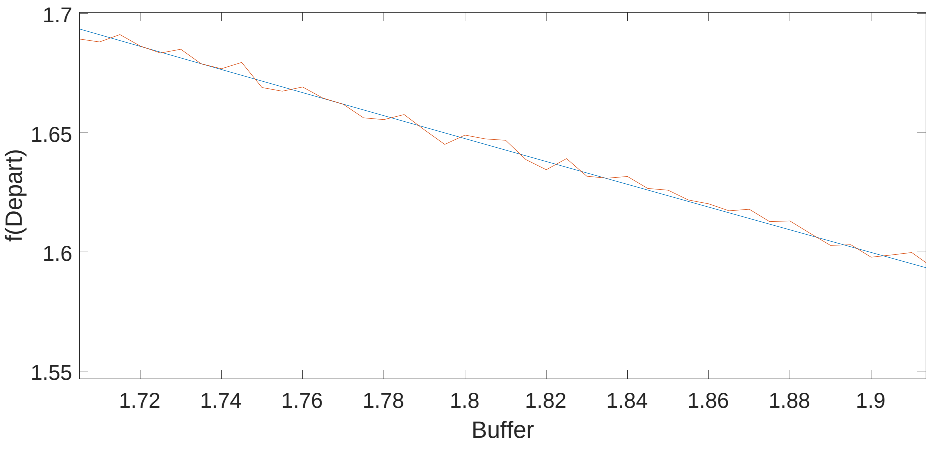


Figure 19: Zoomed in on figure 18, blue line is Simpson's solution, orange line is Monte-Carlo solution

As can be seen, the Monte-Carlo solution has quite some variation between consecutive points and seems to center around the Simpsons's rule solution. Also, the Monte-Carlo solution was computationally much heavier. We conclude that for this case, Simpsons rule is the best choice for numerical integration. Monte-Carlo is especially useful when multidimensional integrals are considered. Here Simpsons rule suffers from the curse of dimensionality.

4.2.3 Integrate linearized function

In the previous sections the approaches were as follows. We need to piecewise-linearize the expectation of a certain function, however determining the expectation was not possible because no analytical solution could be found. The approach was to numerically integrate the function at a certain set of points, after which a piecewise-linearization could be made. An other approach could be the first linearize the function to be integrated. That way, an integral of a piecewise-linear function should be determined, which is an easy problem.

Suppose we want to get an expression for $\int_0^D g(x)dx$ and suppose that $g(x)$ was linearized into $g^{Lin}(x)$ using a piecewise-linearization algorithm. This means that $g^{Lin}(x)$ can be represented by a set of points $\{(x_0, y_0), ..., (x_n, y_n)\}$. Then $\int_0^D g(x)dx \approx \int_0^D g^{Lin}(x)dx$. For $x_j \leq D \leq x_{j+1}$ we get the following:

$$\int_0^D g^{Lin}(x)dx = \sum_{i=0}^{j-1} \frac{\frac{1}{2}(y_{i+1} + y_i)}{x_{i+1} - x_i} + \frac{D - x_j}{x_{j+1} - x_j} y_{j+1} + \left(1 - \frac{D - x_j}{x_{j+1} - x_j}\right) y_j.$$

The summation describes the integral over the interval $[0, x_j]$, the last two terms are the integral over the area $[x_j, D]$. This integral expression is again a piecewise linear function. Note that this way, no numerical integration was performed. So this could be an interesting way to linearize an integral that is not analytically determinable. However, for our problem we need to determine an integral of the form $\int_0^D g(x, D)dx$ instead of $\int_0^D g(x)dx$. As a consequence, it is quite hard to find a piecewise linearization of $g(x, D)$ as it is dependent on 2 variables instead of 1. The technique described above could be very useful for other problems.

4.3 Error bounds for piecewise linear approximation

In the previous chapters, it was shown that we make a piecewise-linear approximation of the goal function because of computational considerations. Since the new problem is an approximation of the original problem, the solution to the problem will of course also be an approximation. In this chapter we will determine if we can give give bounds on the derivation between the original problem and the approximated problem in terms of the goal function.

We will use the following theorem by [28].

Theorem 4. Let $\hat{x}$ be the optimal solution to the problem $\min f(x)$, s.t. $x \in X$, where X is a nonempty set. Furthermore, let x^* be the optimal solution to the piecewise-linear approximation of the problem $\min f^{Lin}(x)$, s.t. $x \in X$, where possibly $\hat{x} \neq x^*$. If $|f(x) - f^{Lin}(x)| \leq \epsilon$, $\forall x \in X$, then $|f(\hat{x}) - f^{Lin}(x^*)| \leq \epsilon$.

Proof. Suppose $|f(x) - f^{Lin}(x)| \leq \epsilon$, $\forall x \in X$. This can also be written as:

$$f^{Lin}(x) - \epsilon \leq f(x) \leq f^{Lin}(x) + \epsilon \ , \forall x \in X.$$

1). Consider the first inequality i.e. $f(x) \geq f^{Lin}(x) - \epsilon$, $\forall x \in X$. Then in particular $f(\hat{x}) \geq f^{Lin}(\hat{x}) - \epsilon$. Because x^* is the optimal solution of f^{Lin}, we have $f(\hat{x}) \geq f^{Lin}(\hat{x}) - \epsilon \geq f^{Lin}(x^*) - \epsilon$.

2). Consider the second inequality i.e. $f(x) \leq f^{Lin}(x) + \epsilon$, $\forall x \in X$. Then in particular $f(x^*) \leq f^{Lin}(x^*) + \epsilon$. Because $\hat{x}$ is the optimal solution of $f(x)$, we have $f(\hat{x}) \leq f(x^*) \leq f^{Lin}(x^*) + \epsilon$

So we have $f^{Lin}(x^*) - \epsilon \leq f(\hat{x}) \leq f^{Lin}(x^*) + \epsilon$, which can also be written as $|f(\hat{x}) - f^{Lin}(x^*)| \leq \epsilon$. ☐

4.3.1 Error bounds for Weibull goal function

Given this theorem, a simple check to determine bounds on the difference between the original problem and the approximated problem is possible. For a given linearization f^{Lin} and original function f, determine $\max_{x \in X} |f(x) - f^{Lin}(x)|$, then this is the maximum difference between the two optimal solutions. However, in the case that we use a Weibull distribution, $f^{Lin}(x)$ is not known for all $x \in X$. Because we have to numerically integrate, we only know $f^{Lin}(x)$ at a finite set of points $\{x_1, ..., x_n\}$. So $\max_{x \in X} |f(x) - f^{Lin}(x)|$ can also not be determined. Suppose $|f(x_i) - f^{Lin}(x_i)| \leq \epsilon$, $\forall i$ and some ϵ. Then we can not conclude that $|f(\hat{x}) - f^{Lin}(x^*)| \leq \epsilon$, as $|f(x) - f^{Lin}(x)|$ can in theory be arbitrarily big (in the points where the value of $f(x)$ is unknown), and thus possibly also the difference between the two optimal solutions. However, by making use of the Lipschitz continuity of the function f, we will give bounds on the difference between the two solutions.

Theorem 5. Let f^{Lin} be a piecewise-linear approximation of f, defined on the interval $[a, b]$. Let the value of f be known at a finite set $\{x_1, ..., x_n\}$, where the distance between consecutive x_i is regular with distance q and $x_1 = a$ and $x_n = b$. Suppose that f is Lipschitz continuous with Lipschitz constant K. Let $\hat{x}$ be the optimal solution of $\min_{x \in X} f(x)$ and x^* the optimal solution of $\min_{x \in X} f^{Lin}(x)$. Then the following holds:

$$|f(\hat{x}) - f^{Lin}(x^*)| \leq \max_{x_i \in \{x_1, \ldots, x_n\}} |f(x_i) - f^{Lin}(x_i)| + K \cdot q.$$

Proof. By theorem 4, we are looking for the maximum difference between $f(x)$ and $f^{Lin}(x)$. Let us consider two consecutive known points of f, some x_i and x_{i+1}. We will first consider if we can determine the maximum difference between f and f^{Lin}, when $x \in [x_i, x_{i+1}]$. The following illustration shows the approach used in the proof:

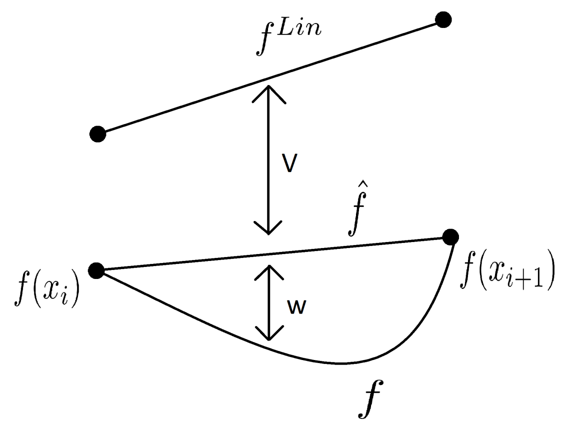


Figure 20: Illustration of maximum difference

The distance between f and f^{Lin} is the distance of $V + W$. Here $\hat{f}$ is a linear line connecting $f(x_i)$ and $f(x_{i+1})$. We will get expressions for both V and W.

V is the distance between f^{Lin} and $\hat{f}$. Since both functions are linear lines, the maximum value will be attained at the boundaries of the interval, so either at x_i or x_{i+1}. So V is maximally $\max\{|f(x_i) - f^{Lin}(x_i)|, |f(x_{i+1}) - f^{Lin}(x_{i+1})|\}$.

For the distance W we will make use of the Lipschitz continuity of f. First notice that $W = \max_x\{|f(x) - \hat{f}(x)|\} \leq \max\{|f(x) - \hat{f}(x_i)|, |f(x) - \hat{f}(x_{i+1})|\}$, because $\hat{f}(x)$ is a linear function. A function f is Lipschitz continuous if $\exists K$ such that $|f(x) - f(y)| \leq K|x - y|$ $\forall x, y$. By making use of Lipschitz continuity, we can say the following:

$$\begin{aligned} |f(x) - \hat{f}(x_i)| &= |f(x) - f(x_i)| \\ &\leq K|x - x_i| \\ &\leq K|x_i - x_{i+1}| \\ &= K \cdot q. \end{aligned}$$

A similar argument can be made to show $|f(x) - \hat{f}(x_{i+1})| \leq K \cdot q$. So the distance between f and $\hat{f}$ is bounded by $K \cdot q$.

The above was determined by considering only the interval between two consecutive points x_i and x_{i+1} with $[x_i, x_{i+1}] \subset [a, b]$. We are of course interested in the interval $[a, b]$. The bound of W between f and $\hat{f}$ holds for every two consecutive points x_i and x_{i+1}, so this bound stays the same when considering the interval $[a, b]$. As we saw, the distance V for a specific x_i and x_{i+1} was bounded by $\max\{|f(x_i) - f^{Lin}(x_i)|, f(x_{i+1}) - f^{Lin}(x_{i+1})|\}$, so for the whole interval it will be bounded by $\max_{x_i \in \{x_1, \ldots, x_n\}} |f(x_i) - f^{Lin}(x_i)|$. This means that $|f(x) - f^{Lin}(x)|$ has the following bound:

$$|f(x) - f^{Lin}(x)| \leq \max_{x_i \in \{x_1, \ldots, x_n\}} |f(x_i) - f^{Lin}(x_i)| + K \cdot q.$$

By making use of Theorem 4, we can say that also the following holds:

$$|f(\hat{x}) - f^{Lin}(x^*)| \leq \max_{x_i \in \{x_1,\ldots,x_n\}} |f(x_i) - f^{Lin}(x_i)| + K \cdot q.$$

□

4.3.2 Determining Lipschitz constants

In the following parts, Lipschitz constants will be determined for the depart, dwell and transfer actions. The Lipschitz constant is defined as follows:

$$K = \sup_D \left| \frac{d}{dD} f(D) \right|.$$

If such a K exists, then by definition of a Lipschitz continuous function, we have:

$$\frac{|f(x) - f(y)|}{|x - y|} \leq K.$$

In the following parts, the Lipschitz constants for the possible actions will be determined. To do so, we will make use of the Leibniz integral rule, which states the following:

Theorem 6. Let $f(x,t)$ be a function such that both $f(x,t)$ and its partial derivative $\frac{d}{dx} f(x,t)$ are continuous in $a(x) \leq t \leq b(x)$. Suppose the functions $a(x)$ and $b(x)$ are continuous and have continuous derivatives. Then:

$$\frac{d}{dx} \int_{a(x)}^{b(x)} f(x,t)dt = f(x, b(x)) \frac{d}{dx} b(x) - f(x, a(x)) \frac{d}{dx} a(x) + \int_{a(x)}^{b(x)} \frac{d}{dx} f(x,t)dt.$$

And in particular:

$$\frac{d}{dx} \int_a^b f(x,t)dt = \int_a^b \frac{d}{dx} f(x,t)dt.$$

Furthermore, [29] explains the theorem can be extended to the case where we are integrating over an infinite region, resulting in the following theorem:

Theorem 7. If in addition to the conditions of Theorem 6, there is a positive function $g(x,t)$ that is integrable with respect to t and $\left|\frac{d}{dx} f(x,t)\right| \leq g(x,t)$, then the following holds:

$$\frac{d}{dx} \int_a^\infty f(x,t)dt = \int_a^\infty \frac{d}{dx} f(x,t)dt.$$

Using this theorem, the following lemmas will be proved.

Lemma 7. The Lipschitz constant of f^{Depart} equals 1.

Proof. It will first be shown that $f^{Depart}(D)$ is a decreasing function in D. Suppose $D_1 \leq D_2$.

$$\begin{aligned}
f^{Depart}(D_1) &= \mathbb{E}(\max\{0, X + Y - D_1\}) \\
&= \int_0^\infty \int_0^\infty \max\{0, x + y - D_1\} f_x(x) f_y(y) dx dy \\
&\geq \int_0^\infty \int_0^\infty \max\{0, x + y - D_2\} f_x(x) f_y(y) dx dy \\
&= \mathbb{E}(\max\{0, x + y - D_1\}) = f^{Depart}(D_2).
\end{aligned}$$

This implicates that $\frac{d}{dD}f^{Depart}(D) \leq 0$. Now let us show $\frac{d}{dD}f^{Depart}(D) \geq -1$:

$$\begin{aligned}
\frac{d}{dD}f^{Depart}(D) &= \frac{d}{dD}\mathbb{E}(\max\{0, X+Y-D\}) \\
&= \frac{d}{dD}\int_0^\infty \int_0^\infty \max\{0, x+y-D\} f_x(x) f_y(y) dx dy \\
&= \frac{d}{dD}\int_0^D \int_{D-x}^\infty (x+y-D) f_x(x) f_y(y) dy dx \\
&+ \frac{d}{dD}\int_D^\infty \int_0^\infty (x+y-D) f_x(x) f_y(y) dy dx.
\end{aligned}$$

We will now show that we can interchange the differentiation and integration operators in the above. To do so, it needs to be shown that the function $x + y - D$ satisfies the conditions of Theorem 6 and Theorem 7.

