An ILP-model for the train platforming problem

Department of Mathematics, Linköping University

Simon Calderon

LiTH-MAT-EX–2023/03–SE

Credits:	**30hp**
Level:	**A**
Supervisors:	**Anders Peterson**, Department of Science and Technology, Linköping University
	Fredrik Johansson, Swedish National Road and Transport Research Institute
Examiner:	**Kaj Holmberg**, Department of Mathematics, Linköping University
Linköping:	**June 2022**

Abstract

The goal of this thesis is to create an optimization model to optimize the routing of trains within railway stations. This problem is known as the train platforming problem, and the model we present is an integer programming model. By this model we aim to optimize factors such as walking distance, switch usage or platform usage.

We validate the model by implementing the model for Linköping station, which is a typical mid size station in the Swedish railway network. This implementation is done for different time horizons, ranging from 2 hours to one day, which corresponds to train sets ranging from 27 to 265 trains.

In the conclusion we see that the model is efficient for optimizing the train platforming problem for the implemented station and timetables, and that the model has a possibility to optimize the four objectives tested. Furthermore we see that optimizing certain objectives gives solutions that are also good with regards to other objective functions.

Keywords:

Optimization, Integer programming, Train platforming, Train scheduling

URL for electronic version:

Sammanfattning

Målet med den här uppsatsen är att skapa en optimeringsmodell för att optimera valet av vägar för tåg genom tågstationer. Modellen vi presenterar är en heltalsmodell, där syftet är att minimera bland annat gångavstånd, användningen av tågväxlar eller användningen av perronger.

För att testa modellen presenterar vi en implementation av modellen för stationen i Linköping, vilken är en typisk mellanstor station i det svenska tågnätet. Impplementeringen är gjord för olika tidslängder, från 2 timmar till ett dygn vilket motsvarar dataset från 27 till 265 tåg.

Vi drar slutsatsen att modellen på ett effektivt sätt kan lösa valet av tågvägar genom stationen, för de fyra tidtabeller och den station vi har implementerat. Vidare ser vi att modellen har potential att optimera de fyra målfunktioner vi testat och att optimering av några av målfunktionerna ger lösningar som är bra även med hänsyn till de andra målfunktionerna.

Nyckelord:

Optimering, Heltalsprogrammering, Tåg, Järnväg, Schemaläggning

URL för elektronisk version:

Acknowledgements

I would like to thank my supervisors Anders Peterson and Fredrik Johansson for introducing me to the subject of train scheduling and suggesting the problem. I would also like to thank Kaj Holmberg for taking his time to be the examiner of this thesis. Last but not least i would like to thank family and friends for their support.

Nomenclature

Notation

$\mathcal{T}$	The set of all trains
$\mathcal{P}$	The set of all platforms
$\mathcal{R}$	The set of all routes
τ_t^a	Arrival time of train t
τ_t^d	Departure time of train t
d_t^a	Arrival direction of train t
d_t^d	Departure direction of train t

Abbreviations

TTP	Train Timetabeling Problem
TPP	Train Platforming Problem
ILP	Integer Linear Programming
MILP	Mixed Integer Linear Programming

Contents

Chapter 1

Introduction

There are multiple different reasons for improving the railway. Travel by railway needs to become more attractive to meet the future's demands for more energy efficient transport. This motivates both the development of new and more efficient technologies and infrastructure, but also optimizing the current railway to make it more attractive to customers. There are also economic motivations. According to the Swedish transport administration the maintenance of the Swedish railway costs 6 billion Swedish crowns annually [21]. Also the production of new railway is much more expensive compared to the costs to produce new roads. It is therefore of interest to make optimal use of the infrastructure that is already in use.

One way to improve the railway is the way track owners work with train scheduling. Scheduling of trains primarily