The derivative of $x + y - D$ with respect to D is continuous, because $\frac{d}{dD}(x + y - D) = -1$. Furthermore, there is an integrable function $g(x, y, D)$ such that $\left|\frac{d}{dD}(x + y - D)\right| \leq g(x, y, D)$, take for example $g(x, y, D) = 1$. The conditions for the theorem apply, so the result can be used and the order of differentiation and integration can be interchanged. We get the following:

$$\begin{aligned}
\frac{d}{dD}f^{Depart}(D) &= \frac{d}{dD}\int_0^D \int_{D-x}^\infty (x+y-D) f_x(x) f_y(y) dy dx \\
&+ \frac{d}{dD}\int_D^\infty \int_0^\infty (x+y-D) f_x(x) f_y(y) dy dx. \\
&= \int_0^D \int_{D-x}^\infty \frac{d}{dD}(x+y-D) f_x(x) f_y(y) dy dx \\
&+ \int_D^\infty \int_0^\infty \frac{d}{dD}(x+y-D) f_x(x) f_y(y) dy dx. \\
&\geq -1
\end{aligned}$$

So $-1 \leq \frac{d}{dD}f^{Depart}(D) \leq 0$. This means that the Lipschitz constant for f^{Depart} is 1. □

Lemma 8. The Lipschitz constant of f^{Dwell} equals 1.

Proof. $f^{Dwell}(D)$ is found by determining the following expectation:

$$f^{Dwell}(D) = \mathbb{E}(D + \max\{0, X+Y-D\}) = D + \mathbb{E}(\max\{0, X+Y-D\}).$$

Since $\frac{d}{dD}\mathbb{E}(\max\{0, X+Y-D\}) \geq -1$, as was shown in the previous lemma, and $\frac{d}{dD}D = 1$, it follows that $\frac{d}{dD}f^{Dwell} \geq 0$. It will now be shown that $\frac{d}{dD}f^{Dwell} \leq 1$:

$$\begin{aligned}
\frac{d}{dD}f^{Dwell}(D) &= \frac{d}{dD}\left(D + \mathbb{E}\max\{0, X+Y-D\}\right) \\
&= 1 + \frac{d}{dD}\mathbb{E}(\max\{0, X+Y-D\}).
\end{aligned}$$

We have already shown that $x + y - D$ satisfies the conditions of theorem 6, allowing us to make the following statement:

$$\begin{aligned}\frac{d}{dD}\mathbb{E}(\max\{0, X+Y-D\}) &= \frac{d}{dD}\int_0^\infty\int_0^\infty \max\{0, x+y-D\} f_x(x) f_y(y) dx dy \\ &= \int_0^D \int_{D-x}^\infty \frac{d}{dD}(x+y-D) f_x(x) f_y(y) dy dx \\ &+ \int_D^\infty \int_0^\infty \frac{d}{dD}(x+y-D) f_x(x) f_y(y) dy dx. \\ &\leq 0\end{aligned}$$

This implies that $0 \leq \frac{d}{dD} f^{Dwell}(D) \leq 1$. It follows that the Lipschitz constant of f^{Dwell} equals 1. $\square$

Lemma 9. The derivative of a transfer action is given by $1 - f_{X+Y}(D)\frac{T}{m}$, where $f_{X+Y}(D)$ is the cumulative distribution function of the stochastic variables X and Y.

Proof. The cost of a transfer action is determined by evaluating the following expectation:

$$f^{Trans}(D) = \mathbb{E}\left(D + \mathbb{1}\{X+Y > D\}\frac{T}{m}\right).$$

Here T is the timetable period and m the frequency of the train a passenger is transferring to. So $\frac{T}{m}$ is the cost of missing a transfer. To determine the Lipschitz constant, we need to determine the derivative:

$$\begin{aligned}\frac{d}{dD} f^{Trans}(D) &= \frac{d}{dD}\mathbb{E}\left(D + \mathbb{1}\{X+Y > D\}\frac{T}{m}\right) \\ &= 1 + \frac{d}{dD}P(X+Y > D)\frac{T}{m} \\ &= 1 + \frac{d}{dD}(1 - P(X+Y \leq D))\frac{T}{m} \\ &= 1 - \frac{d}{dD}F_{X+Y}(D)\frac{T}{m} \\ &= 1 - f_{X+Y}(D)\frac{T}{m}.\end{aligned}$$

Here $F_{X+Y}(D)$ is the cumulative distribution function of the sum of two random variables X and Y, and $f_{X+Y}(D)$ the distribution function of the sum of X and Y. For the Weibull distribution no closed form solution of the distribution of the sum of two Weibull random variables is known. A numerical approach could be pursued to determine the Lipschitz constant for a transfer action. The cumulative distribution function of $X+Y$ is by definition the convolution of the distributions of X and Y, $f_x(x)$ and $f_y(y)$:

$$f_{X+Y}(z) = \int_{-\infty}^\infty f_x(t) f_y(z-t) dt.$$

$\square$

By numerical integration we can find the distribution function $f_{X+Y}(z)$. This results in the following figures:

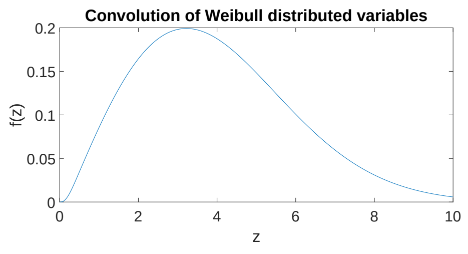


Figure 21: Convolution using $\lambda_x = 4.3$, $k_x = 2$, $\lambda_y = 0.2$, $k_y = 1.7$

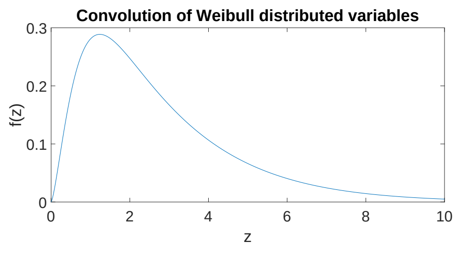


Figure 22: Convolution using $\lambda_x = 2.3$, $k_x = 1.1$, $\lambda_y = 0.6$, $k_y = 1.3$

Using these figures, the value of the Lipschitz constant can easily be determined.

4.3.3 Criteria for the integral of absolute difference

Using Theorem 4, we have a performance bound when we use a max norm to describe the quality of the approximation f^{Lin} of f. Of course other norms could also be considered. For example the integral of the difference between f^{Lin} and f. This is exactly what [30] considered. They showed conditions on the integral of the difference between f^{Lin} and f that guarantee an ϵ-optimal solution i.e. $|f(\hat{x}) - f^{Lin}(x)| \leq \epsilon$. We will consider conditions on the integral of the squared difference between f^{Lin} and f to guarantee ϵ-optimality. This is done to give the piecewise-linearization algorithm, which minimizes the squared error between functions, a criterium after which the linearization is "good enough". Whereas now a fixed amount of segments are to be chosen, the algorithm could use the amount of segments needed for ϵ-optimality. To do so, we will first consider the proof of [30] to understand the structure of the proof, after which we will apply it to the squared error case. Suppose we want to solve the following program:

$$\begin{aligned} \min_x \;\; & f(x) \\ \text{s.t.} \;\; & x \in [a, b] \end{aligned}$$

As $f : [a, b] \to \mathbb{R}$ is nonlinear, we linearize the function f into f^{Lin}. The optimal solution of f is $\hat{x}$ and the optimal solution of f^{Lin} is x^*. We are interested in the difference between the two objectives i.e. $|f(\hat{x}) - f^{Lin}(x^*)|$. [30] defines the following error function:

$$E = \int_a^b |f(x) - f^{Lin}(x)| dx.$$

Let us proof the following theorem:

Theorem 8. $\forall x \in [a, b]$ and $\epsilon > 0$ with $\epsilon \ll b - a$, the following holds:

$$f^{Lin}(x) - \frac{E}{\epsilon} \leq f(x) \leq f^{Lin}(x) + \frac{E}{\epsilon}.$$

Proof. First consider the case where $x \in [a, b)$. Then $\exists \epsilon_1 > 0$ such that $[x, x + \epsilon_1] \subseteq [a, b)$ $\forall x \in [a, b]$. This means the following:

$$\int_x^{x+\epsilon_1} |f(x) - f^{Lin}(x)| dx \leq \int_a^b |f(x) - f^{Lin}(x)| dx = E.$$

If $\epsilon_1 \ll b-a$, the first term can be approximated by $\epsilon_1|f(x) - f^{Lin}(x)|$. Basically the claim is that for every x, with $\epsilon \ll b-a$ then, $\epsilon|f(x) - f^{Lin}(x)|\leq \int_a^b|f(x) - f^{Lin}(x)|$. This is illustrated by the following figure:

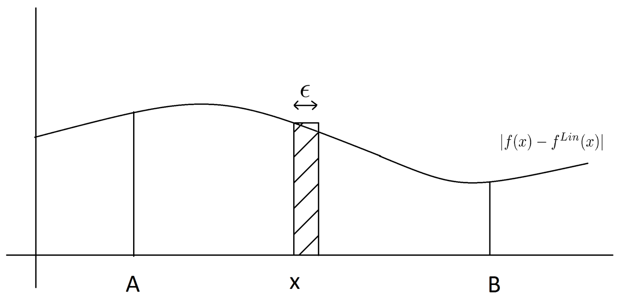


Figure 23: Illustration of ϵ argument

So $\forall x\ \exists\epsilon > 0$ such that the integral over the shaded area is smaller than the total area under the function from a to b. When using this result, this gives $|f(x) - f^{Lin}(x)|\leq \frac{E}{\epsilon_1}$. Now suppose $x = b$. Then $\exists\epsilon_2$ such that $[x-\epsilon_2, x] \subseteq [a, b]$. This means:

$$\int_{x-\epsilon_2}^{x} |f(x) - f^{Lin}(x)|dx \leq \int_a^b |f(x) - f^{Lin}(x)|dx = E$$

In the same way, the first term can be approximated by $\epsilon_2|f(x) - f^{Lin}(x)|$, which gives $|f(x) - f^{Lin}(x)|\leq \frac{E}{\epsilon_2}$.

Concluding from the above, if we take $\epsilon \leq \max\{\epsilon_1, \epsilon_2\}$, then $f^{Lin}(x) - \frac{E}{\epsilon} \leq f(x) \leq f^{Lin}(x) + \frac{E}{\epsilon}$ $\forall x$. □

Using the above, we can proof the following theorem:

Theorem 9. If the conditions of the above theorem hold and if $E \leq \epsilon^2$, then $|f(\hat{x}) - f^{Lin}(x^*)|\leq \epsilon$.

Proof. We showed $f^{Lin}(x) - \frac{E}{\epsilon} \leq f(x) \leq f^{Lin}(x) + \frac{E}{\epsilon}$ $\forall x$, which is equivalent with $|f(x) - f^{Lin}(x)|\leq \frac{E}{\epsilon}$. If we take $E \leq \epsilon^2$, then $|f(x) - f^{Lin}(x)|\leq \epsilon$. Using Theorem 4 it follows that $|f(\hat{x}) - f^{Lin}(x^*)|\leq \epsilon$. □

We will now provide similar conditions when the integral of the squared difference is used as the error function E.

4.3.4 Criteria for integral of squared difference

Let the error function be defined as follows:

$$E = \int_a^b [f(x) - f^{Lin}(x)]^2 dx.$$

Now we can proof the following theorem:

Theorem 10. $\forall x \in [a, b]$ and $\epsilon > 0$ with $\epsilon \ll b - a$, the following holds:

$$f^{Lin}(x) - \sqrt{\frac{E}{\epsilon}} \leq f(x) \leq f^{Lin}(x) + \sqrt{\frac{E}{\epsilon}}.$$

Proof. $\forall x \in [a, b)$ $\exists \epsilon_1 > 0$ such that $[x, x + \epsilon_1] \subseteq [a, b)$. Which means:

$$\int_x^{x+\epsilon_1} \left[f(x) - f^{Lin}(x)\right]^2 dx \leq \int_a^b \left[f(x) - f^{Lin}(x)\right]^2 dx = E.$$

Again if ϵ_1 small enough, we can conclude $\left[f(x) - f^{Lin}(x)\right]^2 \leq \frac{E}{\epsilon_1}$.

Now take $x = b$, then $\exists \epsilon_2 > 0$ such that $[x - \epsilon_2, x] \subseteq [a, b]$. This means:

$$\int_{x-\epsilon_2}^{x} \left[f(x) - f^{Lin}(x)\right]^2 dx \leq \int_a^b \left[f(x) - f^{Lin}(x)\right]^2 dx = E.$$

From the above, we can say $\left[f(x) - f^{Lin}(x)\right]^2 \leq \frac{E}{\epsilon}$, or:

$$f^{Lin}(x) - \sqrt{\frac{E}{\epsilon}} \leq f(x) \leq f^{Lin}(x) + \sqrt{\frac{E}{\epsilon}}, \ \forall x \in [a, b].$$

□

In a similar way as the previous section, we can proof the following:

Theorem 11. If $E \leq \epsilon^3$, with $E = \int_a^b [f(x) - f^{Lin}(x)]^2 dx$, then $|f(\hat{x}) - f^{Lin}(x^*)| \leq \epsilon$.

Proof. From the previous theorem it can be concluded that if $E = \int_a^b [f(x) - f^{Lin}(x)]^2 dx$, then $|f(x) - f^{Lin}(x)| \leq \sqrt{\frac{E}{\epsilon}}$. If we take $E \leq \epsilon^3$, then $|f(x) - f^{Lin}(x)| \leq \epsilon$. Using Theorem 4, we can conclude that $|f(\hat{x}) - f^{Lin}(x^*)| \leq \epsilon$. □

We will now make the connection with the piecewise linearization algorithm. The piecewise-linearization algorithm minimizes the following error: $\sum_{i=1}^N [f(x_i) - f^{Lin}(x_i)]^2$, which is equivalent with minimizing $\frac{1}{N} \sum_{i=1}^N [f(x_i) - f^{Lin}(x_i)]^2$. If f is a smooth enough function then the sum approximates the integral of the difference between the two functions well. So:

$$\frac{1}{N} \sum_{i=1}^N [f(x_i) - f^{Lin}(x_i)]^2 \approx \int_a^b [f(x) - f^{Lin}(x)]^2 dx.$$

The criterium for the squared error was $E \leq \epsilon^3$. It can be concluded that if $\sum_{i=1}^N [f(x_i) - f^{Lin}(x_i)]^2 \lesssim \epsilon^3$, we can guarantee ϵ-optimality. This criterium could be used during piecewise linearization.

4.3.5 Multivariate case

The goal function derived in Chapter 3.3 is a multivariate expression. Also, the goal function is separable, meaning it can be written as a sum of univariate functions. Theorem 4 showed that to conclude $|f(\hat{x}) - f^{Lin}(x^*)| \leq \epsilon$, it is sufficient to show $|f(x) - f^{Lin}(x)| \leq \epsilon$ holds. We can make a similar argument when considering functions which are multivariate, but also separable. A function $f(x)$, $f:\mathbb{R}^n \to \mathbb{R}$ is separable if it can be written as follows:

$$f(x) = \sum_{i=1}^{n} f_i(x_i),$$

for some $f_i(x_i)$. Consider two separable functions $f(x)$ and its piecewise linear approximation $f^{Lin}(x)$. We can bound the maximum difference as follows:

$$|f(x) - f^{Lin}(x)| = \left| \sum_{i=1}^{n} f_i(x_i) - \sum_{i=1}^{n} f_i^{Lin}(x_i) \right| \leq \sum_{i=1}^{n} \left| f_i(x_i) - f_i^{Lin}(x_i) \right| .$$

So for a multivariate separable function, if $\sum_{i=1}^{n} \left| f_i(x_i) - f_i^{Lin}(x_i) \right| \leq \epsilon$, then also $|f(\hat{x}) - f^{Lin}(x^*)| \leq \epsilon$. This result will be used to analyze the quality of approximations in the results section.

5 Model without independence assumption

The following chapter considers an assumption that has implicitly been made in the previous chapters and also in the literature that tries to minimize expected travel time. The simplification will be explained and an approach will be developed to test if the solution of the model which uses this simplification is close to the model which does not make this simplification. The model described here is a more realistic way of minimizing expected passenger travel time. The basic idea is similar to [20], though applying to expected passenger travel time has not been developed before in literature.

In the previous sections we managed to get an expression for the expected travel time and also ways to formulate the problem as an MILP. However, we implicitly did make some assumptions when constructing the model. We basically splitted a travel route into independent actions, with each action having a certain expected cost. Then the total travel time was just the sum of the travel time along these specific actions. This is however not exactly what happens in reality. When determining the expressions for the actions, we assumed the only delay came from the ride activity X and the dwell or transfer activity Y, so the delay that has occurred since the last buffer. This does not take onto account the delay that is still present in the system because of delays from previous actions, which were not entirely absorbed by previous buffers. Let the model developed so far be called the independent model (IM). The independent model states that a train which is on its n^{th} action has a delay of $\max\{0, X_n + Y_n - D_n\}$ on this action. The total delay TD of the train up to action n is as follows:

$$TD^n_{IM} = \sum_{i=1}^{n} \max\{0, X_n + Y_n - D_n\}.$$

However, for example [20], where the Stochastic Optimization Model used at NS to determine buffer sizes is based on, a different relation is used. The total delay up to action n is dependent on the total delay acquired up to action $n-1$ and should be added to the delays X and Y. Let this model be called the dependent model (DM). The dependent model gives the following expression for the delay up to action n:

$$TD^n_{DM} = \max\{0, TD^{n-1}_{DM} + X_n + Y_n - D_n\}.$$

Note that this is a recursive formula, which if expanded would look like this:

$$\begin{aligned} TD^n_{DM} &= \max\{0, TD^{n-1}_{DM} + X_n + Y_n - D_n\} \\ &= \max\{0, X_n + Y_n - D_n + \max\{0, X_{n-1} + Y_{n-1} - D_{n-1} + ... + \max\{0, X_1 + Y_1 - D_1\}...\}\}. \end{aligned}$$

An example for a specific instance will be given to show the difference of the relations defined in the independent model and the dependent model.

Example 2. Consider a train line consisting of 2 pairs of ride and dwell actions. Let the delay of the first ride action be X_1, the first dwell action Y_1 and the first buffer D_1. X_2, Y_2 and D_2 are defined similarly for the second pair. Consider the following realization of the random delays and the chosen buffers.

$$\begin{aligned} x_1 &= 2 & x_2 &= 0.5 \\ y_1 &= 1.5 & y_2 &= 1 \\ D_1 &= 3 & D_2 &= 2 \end{aligned}$$

Now the independent model gives the following total delay:

$$\begin{aligned} TD^2_{IM} &= \max\{0, x_1 + y_1 - D_1\} + \max\{0, x_2 + y_2 - D_2\} \\ &= 0.5 + 0 = 0.5. \end{aligned}$$

The dependent model would give the following:

$$\begin{aligned} TD^2_{DM} &= \max\{0, x_2 + y_2 - D_2 + \max\{0, x_1 + y_1 - D_1\}\} \\ &= \max\{0, -0.5 + \max\{0, 0.5\}\} \\ &= 0. \end{aligned}$$

The essential difference is the following. For both models a delay of 0.5 is achieved after the first ride dwell pair. However for the dependent model, the delay gathered in the first pair is compensated by the buffer of the second pair. This is possible because not all of the second buffer is used to absorb the delays of the second ride dwell pair. The IM does not allow this to happen and as a consequence has a delay of 0.5. In real life the behavior of the DM is observed.

Though the expression for the IM might seem quite different from the expression developed for the dependent model, there are quite some similarities. Basically the independent model assumes that the pre-delay, the delay acquired from previous actions, which were not absorbed by buffers, is non existent or negligible. The justification for this assumption in the literature is that the whole goal of placing these buffers is to avoid delays. So this pre-delay, if it is present, will not be of such a size that it will have a significant influence. This assumption however has nowhere been justified.

We can give an expectation of what the difference between the two models will be. There are two main differences to be expected:

1. *IM will have lower buffer values.* Consider a passenger making a depart action at some station n. The independent model assumes a delay of $X_n + Y_n$ at station n before the buffer. The dependent model assumes a delay of $X_n + Y_n + TD^{n-1}_{DM}$. Although the value of TD^{n-1}_{DM} might be zero, it can also have a positive value. As the dependent model assumes delays at least as big (and possibly bigger) then the independent model, it is to be expected that the buffers of the dependent model are at least as big (and possibly bigger) than the independent model.

2. *IM will have a bigger objective function value.* The independent model assumes that the total delay is the sum of delays at the individual actions. However, in reality it might be that delays acquired at the beginning of the route can be absorbed later on. As Example 2 showed, this is the case when at the beginning a big delay occurs, which is not entirely absorbed by the corresponding buffer. Suppose that at the next action, a small delay occurs, for which not all the buffer is needed to absorb this delay. Then this buffer will (at least) partially absorb the delay that remained from the beginning. This kind of absorption is exactly what the nested max function of the DM describes. The IM does not allow absorption of delays later on, so it is expected that the goal function of the IM will be bigger than the DM, when using the exact same solution.

So as we have seen the DM will describe reality more accurately than the independent model. So why did we even put effort into this IM? It turns out determining a goal function for the DM is quite hard to do analytically. To determine the expected travel time, one of the things necessary would be an expression for the expected delay $\mathbb{E}\left(TD^n_{DM}\right)$. This expression can be written as:

$$\begin{aligned} \mathbb{E}\left(TD^n_{DM}\right) &= \mathbb{E}\left(\max\{0, TD^{n-1}_{DM} + X_n + Y_n - D_n\}\right) \\ &= \int_\Omega \max\{0, x_n + y_n - D_n + \max\{0, x_{n-1} + ... + \max\{0, x_1 + y_1 - D_1\}...\}\} f_x(x) f_y(y) dx dy. \end{aligned}$$

Note that the integral is an $2n$ dimensional integral and that $f_x(x)$ is the distribution function for the vector $x \in \mathbb{R}^n$, defined as $f_x(x) = \prod_{i=1}^n f_i(x_i)$ and $dx = \prod_{i=1}^n dx_i$. A similar relation holds for

$f_y(y)$ and dy. We have not been able to find an expression for this integral. An analytical goal function for the DM will probably not be possible. This is one of the main reasons why the IM is developed.

Ideally we would want to check if the solution of the IM is close to the solution of the DM. As was explained in Chapter 2.4, it is possible to get an arbitrarily close approximation of the solution of the problem using for example the Sample Average Approximation technique. Using this, we can compare the two models and decide whether or not the IM is a good enough approximation of the actual problem, the DM. In order to do so, we need to explicitly determine the goal function of the DM, but in terms of the stochastic variable travel time instead of the expectation of the travel time, which we have seen is not possible. The goal function in terms of the stochastic variable travel time will be needed to minimize the expected value of this goal function by SAA. At this point we just need an expression for the stochastic travel time. Note that in theory we can get an arbitrarily close solution, but as will be seen, this will be computationally very difficult. It is very unlikely that the model to be explained will be applicable to large scale networks.

5.1 Goal function

We want to determine an expression for the travel time, using the relation TD^n_{DM} for the delay of a train. What remains the same with respect to the IM, is that we need expressions for departing, riding and transferring actions. However, they will be defined in a different way. One assumption that we make is that a train leaves on time at its very first station, after a turn around action, so at the beginning of the line. As usually quite big supplements are planned at turn around stations, this is a reasonable assumption.

To get expressions for the travel time we need to make a slight generalization to the relation TD^n_{DM}. This is because TD^n_{DM} describes the total delay when starting from the very first station, station 1. For our model we need slightly general expressions, we need to have expressions for the delay starting from station i until station j, with $i < j$. The total delay from station i until station j will be represented as $TD^{i,j}_{DM}$, which is defined as follows:

$$TD^{i,j}_{DM} = \max\{0, TD^{i,j-1}_{DM} + X_j + Y_j - D_j\}.$$

Here $TD^{ii}_{DM} = 0$. This relation will be used in the following sections.

5.1.1 Departing cost

Suppose that a passenger wants to travel from station i to a station j. The passenger enters the train at station i. However, it could be the case that the train arriving at station i is delayed. The amount of time the train is delayed is the additional travel time for our passenger. The amount of time a train is delayed, is the delay the train has gathered starting from station 1 until station i. This is given by the following expression:

$$\begin{aligned} f^{Depart}_{1,i} &= TD^{1,i}_{DM} \\ &= \max\{0, X_i + Y_i - D_i + \max\{0, X_{i-1} + ... + \max\{0, X_1 + Y_1 - D_1\}...\}\}. \end{aligned}$$

5.1.2 Riding cost

Now that we have an expression for the waiting time as a consequence of a train arriving late, we will determine the travel time of actually traveling from station i to station j. The total delay when traveling from station i to station j was defined as $TD_{DM}^{i,j}$. On top of the delay, we for sure at every station need to wait for a time equal to the buffer. $TD_{DM}^{i,j}$ is just the expression for delay on top of the buffer waiting time, so the total travel time will be the sum of the two:

$$\begin{aligned} f_{i,j}^{Dwell} &= \sum_{n=i}^{j} D_n + TD_{DM}^{i,j} \\ &= \sum_{n=i}^{j} D_n + \max\{0, X_j + Y_j - D_j + \max\{0, X_{j-1} + ... + \max\{0, X_i + Y_i - D_i\}...\}\}. \end{aligned}$$

5.1.3 Transfer cost

The last cost to be determined is the transfer cost, this is the cost as a consequence of missing a transfer. A transfer is missed if, similarly to the IM, the delay up to that point is bigger than the buffer. For a transfer action, we have incorporated a buffer D_{Trans}. This buffer time is added to the travel time no matter what. On top of that we possibly have to wait for the next train. Let T be the timetable period and m be the frequency of the train we are transferring to. Then the cost of missing a transfer is $\frac{T}{m}$. Suppose we want to make a transfer at a station i. The total delay that the train has acquired is the following:

$$\begin{aligned} Q_i^{Trans} &= \max\{0, TD_{DM}^{1,i-1} + X_i + Y_{Trans}\} \\ &= \max\{0, X_i + Y_{Trans} + \max\{0, X_{i-1} + Y_{i-1} + ... + \max\{0, X_1 + Y_1 - D_1\}\}\}. \end{aligned}$$

Note that here we cannot use Y_i, as that is a dwell action at station i, which the passenger does not use. This is why we cannot simply take $TD_{DM}^{1,i}$ but need to use a slightly adapted definition, involving the delay a passenger experiences when traveling at the station from one train to another, Y_{Trans}. Now the transfer cost is the following:

$$f_i^{Transfer} = D_{Trans} + \mathbb{1}\{Q_i^{Trans} > D_{Trans}\}\frac{T}{m}.$$

Total travel time of a passenger following a certain route is now just the sum of combinations of these actions.

5.2 Applying to goal function

We will now apply Sample Average Approximation (SAA) to the goal function determined for the DM, as SAA seems to be the most promising technique. Lets consider a passenger taking a route from station i in network 1, to station j in network 2, having a transfer at station k. First this network consisting of a total of two lines will be explained, after which at the end of this chapter a generalization to an arbitrary sized network will be made. The total delay of this passenger is the following stochastic variable:

$$TD = f_{1,i}^{Depart} + f_{i,k}^{Dwell} + f_k^{Transfer} + \hat{f}_{k,j}^{Dwell}.$$

Here $\hat{f}_{k,j}^{Dwell}$ corresponds to the second network and $f_{i,k}^{Ride}$ to the first network. To use SAA, we need to take samples for the random variables. Suppose we take N samples of the stochastic variables, according to their corresponding distribution. Let $f_{1,i}^{Depart}(l)$ be the l^{th} realization of $f_{1,i}^{Depart}$, i.e.:

$$\begin{aligned} f_{1,i}^{Depart}(l) &= TD_{DM}^{1,i}(l) \\ &= \max\{0, x_i^l + y_i^l - D_i + \max\{0, x_{i-1}^l + ... + \max\{0, x_1^l + y_1^l - D_1\}...\}\}. \end{aligned}$$

Here x_i^l is the l^{th} realization of X_i. Similarly we can get expressions for the other actions. The expressions for the actions are entirely the same as in the previous chapter, however now we have the dependency on l^{th} realization of a stochastic variable instead of the stochastic variable itself.

$$\begin{aligned}
f_{i,k}^{Dwell}(l) &= \sum_{n=i}^{k} D_n + TD_{DM}^{i,k}(l) \\
&= \sum_{n=i}^{k} D_n + \max\{0, x_k^l + y_k^l - D_k + \max\{0, x_{k-1}^l + ... + \max\{0, x_i^l + y_i^l - D_i\}...\}\} \\
f_k^{Transfer}(l) &= D^{Trans} + \mathbb{1}\{Q_k^{Trans}(l) > D_{Trans}\}\frac{T}{m} \\
Q_k^{Trans}(l) &= \max\{0, TD_{DM}^{1,k-1}(l) + x_k^l + y_{Trans}(l)\} \\
&= \max\{0, x_k^l + y_{Trans}^l + \max\{0, x_{k-1}^l + y_{k-1}^l + ... + \max\{0, x_1^l + y_1^l - D_1\}\}\} \\
\hat{f}_{k,j}^{Dwell}(l) &= \sum_{n=k}^{j} \hat{D}_n + \hat{T}D_{DM}^{k,j}(l) \\
&= \sum_{n=k}^{j} \hat{D}_n + \max\{0, \hat{x}_j^l + \hat{y}_j^l - \hat{D}_j + \max\{0, \hat{x}_{j-1}^l + ... + \max\{0, \hat{x}_k^l + \hat{y}_k^l - \hat{D}_k\}...\}\}.
\end{aligned}$$

Here, if a variable has a hat, it means that we take the variable in the second train network, instead of the first one. Now the SAA approximation of the goal function, if we take N samples, is the following:

$$TD^* = \frac{1}{N}\sum_{l=1}^{N}\left[f_{1,i}^{Depart}(l) + f_{i,k}^{Dwell}(l) + f_k^{Transfer}(l) + \hat{f}_{k,j}^{Dwell}(l)\right]. \qquad (2)$$

This is the goal function that is to be optimized in the SAA scheme. TD^* is an approximation of TD, which was defined as an integral which we could not analytically determine. However, because we know the SAA technique converges to the optimal solution, we know that TD^* will also converge to the optimal solution, as the sample size grows to infinity.

5.3 Implementing as a linear program

From the previous paragraph, we now have a goal function to be optimized. However, it is not yet in such a way that in can be written as a linear program. This is caused by two reasons. First we have nested max functions in the goal function. Furthermore we also have an indicator function in the goal function. In the following we will transform the above goal function to an actual linear goal function, by adding constraints in a certain way.

5.3.1 Linearizing a max function

Suppose we have to minimize a max function, for example the very simple case:

$$\begin{aligned}
\min_x \ & \max\{0, x\} \\
\text{s.t. } & Ax = b
\end{aligned}$$

We can make an equivalent linear program of the following form:

$$\begin{aligned} \min_x \ & t \\ \text{s.t. } & Ax = b \\ & t \geq x \\ & t \geq 0 \end{aligned}$$

This is equivalent as we are considering a minimization problem. If $x < 0$, t will be chosen to be 0. If $x > 0$, t will be chosen to be x. This is exactly the same as a max function would do. Let us now take a look at how a nested max function should be linearized. Consider the following example:

Example 3.

$$\min_D \ \max\{0, x_1 + y_1 - D_1\}.$$

By first linearizing $\max\{0, x_1 + y_1 - D_1\}$, we get the following equivalent program:

$$\begin{aligned} \min_D \ & \max\{0, x_2 + y_2 - D_2 + t_1\} \\ \text{s.t. } & t_1 \geq 0 \\ & t_1 \geq x_1 + y_1 - D_1 \end{aligned}$$

Using the same steps as before, this leads to the following linear program:

$$\begin{aligned} \min_D \ & t_2 \\ \text{s.t. } & t_1 \geq 0 \\ & t_1 \geq x_1 + y_1 - D_1 \\ & t_2 \geq 0 \\ & t_2 \geq x_2 + y_2 - D_2 + t_1 \end{aligned}$$

Let us now take a look at how a nested function of general length:

$$\min_D \max\{0, x_j + y_j - D_j + \max\{0, x_{j-1} + ... + \max\{0, x_i + y_i - D_i\}...\}\}.$$

This expression indicates the delay a passenger experiences if he boards the train at station i and leaves the train at station j. Using the same approach used in the example, would give the following linearization:

$$\begin{aligned} \min \ & t_{ij} & \\ \text{s.t. } & t_{i,q} \geq 0 & q = i+1...j \\ & t_{i,q} \geq x_q + y_q - D_q - t_{i,q-1} & q = i+1...j \end{aligned}$$

Now for example $f_{1,i}^{Depart}(l)$ will be linearized as follows:

$$\begin{aligned} \min \ & t_{1,i}^l & \\ \text{s.t. } & t_{p,q}^l \geq 0 & p = 1...i,\ q = p+1...i,\ l = 1...N \\ & t_{p,q} \geq x_q^l + y_q^l - D_q - t_{p,q-1}^l & p = 1...i,\ q = p+1...i,\ l = 1...N \end{aligned}$$

5.3.2 Linearizing an indicator function

Now we will consider how to linearize an indicator function. Suppose we have a problem which has some indicator function $\mathbb{1}\{A > x\}$ in the goal function. We can replace this indicator variable by a binary variable when introducing some additional constraints. This results in the following:

$$\begin{aligned} \min\ & z \\ \text{s.t.}\ & z \in \{0,1\} \\ & \frac{A-x}{M} \leq z \leq 1 + \frac{A-x}{M} \end{aligned}$$

Here M is some big number, or if A has an upper bound, it can be chosen to be slightly bigger then this upper bound. Why is this an equivalent representation? Suppose $A > x$. Then the indicator value should have value 1. If $A > x$, this implies that $\epsilon = \frac{A-x}{M} > 0$ and $1 = \epsilon = 1 + \frac{A-x}{M} > 1$. This results in the following equality: $\epsilon < z < 1+\epsilon$ with $0 < \epsilon < 1$. As $z \in \{0,1\}$, z is forced to take value 1. Now suppose that $A < x$. This means that the indicator value should have value 0, and thus that our variable z should also take value 0. If $A < x$, we have $\epsilon = \frac{A-x}{M} < 0$ and $1+\epsilon = 1 + \frac{A-x}{M} < 1$. We get the following inequality: $-\epsilon < z < 1-\epsilon$ with $0 < \epsilon < 1$. This forces z to take value 0. Last suppose $A = x$. Then we get the inequality $0 \leq z \leq 1$, so z can take either value 0 or 1. However, since we are considering a minimization problem, z will take value 0, which is exactly what needs to happen.

Now concerning the indicator variable in our goal function, we have an indicator expression $\mathbb{1}\{Q_k^{Trans}(l) > D_{Trans}\}\frac{T}{m}$ in our goal function, where $Q_k^{Trans}(l)$ is defined as follows:

$$Q_k^{Trans}(l) = \max\{0, x_k^l + y_{Trans}^l + \max\{0, x_{k-1}^l + y_{k-1}^l + ... + \max\{0, x_1^l + y_1^l - D_1\}\}\}.$$

Now we replace the function $\mathbb{1}\{Q_k^{Trans}(l) > D_{Trans}\}\frac{T}{m}$ with:

$$\begin{aligned} & z^l \frac{T}{m} \\ \text{s.t.}\ & \frac{Q_k^{Trans}(l) - D_{Trans}}{M} \leq z^l \leq 1 + \frac{Q_k^{Trans}(l) - D_{Trans}}{M} && \forall l = 1...N \\ & z^l \in \{0,1\} && \forall l = 1...N \end{aligned}$$

5.3.3 Goal function

We have now seen techniques to linearize max functions and indicator functions. By applying these techniques to equation 2 we get the following linear program:

$$\begin{aligned}
\min\ & \frac{1}{N}\sum_{l=1}^{N}\left[t^l_{1,i}+t^l_{i,k}+z^l\frac{T}{m}+\hat{t}^l_{k,j}\right]+\sum_{p=i}^{k}D_p+\sum_{p=k}^{j}\hat{D}_p+\quad D_{Trans} & \\
\text{s.t. } & t^l_{p,q}\geq 0 & l=1,...,N,p=1,...,k-1,q=p+1,...,k\\
& t^l_{p,q}\geq x^l_q+y^l_q-D_q+t^l_{p,q-1} & l=1,...,N,p=1,...,k-1,q=p+1,...,k\\
& \hat{t}^l_{p,q}\geq 0 & l=1,...,N,p=1,...,j-1,q=p+1,...,j\\
& \hat{t}^l_{p,q}\geq \hat{x}^l_q+\hat{y}^l_q-\hat{D}_q+\hat{t}^l_{p,q-1} & l=1,...,N,p=1,...,j-1,q=p+1,...,j\\
& \frac{Q^{Trans}_k(l)-D_{Trans}}{M}\leq z^l\leq 1+\frac{Q^{Trans}_k(l)-D_{Trans}}{M} & l=1,...,N\\
& z^l\in\{0,1\} & l=1,...,N\\
& Q^{Trans}_k(l)\geq 0 & l=1,...,N\\
& Q^{Trans}_k(l)\geq x^l_k+y^l_{Trans}+t^l_{1,k-1} & l=1,...,N
\end{aligned}$$

Note that the matrix $t^l_{p,q}$ for fixed l is a strict upper triangular matrix, where the value of $t^l_{p,q}$ states the delay when traveling from station p to station q with $p<q$ in the l^{th} realization. The above expression is the travel time of a single passenger traveling from station i in network 1 to station j in network 2, by making a transfer at station k. The goal function to be used in optimization sums over all different OD pairs, multiplied with their respective amount of passengers using this route, just like the IM. This results in the following goal function:

$$\sum_i\sum_j w_{ij}\left[\frac{1}{N}\sum_{l=1}^{N}\left[t^l_{1,i}+t^l_{i,k}+z^l\frac{T}{m}+\hat{t}^l_{k,j}\right]+\sum_{p=i}^{k}D_p+\sum_{p=k}^{j}\hat{D}_p+D_{Trans}\right].$$

Here w_{ij} is the amount of passengers traveling from route i to j. Note that the definition of the first and second network, indicated by f having a hat or not, is dependent on the passenger and should be adapted accordingly. Of course it is also possible that a passenger does not need a transfer to get from i to j. For these passengers, the following goal function will be used:

$$\sum_i\sum_j w_{ij}\left(\left[\frac{1}{N}\sum_{l=1}^{N}t^l_{1,i}+t^l_{i,j}\right]+\sum_{p=i}^{j}D_p\right).$$

5.3.4 General sized networks

The above expressions were developed for a network consisting of two trains lines with exactly 1 transfer station. However a more general formulation of the problem is useful. Let *Route* be the total possible route set of a network with an arbitrary number of train lines, where $p\in$ *Route* is a specific route. A route p consists of a list of stations to be traveled through, so $p=\{p_1,p_2,...,p_{n-1},p_n\}$, where p_1 is the starting station, $p_2,...,p_{n-1}$ are transferring stations and p_n is the end station. We introduce the notation $t^l_{i,j}(q)$. This indicates the travel time from station i to j in the q^{th} train network for sample number l. For the route $p\in$ *Route* we can determine for every station p_i which network it is in. Let the index i of p_i be the number of the network it is in. Then the transfer stations, p_i with $i = 2, ..., n-1$ are all in network i, but also in network $i-1$. The station p_1 is of course in network 1 and the station p_n is in network $n-1$. Using this notation we get the following expression for travel time for a general network:

$$\sum_{p \in Route} w_p \left[\frac{1}{N} \sum_{l=1}^{N} \left(t^l_{1,p_1}(1) + \sum_{q=1}^{n-1} t^l_{p_q,p_{q+1}}(q) + \sum_{q=2}^{n-1} z^l_{p_q}(q-1) \frac{T}{m} \right) \right.$$
$$\left. + \sum_{q=2}^{n-1} D_{Trans}(q) + \sum_{q=1}^{n-1} \sum_{i=p_q}^{p_{q+1}} D_i(q) \right].$$

The expression explained before for a network of size 2 with 1 transfer station is a special case of the above formula. Note that for every possible transfer between lines, N binary variables are needed to model these transfers. This makes it numerically significantly harder to apply the model to large scale networks, as N, the amount of samples considered, should ideally be quite big to get a higher accuracy approximation of the problem.

5.4 Heuristic

As explained before, the solution of the DM will most likely have bigger buffer values than the IM. This is because a boarding passenger at station n in the DM experiences a delay of $\max\{0, X_n + Y_n - D_n + PD\}$, where $PD \geq 0$ is the delay that is still left in the system after the last buffer. However, for the IM a boarding passenger experiences a delay of $\max\{0, X_n + Y_n - D_n\}$. As a consequence, the buffer in the DM should be bigger to compensate for the predelay PD, which is not taken into account in the IM. We can develop a heuristic to compensate for the predelay not taken into account. Basically the delays for independent model are too small. We could adjust the delay parameters such that in expectation the total delay equals the delay when also PD is taken into account. More specifically we could do the following:

1. *Compute solution using IM.* First we compute a solution using the original parameters for the independent model. We store the solution.
2. *Plug solution into DM.* We want to know what the predelay is that we do not take into account using the IM. To determine this predelay, we plug the solution found at the first stem into the DM. If we do so, we can determine the average value of the predelay PD. As the solution or timetable is fixed, this step is basically a Monte-Carlo simulation to determine PD for the solution found in step 1.
3. *Update parameters in IM.* We want to adjust the parameters in the IM such that we somehow take into account the predelay PD that we have found in the previous step. For every stochastic variable X_i and Y_i in the original problem, we define two new variables $\hat{X}_i$ and $\hat{Y}_i$. We will now choose the parameters of $\hat{X}_i$ and $\hat{Y}_i$ such that the expectation of $\hat{X}_i + \hat{Y}_i$ equals the expectation of $X_i + Y_i + PD$. We choose a factor b and define $\hat{\lambda}_{x_i} = b\lambda_{x_i}$ and $\hat{\lambda}_{y_i} = b\lambda_{y_i}$ such that $\mathbb{E}\left(\hat{X}_i + \hat{Y}_i\right) = \frac{1}{b\lambda_{x_i}} + \frac{1}{b\lambda_{y_i}} = \frac{1}{\lambda_{x_i}} + \frac{1}{\lambda_{y_i}} + PD = \mathbb{E}(X_i + Y_i + PD)$. Using these parameters a new solution will be found using the IM.
4. *Plug solution into DM.* We have now found a solution of the IM using the updated parameters. In theory, the solution should now more resemble the solution of the actual solution of the DM, because we have now compensated for the predelay, which was not taken into account before. If we plug in the solution to the DM, we can see if we are closer to the actual optimal solution.

This heuristic could make the solution of the IM close to the solution of the DM, since the experienced delays in the IM are by this heuristic more similar to the delays experienced by the DM.

6 Results

In this section the models explained in the previous chapters will be applied to train networks. The first part are the results of applying the independent model developed in Chapter 3 and 4 to the Dutch intercity network. The second part are the results of checking the independence assumption, using the model developed in Chapter 5, on a small network.

In order to apply the models to the train networks, we need to know the amount of passengers following specific routes. NS has developed a model distributing passengers over the network in the case that the timetable is not known yet and also when the timetable is known. The model is called "TRANS toedeler" and makes a distribution of passengers over the network depending on the OD-Matrix, amount of transfers needed for a certain route, financial cost of a route etc. This will be used as input for our model. Of course the results from the TRANS toedeler are estimates of passenger numbers, especially in the case no timetable is known yet.

The experiments were run using CPLEX 12.6.2 on an Intel Xeon CPU E5-1650 3.6 GHz processor with 16 GB RAM.

6.1 Results for the independent model

The model will be applied to the Dutch intercity network, pictured below.

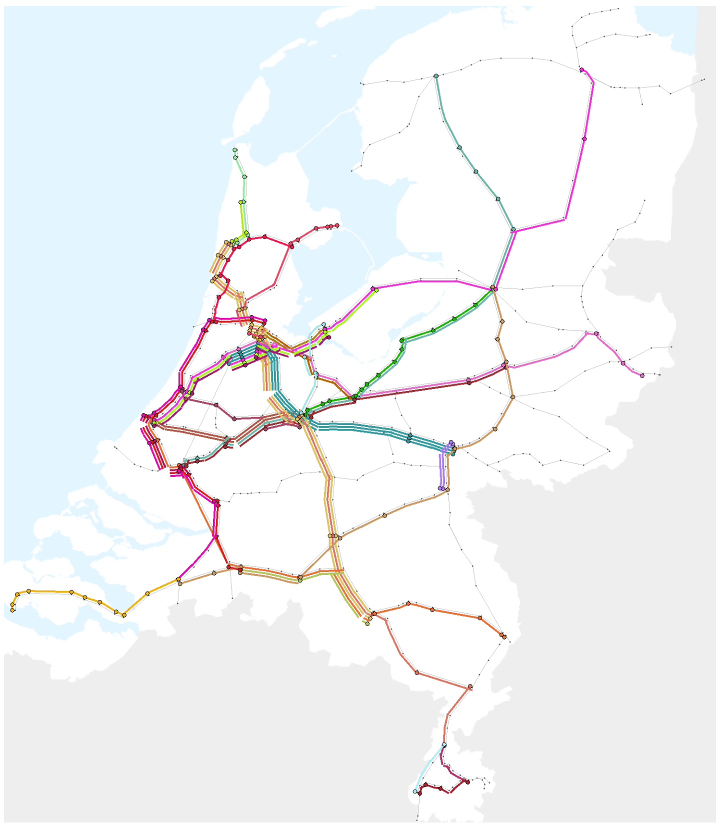


Figure 24: The Dutch intercity network

The network consists of 34 lines where every line consists of a forward connection and a backward connection. For every edge in the graph formulation of the network, a technical minimum time is known. We will only consider delays that are exponentially distributed and not Weibull distributed, to save us the effort of numerically integrating the goal function prior to linearization. For such a big network, this would take a lot of time. The distribution parameters used are dependent on the technical minimum activity time. To set the parameters of the delay distribution, we assume the expected delay is a certain percentage of the technical minimum time, as longer routes tend to have bigger delays. Taking a certain percentage of the technical minimum time is a standard approach in literature. Let this delay factor by a and suppose we have a minimum activity time of m_e on edge e. The expected delay for this edge will be $m_e a$. Since the expectation of an exponential distributed variable is given by $\frac{1}{\lambda_e}$ for an edge e, the distribution parameter will be determined as $\lambda_e = \frac{1}{m_e a}$. Unless stated otherwise, the delay factor used is 0.09. When we consider all possible transfer generated by the TRANS toedeler, we get 561 transfers. For a transfer edge we assume a minimum activity time of 2 minutes and a delay parameter of 2. Furthermore, there are 710 headway constraints. Since every intercity train has a frequency of exactly 2 in a cycle period of 60 minutes and frequencies of the same train should be spreaded perfectly i.e. 30 minutes, the problem is equivalent with considering a network with a cycle time of 30 minutes, where we only consider the first frequency. Unless stated otherwise, the piecewise linearization of the goal function will consist of 5 linear segments. Later on in the report the effect of increasing and decreasing the amount of segments on computation time will be shown.

First we will consider if, and for what cases, the newly developed model for the spreading of trains is better than the model developed in [5]. The TRANS toedeler determines if there are multiple possible routes connecting an origin and destination. We will use this as input for the model and add these pairs to the spread objective function. The amount of spread pairs generated by the TRANS toedeler is stated in the following table:

	#Instances
2 pairs	93
3 pairs	46
4 pairs	25
5 pairs	5
6 pairs	11
7 pairs	1
8 pairs	0

Table 2: Total spreading set showing the amount of occurrences of a specific amount of trains to be spreaded

Since we are considering a timetable with a cycle time of 30 minutes, it can be debated how much for example spreading 6 trains pairs adds in comparison to spreading 5 train pairs. Especially since computation time is expected to rise severely as the number of pairs increases, it might be better for those cases to only consider a subset of the total train set to be spread. For the pairs which have more than 4 trains to be spread, we will only take 4 trains into account during optimization.

6.1.1 Comparing the newly developed spreading method to the model in literature

In Chapter 3.2 the hypothesis was made that the newly developed model will perform better when there are 2 or 3 trains to be spread and it was unclear which model would perform better when 4 or more trains need to be spread. This hypothesis will be tested in the following. For 2,3 and 4 alternative spread pairs, both models will be given the same amount of solver time. The goal is to determine if there are significant differences in performance between the two. Next to spreading, transfers also have a big influence on computation time since integer variables are necessary to model transfers. To really measure the differences in *spreading*, we impose a transfer threshold, as it is found that when no transfer threshold is considered, the computation time is dominated by the transfer constraints and objective. If the number of passengers using this transfer is smaller than the threshold, the transfer will not be taken into account in the constraints and goal function. For a different reason a spreading threshold is necessary. If we do not impose a spreading threshold, the solver is not able to find a feasible solution. The two methods will be compared in terms of MIP gap and objective function found. The concept of MIP gap will be briefly discussed here.

The common way to solve an MIP is by using a so called branch and bound algorithm. A relaxation of the problem is formulated, where the integers variables are not needed to be integer anymore, but can be real numbers. This problem is solvable in polynomial time and the solution to this relaxed problem gives information about the optimal solution, because the search space is reduced. The algorithm recursively splits the search space into smaller spaces, where the goal function is minimized on these smaller spaces. This splitting process is called branching. Using these branches, lower bounds on the optimal objective value can be determined. The lower bound is updated during optimization, for a minimization problem the best lower bounds will be in increasing sequence. In case the best solution found so far has the same value as the theoretical lower bound, optimality of the solution is proved. However, as MIP problems are generally NP-hard, an actual optimal solution is not always found. The MIP gap is defined as the procentual difference between the best found solution and the theoretical lower bound:

$$Gap = \frac{\text{Best solution} - \text{Best lower bound}}{\text{Best lower bound}} \cdot 100$$

So if a solution has for example an MIP gap of 20%, this means that the solution is maximally 20% off of the actual optimal solution. However, it might also be for example 5% off of the optimal solution. The MIP Gap states the maximum difference between the two.

The results of comparing the two spreading methods are represented in the following. The first table gives the result for 600 seconds of computation time, where it is assumed that every passenger arrives randomly so every passenger experiences excess journey time. The second table shows the results for 2400 seconds of computation time, where it is assumed that only 20% of passengers arrive randomly, so only for these passengers the excess journey time will be added to the goal function. Furthermore, the amount of binary variables, general variables and Special Ordered Sets (SOS) stated by CPLEX after presolve are stated.

Pair size	Spr. Thres.	Tr. Thres.	#Instances	Model	#Bin	#Gen	#SOS	Gap	Obj
2	600	1000	29	New	0	1528	698	5.61	2215147
				Old	42	1532	812	9.60	2305833
3	400	1000	23	New	38	1672	728	14.83	2341767
				Old	342	1562	842	18.57	2455034
4	400	1000	4	New	24	1496	630	12.06	1446454
				Old	64	1460	630	11.27	1423384

Results for 600 seconds of computation time, 100% of passengers arriving randomly

Pair size	Spr. Thres.	Tr. Thres.	#Instances.	Model	#Bin	#Gen	#SOS	Gap	Obj.
2	600	1000	29	New	0	1528	698	5.51	1456698
				Old	42	1532	812	8.77	1505818
3	400	1000	23	New	38	1672	728	10.51	1491910
				Old	342	1562	842	15.06	1566617
4	400	1000	4	New	24	1496	630	12.14	1375055
				Old	64	1460	630	4.08	1258859

Table 4: Results for 2400 seconds of computation time, 20% of passengers arriving randomly

As can be seen, when 2 or 3 trains need to be spreaded, the newly developed model achieves a better optimality gap and a better objective function than the model described in literature. If 4 trains needed to be spreaded, the new model performs worse than the old model. Concluding, the best way to take into account spreading is to use the newly developed model when 2 of 3 trains need to be spreaded and use the model already developed in literature in case 4 trains need to be spreaded. The results from now on are generated using this "mixed model". One of the main questions of this research was if a new way of spreading alternative train routes can be developed, which can give a satisfactional optimality gap when applied to large networks. The results of this previous research, [5], are stated in the first table where the Belgium network was considered. The second table states the results of applying the mixed model to the Dutch intercity network.

Spread Threshold	900	800	700	600	500	400
# Spreading instances	14	16	22	24	30	31
Gap	13.2	14.3	17.13	-	-	-
Computation time	1200	1200	1200	2400	2400	2400

Table 5: Previous research with a transfer threshold of 2000, 26 lines and 2 linear segments

Spread Threshold	1000	800	600	400	200	0
# Spreading instances	43	48	52	67	79	164
Gap	20.56	19.71	20.47	20.78	29.75	-
Computation time	2400	2400	2400	2400	2400	2400

Table 6: Results of new model with a transfer threshold of 700, 34 lines and 5 linear segments

For the previous research, when 24 train lines needed to be spreaded, no solution could be found in 2400 seconds. The new model applied to the Dutch network performs significantly better. Spreading up to 79 trains is possible in 2400 seconds with an quite acceptable optimality gap. This suggests that the newly developed model performs significantly better. However another explanation could be a difference in network structure or an inferior implementation in the previous research. To draw a definite conclusion, the model already developed in literature will be compared to the newly developed mixed model on the same instances, namely the Dutch networks with different values for the transfer and spreading thresholds. This results in the following table:

			Transfer Threshold					
			0	0	100	100	300	300
			Mixed	Old	Mixed	Old	Mixed	Old
Spreading Threshold	200	Gap	-	-	-	-	-	-
		Obj	-	-	-	-	-	-
	400	Gap	-	-	32.72	-	35.58	36.88
		Obj	-	-	4056433	-	3441572	3875430
	600	Gap	37.07	-	37.06	42.93	31.74	32.96
		Obj	4304657	-	4238497	4593576	3614792	3605577
	800	Gap	37.92	-	34.46	38.73	30.40	33.99
		Obj	4363728	-	4075983	4267258	3503657	3631924

Table 7: Comparing the newly proposed mixed model with the old model already developed in literature

The mixed model always performs better in terms of optimality gap achieved and performs better in terms of objective function in all but one case. Furthermore there are 3 instances where the mixed model was able to find a solution, whereas the old model could not. In the case that both models find a solution the mixed model tends to perform better, but not extremely better. Important to notice is that the old model performs significantly better than was reported in previous research. This can be caused by various reasons, for example it could be that the network used in previous research considered more sets where 4 trains are spreaded compared to this test case. Spreading 4 trains is found for both models to be significantly harder than spreading 2 or 3 trains. Other reasons could be thought of like different network structures or implementation. The conclusion can be made that significant better results are achieved in terms of spreading than previous research and that the newly developed model is slightly superior to the model already developed.

6.1.2 Threshold analysis

As was stated before both a transfer threshold as well as a spreading threshold have been introduced to actually find feasible solutions. The following table shows for the mixed model the results for different threshold combinations. In parentheses, the percentage of passengers taken into account for a certain threshold are stated.

			Transfer Threshold				
			0 (100%)	100 (96.7%)	300 (82.8%)	500 (66.9%)	700(46.8%)
Spreading Threshold	200 (65.1%)	Gap	-	-	-	-	29.75
		Obj	-	-	-	-	2885772
	400 (49.35%)	Gap	-	32.72	34.58	31.58	20.78
		Obj	-	4056433	3841572	3313979	2487416
	600 (38.8%)	Gap	37.07	37.06	31.74	28.25	20.47
		Obj	4304657	4238497	3614792	3094823	2437618
	800 (31.4%)	Gap	37.92	34.46	30.40	27.03	19.71
		Obj	4363728	4075983	3503657	3076353	2380011
	1000 (25.6%)	Gap	35.50	33.96	30.96	27.85	20.56
		Obj	4148878	3980864	3495965	3026566	2374210

Table 8: MIP gap and objective function found after 2400 seconds of computing

The general trend, as to be expected, is that a lower thresholds results in a bigger MIP gap. An interesting question is what the influence of not taking all passengers into account is on those passengers. By not taking these passengers into account by the imposed thresholds, it might be that their transfer and spreading times are very bad. On the other hand, it might be the case that imposing these thresholds does not affect these passengers too much. In the following, a solution with relatively high transfer and spreading thresholds of 700 and 1200 will be generated and the average transfer and excess journey time will be computed. Then using the exact same solution, the average transfer and excess journey time will be computed when the thresholds are lowered. This results in the following figures:

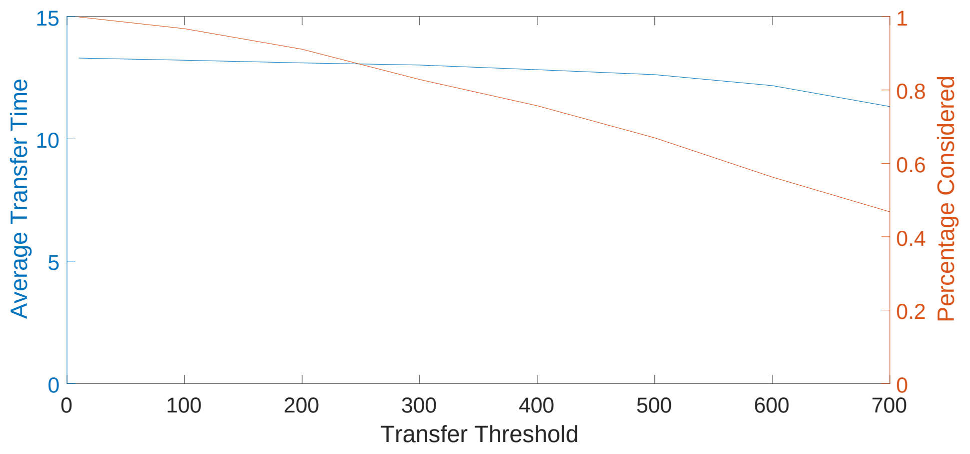


Figure 25: Average transfer time for different transfer thresholds, using solution computed with a threshold of 700

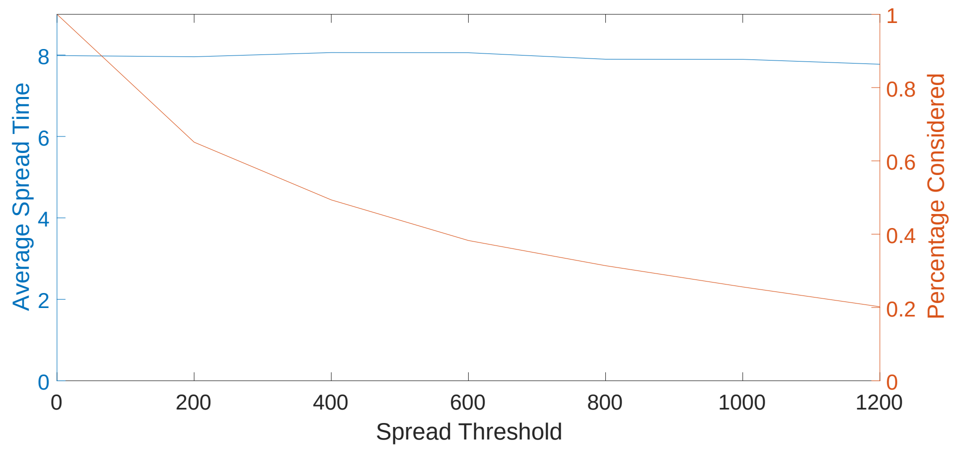


Figure 26: Average excess journey time for different spreading thresholds, using solution computed with a threshold of 1200

For the average transfer time, an increase is observed when lowering the transfer threshold. However this increase is not dramatically big. For the spreading threshold, no notable change is recorded. This can be explained by the following. Consider 3 consecutive stations. Suppose the introduced spreading thresholds results in the spreading of station 2 not taken into account. As station 1 and station 3 are taken into account, the trains will be spread as best as possible at those stations. It would be very unlikely that the trains at station 2 will now be spread very badly. Both figures give rise to the conclusion that imposing these transfer and spreading thresholds does not affect solution quality too much, especially if the spreading sets resulting from the imposed threshold are distributed rather uniformly over the network.

6.1.3 Analyzing the solution in terms of inter arrival times

Now the solution with regards to spreading on a specific instance will be analyzed. A transfer threshold of 500 and a spreading threshold of 600 are considered. A solution is computed when all passengers are assumed to be arriving randomly and a solution is computed when 50% of passengers are assumed to arrive randomly. Both solutions will be compared in terms of the interarrival times determined by the solver. The results are in the figures below, where the left column is the solution of the case where 100% of passengers experience excess journey time, the middle column is the solution where 50% of passengers experience excess journey time and the right column states the inter arrival times in a schedule developed by NS. This schedule was the result of a study by NS where a schedule was developed that could have 6 hourly trains between Utrecht and Amsterdam, but has not been put to practice. Both solutions were computed using 1 hour of computation time. The horizontal line indicates the value if the trains were to be spreaded perfectly.

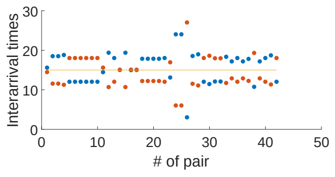


Figure 27: Spreading 2 trains, 100% random arrivals

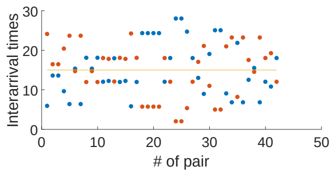


Figure 28: Spreading 2 trains, 50% random arrivals

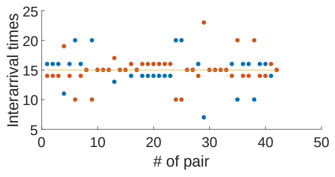


Figure 29: Spreading 2 trains, schedule by NS

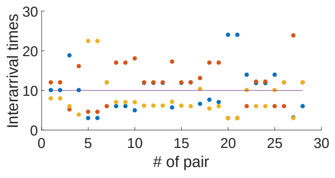


Figure 30: Spreading 3 trains, 100% random arrivals

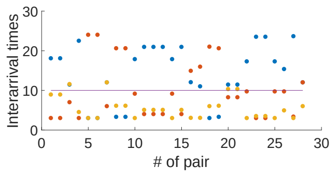


Figure 31: Spreading 3 trains, 50% random arrivals

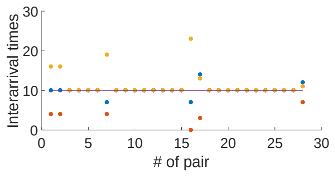


Figure 32: Spreading 3 trains, schedule by NS

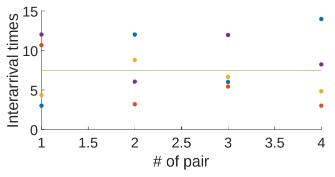


Figure 33: Spreading 4 trains, 100% random arrivals

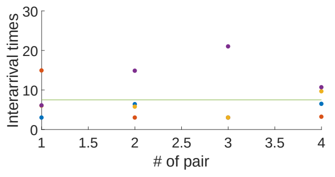


Figure 34: Spreading 4 trains, 50% random arrivals

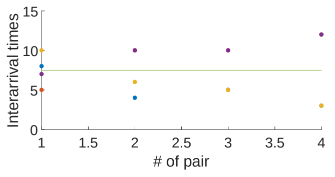


Figure 35: Spreading 4 trains, schedule by NS

As can be seen, the value of the interarrival times for the solution where all passengers experience excess journey time are generally closer to the optimum spreading value than the inter arrival times for the solution where 50% experience excess journey time. Because more passengers arrive randomly in the first case, more weight will be put on the spreading of trains. There are some interesting conclusions to be made from these figures. First, even when all passengers are assumed to be arriving randomly, trains are not perfectly spread in the solution. Consider the following example. Suppose at a certain station a passenger can take train 1 and train 2, to get to its destination. The inter arrival times between these 2 trains are added to the goal function and have an optimum at 15 minutes. Suppose there is a second passenger at the exact same station which can take train 1,2 and 3 to get to it's destination. These trains are tried to be spread 10 minutes apart. As a consequence train 1 and train 2 can not be spreaded perfectly in both cases. So a trade-off will be made between spreading them as close as possible to 15 and spreading them as close as possible to 10. Another explanation could be that spreading trains perfectly might have negative effects on other passengers. For example a perfectly spreaded train pair might induce a bad transfer possibility for an other passenger. Again a tradeoff is going to be made. This effect is exactly what we see in the results. For the case of every passenger arriving randomly, an average transfer time of 12.44 is found in the solution and an average excess journey time of 7.15 minutes per passenger. In case 50% of passengers arriving randomly, the average transfer time is 12.07 and the average excess journey time is 7.97. So the average transfer time declines, whereas the average excess journey time rises. Concluding, there is a tradeoff between having well spreaded trains and providing fast transfers.

Comparing the solutions with the timetable developed by NS, we can see that generally NS spreads trains more evenly, especially in the case where 3 trains need to be spreaded. In current scheduling, NS attaches great importance to spreading trains perfectly. The found interarrival times by the model in this research suggest that it is not necessarily best to do this. These solutions suggest that, in terms of expected passenger travel time, NS is overspreading trains. As spreading trains perfectly affects other passengers negatively, a tradeoff between the two should be made. This model is a way of making this tradeoff.

6.1.4 Computational analysis

We will now analyze some computational aspects. First, the influence of the amount of linear segments used in the goal function on the solution found is discussed. The results below state for different amounts of linear segments used the optimality gap after 1 hour of computing. Furthermore, the goal function of the solution using the linearized goal function as well as the non-linearized goal function are stated.

Numpiece	2	5	8	10	15	40
Gap	18.99	19.51	18.30	20.84	21.99	22.22
Objective	2640484	2285103	2217306	2279591	2294381	2261771
Non-linearized objective	2355440	2255644	2207290	2273046	2291840	2261770

Table 9: Number of linear segments used compared to the optimality gap achieved, Transfer Threshold=700, Spreading Threshold=1200

The general trend for the optimality gap is that the bigger the amount of segments used, the bigger the optimality gap. However, for the cases considered, the optimality gap is not very sensitive to the amount of segments used. Though this could also be explained by the relatively high thresholds used, making the problem "easy" for any amount of linear segments. Throughout this report, 5 linear segments have been used to compute solutions. For this case, the difference between the approximated goal function and the non-linearized goal function is 1.306%. So there is not a big difference between the approximated goal function and the actual goal function. This suggests that using 5 linear segments approximates the original function reasonably well.

An interesting observation is the value of the non-linearized goal function for different amounts of segments used. There seems to be a trade-off between having a more inaccurate approximation of the goal function, which is easier to optimize and on the other hand, having a very accurate goal function, which is more difficult to optimize. The optimal amounts of segments in terms of non-linearized goal function, for this case, is somewhere in the middle of the two, at 8 segments.

The last aspect that will be considered is the progress of the best found solution and the biggest lower bound during optimization. It is quite interesting to consider how these develop during optimization. For example it might be that the value of best solution found is decreasing steadily over time or rapidly at specific moments. The same holds for the lower bounds. For a specific instance the following graph illustrates the development of these factors, where the solver had 26000 seconds of computation time, or a little more than 7 hours.

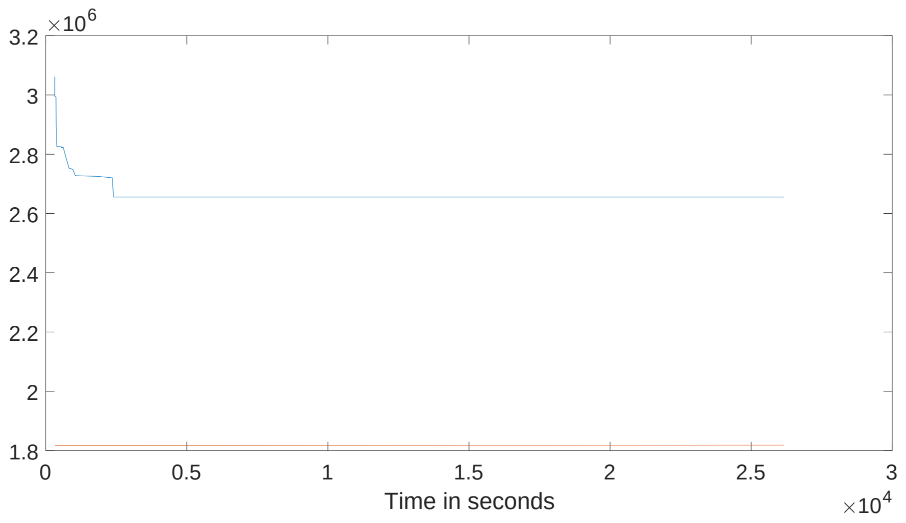


Figure 36: Biggest lower bound and best solution found over time

The top line is the value of the best solution found. The bottom line is the value of the best upper bound. The value of both have been recorded since the first solution was found. Two observations can be made. First, after around an hour of computation time, the solution is not improved anymore. Second, the lower bound increases very subtle and not clearly visible in the graph. This might be caused by the non-convexity of the goal function. Because of this non-convexity, the solver can not conclude if a local minimum of a global minimum is found. This possibly explains the behavior of the lower bound. It might be the case that the solution is of quite a good quality, but due to this non-convexity it can not be proved that it is of good quality, because of the lower bound that is not increasing. Before a first solution was found, the lower bound did increase significantly.

6.1.5 Error bounds

As was mentioned before, unless stated otherwise, 5 linear segments have been used in the piecewise linear approximation of the goal function. In Chapter 4 error bounds were shown between the original problem and its piecewise linear approximation. From [28], we found that if $\max|f(x) - f^{Lin}(x)| \leq \epsilon$, then also $|f(\hat{x}) - f^{Lin}(x^*)| \leq \epsilon$, for a univariate function. In Chapter 4.3.5 error bounds when using a multivariate separable goal function have been formulated. The maximum error found was $\sum_{i=1}^{n} \left|f_i(x_i) - f_i^{Lin}(x_i)\right|$. For the case of using 5 linear segments, we ideally would like to compute this sum. However, because we would have to determine the maximum difference for every function used in the optimization problem, this is impractical. We will give a general indication of the maximum error by considering a single passenger following a certain route. When using 5 linear segments, for a certain set of delay parameters, the maximum difference between the linearized goal function with the spread, dwell, departure and transfer functions are determined. This results in 0.145, 0.014, 0.011 and 0.249 respectively. So when considering a single passenger following a route which consists of for example a spread action, a departure action, 3 consecutive dwell actions, a transfer action and last 4 consecutive dwell actions, the maximum difference between the value of the actual optimum and the solution of the approximated problem is as follows: $0.145 + 0.011 + 3 \cdot 0.014 + 0.249 + 4 \cdot 0.014 = 0.503$. This is just the maximum difference in terms of goal function for a single passenger following a specific route. Ideally we would determine the maximum difference for every single passenger and take the sum to get the total maximum difference between the optimal solutions. Considering a single passenger gives a general indication of the quality of approximation when having 5 linear segments.

6.1.6 Some remarks regarding the TRANS toedeler

Because the order of trains is not yet known, the TRANS toedeler made some distributions of passengers which would not be realistic in an actual timetable. Consider for example a passenger traveling from origin to destination where he needs to make a transfer at a certain station. Suppose there are two trains where the passenger can start from, and two trains to where the passenger can transfer to. Then the TRANS toedeler distributes passengers over all possible transfer combination of trains because it is unknown which train comes after which train. The following figure explains this:

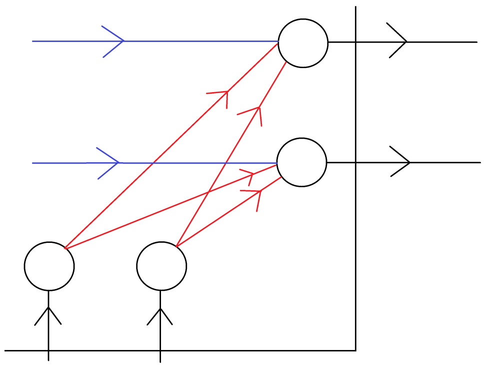


Figure 37: Transfers generated by TRANS toedeler when no order of trains is known

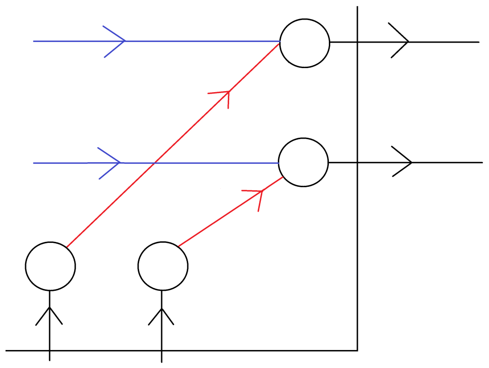


Figure 38: Example of transfers happening in reality

In reality when a timetable is given, passengers would make use of a single connection, namely to the train which arrives first at the transfer station, for example as shown in Figure 39. However, this is an inevitable problem. It can not be implemented in the TRANS toedeler because it is unknown to which train passengers are actually traveling, because of the unknown order of trains. Fixing the order of trains, for example after a first schedule has been found, could solve this problem.

6.2 Result independence assumption

We will now consider if the solution of the independent model is a good approximation of the solution of the dependent model. To do so we will consider a small train network consisting of the intercity between Enschede and Hoofddorp and the sprinter between Utrecht and Zwolle, shown in the figure below.

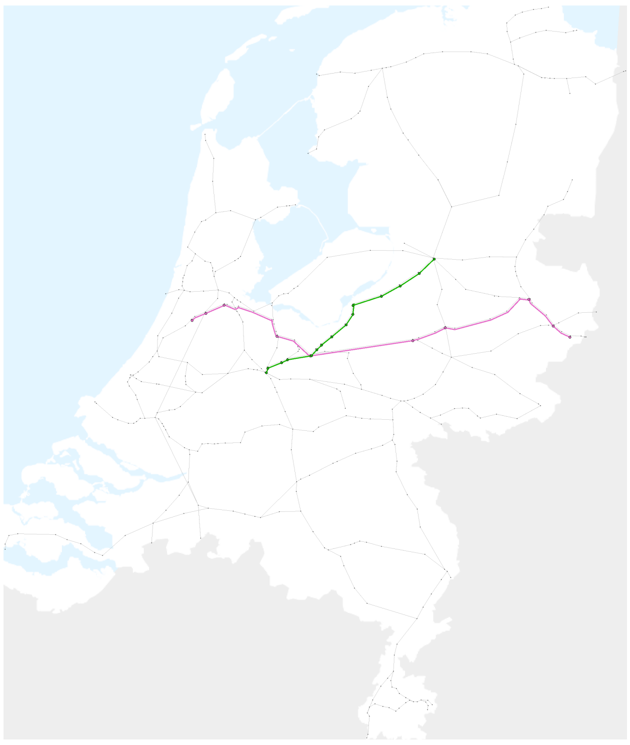


Figure 39: The intercity between Enschede and Hoofddorp and the sprinter between Utrecht and Zwolle

We will compute a solution to this network using both the independent model and the dependent model. Since we think the dependent model is a more accurate description of reality, we will plug the solution of the independent model into the dependent model and determine the value of the goal function. Then we will compare the difference in goal function between the two solutions. Ideally the goal function of the solution of the independent model plugged into the dependent model is close to the goalfunction of the solution using the dependent model. In that case we can conclude, at least for this network, that splitting the network in several independent parts is not a bad approach.

Of course when comparing the two, we need to keep in mind that we are only considering a small network. The results found for this network may not hold for a bigger network, like the entire intercity network. Because of computational considerations for the dependent model, a small network was chosen as a test case.

For several values of the delay factor a, the models will be compared. Also the case where there are no transfers, so the two networks can be viewed as two independent networks, will be examined. For the independent model, 50 piecewise linear segments will be used. For the dependent model, 200 samples were used in the goal function. This results in the following figures:

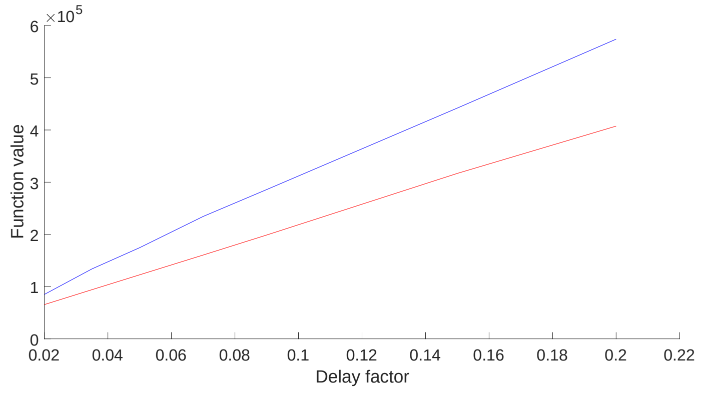


Figure 40: With transfers, top line independent model, bottom line dependent model

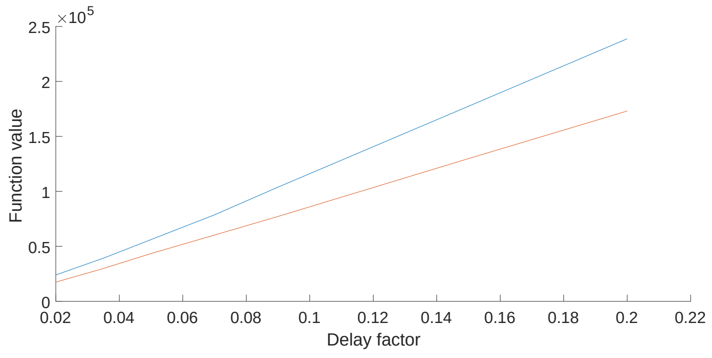


Figure 41: Without transfers, top line independent model, bottom line dependent model

As we can see, there is quite a difference between the goal function for the two models. The objective function of the solution of the independent model plugged in to the dependent model is roughly 25% bigger than the solution of the dependent model. The main difference between the solution of the two models is that the buffers in the dependent model are in general slightly bigger than the buffers generated by the dependent model. For a delayfactor of $a = 0.09$ the buffers of the two models are stated below.

IM	Forward	15.922	0.28	0.971	9.88	1.88	0.62	1.71	0.68
	Backward	0.42	0.39	1.57	0.82	12.5	0.34	0.97	0
DM	Forward	2.92	0.47	1.49	8.89	1.23	1.32	2.92	0.85
	Backward	0.83	0.65	3.73	1.53	11.9	0.37	1.31	0

Table 10: Buffers Intercity Enschede-Hoofddorp

IM	Forward	0.10	0.10	0.04	0.84	0.08	0	0.22	0.15	0.28	0.27	0.22	0.10	0.08
	Backward	0.18	0.31	0.19	0.58	0.21	0.10	0.28	0	0.35	0.51	0.08	0.08	0.08
DM	Forward	0.12	0.31	0.30	1.04	0.09	0	0.47	0.23	0.43	0.43	0.27	0.23	0.14
	Backward	0.33	0.69	0.80	1.08	0.39	0.22	0.71	0	0.69	0.93	0.14	0.18	0.12

Table 11: Buffers Sprinter Utrecht-Zwolle

Some buffers in the solution are really big and some buffers are zero. This is explained by the datasets used. The first buffer is really big, because the very first station is a rolling stock repository. This station is not a station where passengers can board a train. The train will have no passengers boarded when arriving at the second station, which is why a big buffer is chosen there. There is a similar explanation for the buffers which are chosen to be 0. Some stations are not known in the OD-matrix, as they did not exist yet at that time. The line plan used in optimization does traverse through these stations. Because the station is not known in the OD-matrix, no passengers will be boarding the train here, but there are already passengers in the train from previous stations. This is why these buffers are chosen to be 0.

As can be seen the buffers of the dependent model are generally bigger than the buffers of the independent model. This is what we expected beforehand because apparently the predelay, which the independent model assumed to be (close to) 0, is not zero. Where the independent model assumes a delay of $\max\{0, x_n + y_n - D_n\}$ at edge n, in reality there is a preadelay $PD > 0$ at edge n. So the actual delay is $\max\{0, PD + x_n + y_n - D_n\}$ at edge n.

Apparently, the assumption of the independent model that the predelay tends to be (close to) 0 is not right. As we can see, not taking into account this predelay has severe negative effects on the quality of the solution. The heuristic explained in Chapter 5.4 to cope with this problem has been applied to the same network. This results in the following figures, where the middle line is the solution of the problem using the heuristic and the top and bottom line are the exact same solutions we found already.

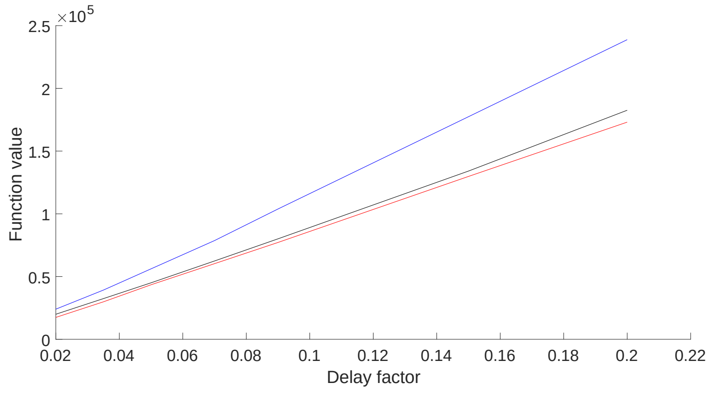


Figure 42: Without transfers

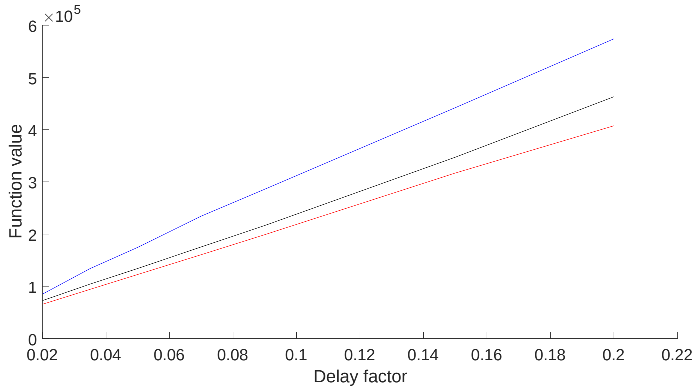


Figure 43: With transfers

As we can see the solutions are far closer to the actual optimal solution than before. Especially for the model without transfers this is the case. The difference between the actual optimal solution and the solution as a result of the scheme is now generally between 5 and 10 percent. This is a big improvement. This result suggests that the independent model can be a close resemblance of reality, however a parameter update scheme is necessary.

7 Conclusions and future research

Conclusions

We have developed expressions for the expected passenger travel time for both exponential and Weibull distributed delays. For a Weibull distribution however, no analytical expressions are possible. By numerical integration Weibull distributed delays are still applicable to the problem. A new way of modeling excess journey time has been developed. By considering the cyclic order of trains, for certain cases, less additional variables and constraints were necessary to take into account the yet unknown order of trains than previous research has provided. The goal function was approximated by a piecewise linear function. Several criteria have been given to guarantee a certain performance of the approximation with respect to the original function. Furthermore, the simplification of viewing a passenger route as a set of consecutive independent actions was considered. A model has been derived to solve the problem without making this simplification.

It is found that the new model for spreading trains has significantly better results in both computation time as well as optimality gap than previous research of the subject. However, it is found that the approach used in previous research also works quite well on the instances provided in this report, though not as good as the newly developed model. It is unclear why the previously developed model works significantly better on the Dutch network than on the network described in [5]. A possible explanation is that in the research of [5] a lot more spreading instances where more than 3 trains have to be spread were considered. These have shown to increase computation time dramatically compared to spreading 2 or 3 trains. In any case, for the instances to be considered for the Dutch network, quite satisfactional results have been found. It was also shown that the thresholds needed in order to find a feasible solution do not affect the travel time of the passengers not taken into account as a consequence that much. We can confidently implement these buffers, though they should of course not be extremely big. One of the main research questions was if the model can be applied to large scale networks. The model is able to find solutions for the entire Dutch intercity network which, depending on the thresholds chosen, results in a quite acceptable optimality gap. An interesting follow up question would be if the model is able to find solutions for the entire Dutch network, including so called "sprinters". This has not been tried, but it is suspected to be quite a bit more difficult than considering just the intercity network. This is because the Dutch network is very dense in the sense of trains using the same pieces of infrastructure.This is especially the case in the Randstad. Adding these sprinters could make it much more difficult to even find a feasible solution, let alone an optimal one.

An interesting conclusion was made when comparing the distribution of inter arrival times of the solutions found with the schedule provided by NS. The interarrival times of the schedule of NS were generally closer to the theoretical optimal inter arrival times. However, spreading trains very well can have negative effects on the quality of transfers. This is exactly what was noticed when comparing two solutions of the model, where a trade off was observed between the spreading of trains and the quality of transfers. As consequence, NS might emphasize perfect spreading too much at the moment and could profit, in terms of expected passenger travel time, from relaxing the need for perfect spreading and thus create an overall better schedule.

Lastly it was found that considering a passenger route as a set of independent action has a significant negative impact on the quality of the solution. Because of not taking into account the delay still in the network as a result of previous delays, the buffers generated were generally too small. A heuristic has been developed which has made impressive results on the test case described. However, we should be careful with drawing rushed conclusions on these results, as no mathematical explanation has been found to show this is a technique that is guaranteed to work.

Future research

Several directions for future research can be given. An interesting approach would be to somehow fix the order of trains. If the order of trains is fixed, the problem of the spreading of trains becomes much easier. No additional binary variables have to be added to model this unknown order of trains and this is expected to reduce computation time. The order of trains can for example be fixed by running the model described to get an initial timetable. This order could be fixed in the next computation. As explained, passengers can more accurately be distributed by the TRANS toedeler when the order of trains is fixed.

We have seen that it was necessary to introduce thresholds for both transfers as well as spread instances. If a spread instance was not used by enough passengers then it was not taken into account in the goal function. The spreading instances taken into account in optimization could be selected in a different way. As was explained before, if trains are spreaded well at station 1 and 3, it is unlikely they are very badly spread at station 2. By selecting the spreading instances depending on the structure of the network, we could implicitly also spread trains at stations not considered in the goal function.

Another aspect to be researched is the independence assumption. It was shown that dividing the network up into several independent parts has a severely negative effect on the solution quality. A scheme was developed to adjust parameters to compensate for the predelay not taken into account. For the network considered this resulted in significant improvement in solution quality. It would be interesting to see if the same holds for bigger networks because now only a fairly small network of two lines was considered. Also other approaches than the one explained of updating the parameters could be considered. Furthermore, a mathematical analysis could be made to show whether or not the scheme can be guaranteed to work on arbitrary networks.

Lastly some computational suggestions may be taken into account in future research. Firstly, the spreading objective, in which functions of the form s_i^2 were added to the goal function, can possibly be modeled more efficiently. This is a quadratic programming problem with a positive definite goal function. Positive definite quadratic programs are known to be solvable in polynomial time, see for example [31]. In the model used however, the goal function is approximated by a piecewise linear function just like the other goal functions. This is not necessarily solvable in polynomial time, so explicitly implementing the s_i^2 terms as a quadratic program might reduce solution time. Another computational consideration is the following. As we have seen the transfer action is not convex. The very start of the graph tends to have a slight kink, causing the non-convexity. As a consequence, it might be that the solver uses an entire different solving technique than would be used for a convex function. As a consequence, solver time might increase significantly because of this, as it can not be guaranteed that a global optimum has been found instead of just a local optimum. This might also explain why the solver has a hard time updating the biggest lower bound found during computation. This kink however is very slight. One could choose to just ignore the kink and make an approximation that is actually convex. Now convex solver techniques can be applied. It might even be the case that a piecewise linear approximation of the goal function is not necessary as interior-point algorithms can give quite good results for convex problems, though not every convex problem is solvable in polynomial time. The solution space stays non-convex in any case because of the integer constraints.

References

[1] Leo Kroon, Dennis Huisman, Erwin Abbink, Pieter-Jan Fioole, Matteo Fischetti, Gábor Maroti, Lex Schrijver, Adri Steenbeek, and Roelof Ybema. The new dutch timetable: The or revolution. 2008.

[2] Sofie Burggraeve, Simon Henry Bull, Richard Martin Lusby, and Pieter Vansteenwegen. *Integrating robust timetabling in line plan optimization for railway systems.* TU Management Engineering, 2016.

[3] Lukas Back, Twan Dollevoet, and Dennis Huisman. *Integrating Timetabling and Crew Scheduling at a Freight Railway Operator.* Econometric Institute Report, 2014.

[4] Luis Cadarso and Angel Marin. Integration of timetable planning and rolling stock in rapid transit networks. 2011.

[5] Peter Sels. Large-scale, passenger oriented, cyclic railway timetabling and station platforming and routing. 2016.

[6] Leon W. P. Peeters. Cyclic railway timetable optimization. 2003.

[7] A. Schrijver and A. Steenbeek. Spoorwegdienstregelingontwikkeling. 1993.

[8] P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, page 550–581, 1989.

[9] Odijk. Construction of periodic timetables; part i: A cutting plane algorithm. *Tech. Rep.*, 1994.

[10] Gert-Jaap Polinder. Resolving infeasibilities in the pesp model of the dutch railway timetabling problem. 2015.

[11] Steffen Hölldobler, Norbert Manthey, and Peter Steinke. Solving periodic event scheduling problems with sat. 2012.

[12] Rob Goverde, Ingo Hansen, Gerard Hooghiemstra, and Hendrik Lopuhaa. Delay distribution is railway stations. 2001.

[13] Mark Harris. Analysis and modelling of train delay data. 2006.

[14] J. Yuan. Stochastic modelling of train delays and delay propagation in stations. 2006.

[15] Kannemadugu Reddikrishna. Study on delay distribution of trains. 2015.

[16] Ryan Martin. Lectore notes on statistical theory. 2015.

[17] Felix Noyanim Nwobi and Chukwudi Anderson Ugomma. A comparison of methods for the estimation of weibull distribution parameters. 2014.

[18] Paritosh Bhattacharya. Weibull distribution for estimating the parameters. 2011.

[19] P. Vansteenwegen and D. van Oudheusden. Developing railway timetables which guarantee a better service. *European Journal of Operations Research*, 2004.

[20] Leo G. Kroon, Rommert Dekker, and Michiel J.C.M. Vromans. Cyclic railway timetabling: a stochastic optimization approach. 2005.

[21] John R Birge and Francois Louveaux. *Introduction to Stochastic Programming.* Springer, 1997.

[22] Stochastic linear programming. https://neos-guide.org/content/stochastic-linear-programming. Accessed: 10-02-2017.

[23] Andrew Schaefer. An overview of an overview of sampling methods in sampling methods in stochastic stochastic programming programmin. http://egon.cheme.cmu.edu/ewo/docs/SPSeminarSchaefer_October16.pdf. Accessed: 10-02-2017.

[24] Sample Average Approximation (SAA) for Stochastic Programs with an eye towards computational saa. https://www.ima.umn.edu/materials/2015-2016/ND8.1-12.16/25374/morton_ima.pdf. Accessed: 10-02-2017.

[25] Mark A. Turnquist. A model for investigating the effects of service frequency and reliability on bus passenger waiting times. 1978.

[26] Aduardo Camponogara and Luiz Fernando Nazari. Models and algorithms for optimal piecewise-linear function approximation. *Hindawi*, 15, 2015.

[27] Nenad Ujevic. New error vounds for the simpson's quadrature rule and applications. *Elsevier*, 2006.

[28] Arthur M. Geoffrion. Objective function approximations in mathematical programming. 1977.

[29] Rob Harron. The leibniz rule. http://math.hawaii.edu/~rharron/teaching/MAT203/LeibnizRule.pdf. Accessed: 18-07-2017.

[30] A. M. Vaziri nd A.V. Kamyad, A. Jajarmi, and S. Effati. A global linearization approach to solve nonlinear nonsmooth constrained programming problems. 2011.

[31] Paul Tseng. A simple polynomial-time algorithm for convex quadratic programming. 1988.

Appendices

A Goal function

A.1 Depart action

A.1.1 Exponential distribution

$$\begin{aligned}
f^{Depart}(D) &= \mathbb{E}\left(C^{Depart}(D)\right) \\
&= \int_0^\infty \int_0^\infty \max\{0, x+y-D\} f_x(x) f_y(y) dy dx \\
&= \int_0^D \int_{D-x}^\infty (x+y-D) f_x(x) f_y(y) dy dx \\
&+ \int_D^\infty \int_0^\infty (x+y-D) f_x(x) f_y(y) dy dx.
\end{aligned}$$

$$\begin{aligned}
\int_0^D \int_{D-x}^\infty (x+y-D) f_x(x) f_y(y) dy dx &= \int_0^D f_x(x) \int_{D-x}^\infty (x+y-D) f_y(y) dy dx \\
&= \int_0^D \left[(x-D) e^{-\lambda_y (x-D)} + \int_{D-x}^\infty y f(y) dy \right] f_x(x) dx \\
&= \int_0^D \left[(x-D) e^{-\lambda_y (D-x)} + \frac{(\lambda_y (D-x)+1) e^{-\lambda_y (D-x)}}{\lambda_y} \right] f_x(x) dx \\
&= \int_0^D f_x(x) \frac{1}{\lambda_y} e^{-\lambda_y (D-x)} dx \\
&= \frac{1}{\lambda_y} e^{-\lambda_y D} \frac{\lambda_x}{\lambda_y(\lambda_x - \lambda_y)} \left(1 - \frac{e^{D(\lambda_x - \lambda_y)}}{\lambda_x - \lambda_y} \right) \\
&= \frac{\lambda_x}{\lambda_y(\lambda_x - \lambda_y)} \left(e^{-\lambda_y D} - e^{-\lambda_x D} \right).
\end{aligned}$$

$$\begin{aligned}
\int_D^\infty \int_0^\infty (x+y-D) f_x(x) f_y(y) dy dx &= \int_D^\infty f_x(x) \left[\int_0^\infty (x-D) f_y(y) dy + \int_0^\infty y f_y(y) dy \right] dx \\
&= \int_D^\infty \left[x - D + \frac{1}{\lambda_y} \right] f_x(x) dx \\
&= \int_D^\infty \left(\frac{1}{\lambda_y} - D \right) f_x(x) dx + \int_D^\infty x f_x(x) dx \\
&= \left(\frac{1}{\lambda_y} - D \right) e^{-\lambda_x D} + D e^{-\lambda_x D} + \frac{1}{\lambda_x} e^{-\lambda_x D} \\
&= \frac{1}{\lambda_y} e^{-\lambda_x D} + \frac{1}{\lambda_x} e^{-\lambda_x D}.
\end{aligned}$$

So we have:

$$f^{Depart}(D) = \frac{\lambda_x}{\lambda_y(\lambda_x - \lambda_y)} \left(e^{-\lambda_y D} - e^{-\lambda_x D} \right) + \frac{1}{\lambda_y} e^{-\lambda_x D} + \frac{1}{\lambda_x} e^{-\lambda_x D}.$$

A.1.2 Weibull distribution

The distribution of a Weibull distributed variable is $f(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-\left(\frac{x}{\lambda}\right)^k}$, the cumulative distribution is $F(x) = 1 - e^{-\left(\frac{x}{\lambda}\right)^k}$. Furthermore the expectation is as follows: $\mathbb{E}(X) = \lambda\Gamma(1+\frac{1}{k})$, with $\Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx$.

$$\begin{aligned}
f^{Depart}(D) &= \mathbb{E}\left(C^{Depart}(D)\right)\\
&= \int_0^\infty \int_0^\infty \max\{0, x+y-D\} f_x(x) f_y(y) dy dx\\
&= \int_0^D \int_{D-x}^\infty (x+y-D) f_x(x) f_y(y) dy dx\\
&+ \int_D^\infty \int_0^\infty (x+y-D) f_x(x) f_y(y) dy dx.
\end{aligned}$$

Let us first consider the second integral in the last expression:

$$\begin{aligned}
\int_D^\infty \int_0^\infty (x+y-D) f_x(x) f_y(y) dy dx &= \int_D^\infty f_x(x) \int_0^\infty (x-D+y) f_y(y) dy dx\\
&= \int_D^\infty \left(\int_0^\infty (x-D) f_y(y) dy + \int_0^\infty y f_y(y) dy\right) dx\\
&= \int_D^\infty \left(x - D + \lambda_y \Gamma\left(1+\frac{1}{k_y}\right)\right) f_x(x) dx\\
&= \int_D^\infty \left(\lambda_y \Gamma\left(1+\frac{1}{k_y}\right) - D\right) f_x(x) dx + \int_D^\infty x f_x(x) dx\\
&= \left(\lambda_y \Gamma\left(1+\frac{1}{k_y}\right) - D\right) e^{-\left(\frac{D}{\lambda_x}\right)^{k_x}} + \int_D^\infty \frac{k_x}{\lambda_x^{k_x}} x^{k_x} e^{-\left(\frac{x}{\lambda_x}\right)^{k_x}} dx.
\end{aligned}$$

Consider this last integral, $\int_D^\infty \frac{k_x}{\lambda_x^{k_x}} x^{k_x} e^{-\left(\frac{x}{\lambda_x}\right)^{k_x}} dx = \frac{k_x}{\lambda_x^{k_x}} \int_D^\infty x^{k_x} e^{-\left(\frac{x}{\lambda_x}\right)^{k_x}} dx$. By substituting $u = \left(\frac{x}{\lambda_x}\right)^{k_x+1}$ and $\frac{du}{dx} = (k_x+1)\left(\frac{x}{\lambda_x}\right)^{k_x}$, we get:

$$\begin{aligned}
\int_D^\infty x^{k_x} e^{\frac{-x^{k_x}}{\lambda_x}} dx &= \frac{\lambda_x^{k_x+1}}{k_x+1} \int_{\left(\frac{D}{\lambda_x}\right)^{k_x+1}}^\infty e^{-u^{\frac{k_x}{k_x+1}}} du\\
&= \frac{\lambda_x^{k_x+1}}{k_x+1} \left[- \frac{(k_x+1)\Gamma\left(1+\frac{1}{k_x}, u^{\frac{k_x}{k_x+1}}\right)}{k_x} \right]_{u=\left(\frac{D}{\lambda_x}\right)^{k_x+1}}^{u=\infty}\\
&= \frac{\lambda_x^{k_x+1}}{k_x} \Gamma\left(1+\frac{1}{k_x}, \left(\left(\frac{D}{\lambda_x}\right)^{k_x+1}\right)^{\frac{k_x}{k_x+1}}\right)\\
&= \lambda_x \Gamma\left(1+\frac{1}{k_x}, \left(\frac{D}{\lambda_x}\right)^{k_x}\right).
\end{aligned}$$

Where the third equality follows because $\Gamma(a,b)$ is defined as $\int_b^\infty t^{a-1}e^{-t}dt$, so $\Gamma(a,\infty) = 0$, and thus $-\left.\frac{(k_x+1)\Gamma\left(1+\frac{1}{k_x}, u^{\frac{k_x}{k_x+1}}\right)}{k_x}\right|_{u=\infty} = 0$. Now in total we have:

$$\int_D^\infty \int_0^\infty (x+y-D) f_x(x) f_y(y) dy dx = \left(\lambda_y \Gamma\left(1+\frac{1}{k_y}\right) - D\right) e^{-\left(\frac{D}{\lambda_x}\right)^{k_x}} + \lambda_x \Gamma\left(1+\frac{1}{k_x}, \left(\frac{D}{\lambda_x}\right)^{k_x}\right).$$

By making the same substitution as shown above and making use of the known cumulative distribution function, we get the expression for the remaining integral:

$$\begin{aligned}\int_0^D\int_{D-x}^{\infty}(x+y-D)f_x(x)f_y(y)dydx &= \int_0^D f_x(x)\left(\int_{D-x}^{\infty}(x-D)f_y(y)dy+\int_{D-x}^{\infty}yf_y(y)dy\right)dx\\ &= \int_0^D f_x(x)\left((x-D)e^{-\left(\frac{D-x}{\lambda_y}\right)^{k_y}}+\frac{k_y}{\lambda_y^{k_y}}\int_{D-x}^{\infty}y^{k_y}e^{-\left(\frac{y}{\lambda_y}\right)^{k_y}}dy\right)dx\\ &= \int_0^D\left[(x-D)e^{-\left(\frac{D-x}{\lambda_y}\right)^{k_y}}+\lambda_y\Gamma\left(1+\frac{1}{k_y},\left(\frac{D-x}{\lambda_y}\right)^{k_y}\right)\right]f_x(x)dx.\end{aligned}$$

Because of the dependency of the gamma function on x, it is unlikely an analytical expression can be found for the above integral. Now f^{Depart} is as follows:

$$\begin{aligned}f^{Depart}(D) = &\left(\lambda_y\Gamma\left(1+\frac{1}{k_y}\right)-D\right)e^{-\left(\frac{D}{\lambda_x}\right)^{k_x}}+\lambda_x\Gamma\left(1+\frac{1}{k_x},\left(\frac{D}{\lambda_x}\right)^{k_x}\right)\\ &+\int_0^D\left[(x-D)e^{-\left(\frac{D-x}{\lambda_y}\right)^{k_y}}+\lambda_y\Gamma\left(1+\frac{1}{k_y},\left(\frac{D-x}{\lambda_y}\right)^{k_y}\right)\right]f_x(x)dx.\end{aligned}$$

A.2 Transfer Action

A.2.1 Exponential distribution

$$\begin{aligned}f^{Depart}(D) &= D+\int_0^{\infty}\int_0^{\infty}\mathbb{1}\{X+Y>D\}\frac{T}{m}f_x(x)f_y(y)dxdy\\ &= D+\frac{T}{m}\left[\int_0^D\int_{D-x}^{\infty}f_x(x)f_y(y)dydx+\int_D^{\infty}\int_0^{\infty}f_x(x)f_y(y)dydx\right].\end{aligned}$$

Consider the first integral:

$$\begin{aligned}\int_0^D\int_{D-x}^{\infty}f_x(x)f_y(y)dydx &= \int_0^D f_x(x)\left[\int_{D-x}^{\infty}f_y(y)dy\right]dx\\ &= \int_0^D f_x(x)e^{-(D-x)\lambda_y}\\ &= \frac{\lambda_x}{\lambda_y-\lambda_x}\left(e^{-D\lambda_x}-e^{-D\lambda_y}\right).\end{aligned}$$

Consider the second integral:

$$\begin{aligned}\int_D^{\infty}\int_0^{\infty}f_x(x)f_y(y)dydx &= \int_D^{\infty}f_x(x)dx\\ &= e^{-D\lambda_x}.\end{aligned}$$

So in total we get:

$$\begin{aligned}f^{Depart}(D) &= D+\frac{T}{m}\left[e^{-D\lambda_x}+\frac{\lambda_x}{\lambda-y-\lambda_x}\left(e^{-D\lambda_x}-e^{-D\lambda_y}\right)\right]\\ &= D+\frac{T}{m}\left[\frac{(\lambda_y-\lambda_x)\,e^{-D\lambda_x}}{\lambda_y-\lambda_x}+\frac{\lambda_x\left(e^{-D\lambda_x}-e^{-D\lambda_y}\right)}{\lambda_y-\lambda_x}\right]\\ &= D+\frac{T}{m}\frac{\lambda_y e^{-D\lambda_x}-\lambda_x e^{-D\lambda_y}}{\lambda_y-\lambda_x}.\end{aligned}$$

A.2.2 Weibull distribution

$$f^{Depart}(D) = D + \frac{T}{m}\left[\int_0^D \int_{D-x}^{\infty} f_x(x)f_y(y)dydx + \int_D^{\infty}\int_0^{\infty} f_x(x)f_y(y)dydx\right].$$

Consider the first integral:

$$\begin{aligned}\int_0^D \int_{D-x}^{\infty} f_x(x)f_y(y)dydx &= \int_0^D f_x(x)\left[\int_{D-x}^{\infty} f_y(y)dy\right]dx\\ &= \int_0^D e^{\left(\frac{D-x}{\lambda_y}\right)^{k_y}} f_x(x)dx.\end{aligned}$$

This is not further integrable. Consider the second intergral:

$$\begin{aligned}\int_D^{\infty}\int_0^{\infty} f_x(x)f_y(y)dydx &= \int_D^{\infty} f_x(x)dx\\ &= e^{-\left(\frac{D}{\lambda_x}\right)^{k_x}}.\end{aligned}$$

So in total we have:

$$f^{Depart}(D) = D + \frac{T}{m}\int_0^D e^{\left(\frac{D-x}{\lambda_y}\right)^{k_y}} f_x(x)dx + e^{-\left(\frac{D}{\lambda_x}\right)^{k_x}}.$$

B Convexity of goal function

We will prove f^{Depart} is a convex function by showing the second derivative is bigger than 0.

$$\begin{aligned}\mathbb{E}\left(C(D))\right) &= \frac{\lambda_x}{\lambda_y(\lambda_x-\lambda_y)}\left(e^{-\lambda_y D}-e^{-\lambda_x D}\right)+\frac{1}{\lambda_x}e^{-\lambda_x D}+\frac{1}{\lambda_y}e^{-\lambda_x D}\\ \frac{d}{dD}\mathbb{E}(C(D)) &= \frac{\lambda_x}{\lambda_y(\lambda_x-\lambda_y)}\left(\lambda_x e^{-\lambda_x D}-\lambda_y e^{-\lambda_y D}\right)-\frac{\lambda_x}{\lambda_y}e^{-\lambda_x D}-e^{-\lambda_x D}\\ \frac{d^2}{dD^2}\mathbb{E}(C(D)) &= \frac{-\lambda_x^3 e^{-\lambda_x D}+\lambda_x\lambda_y^2 e^{-\lambda_y D}}{\lambda_y(\lambda_x-\lambda_y)}+\frac{\lambda_x^2}{\lambda_y}e^{-\lambda_x D}+\lambda_x e^{-\lambda_x D}\\ &= \frac{-\lambda_x^3+\lambda_x^2(\lambda_x-\lambda_y)+\lambda_x\lambda_y(\lambda_x-\lambda_y)}{\lambda_y(\lambda_x-\lambda_y)}e^{-\lambda_x D}+\frac{\lambda_x\lambda_y^2}{\lambda_y(\lambda_x-\lambda_y)}e^{-\lambda_y D}\\ &= \frac{\lambda_x\lambda_y\left(e^{-\lambda_y D}-e^{-\lambda_x D}\right)}{\lambda_x-\lambda_y}.\end{aligned}$$

If $\lambda_x \geq \lambda_y$, then $e^{-\lambda_y D}-e^{-\lambda_x D}\geq 0$ and $\lambda_x-\lambda_y\geq 0$. If $\lambda_x<\lambda_y$ then $e^{-\lambda_y D}-e^{-\lambda_x D}<0$ and $\lambda_x-\lambda_y<0$. So the second derivative is strictly positive, so the function is convex.

C Piecewise linearization algorithm

Now the matrix I determined by the algorithm determines the break points resulting in the smallest squared error.

Algorithm 1 Algorithm used to generate piecewise linearization

```
for $i = 1$ to $n$ do
    $E(i,i) = \infty$
    for $j = i + 1$ to $n$ do
        $a_{i,j} = \frac{(j-i-1)\sum_{k=1}^{j} x_k y_k - \left(\sum_{k=1}^{j} x_k\right)\left(\sum_{k=1}^{j} y_k\right)}{(j-i-1)\sum_{k=1}^{j} x_k^2 - \left(\sum_{k=i}^{j} x_k\right)^2}$
        $b_{i,j} = \frac{\sum_{k=i}^{j} y_k - a_{i,j}\sum_{k=i}^{j} x_k}{j-i+1}$
        $A_{i,j} = a_{i,j}$, $B_{i,j} = b_{i,j}$
        $E(i,j) = \sum_{k=1}^{j}(y_k - a_{i,j}x_k - b_{i,j})^2$
    end for
end for

for $j = 1$ to $n$ do
    for $t = j$ to $T$ do
        $F(j,t) = \infty, I(j,t) = 0; , X_{j,t}$
    end for
    $F(j,1) = E(1,j)$
    $I(j,1) = 1$
    $X(j,1) = 0$
    for $t = 2$ to $\min\{j-1, , T\}$ do
        $F(j,t) = \infty, I(j,t) = 0, X(j,1) = 0$
        for $i = t$ to $j - 1$ do
            $k = I(i, t-1)$
            if $k \neq 0$ and $A(k,i) \neq A(i,j)$ then
                $x = \frac{B(k,i)-B(i,j)}{A(i,j)-A(k,i)}$
                if $x_{\max\{1,i-1\}} \leq x \leq x_{\min\{n,i+1\}}$ and $F(j,t) > F(i,t-1) + E(i,j)$ then
                    $F(j,t) = F(i,t-1) + E(i,j)$
                    $I(j,t) = i$
                    $X(j,t) = x$
                end if
            end if
        end for
    end for
end for
```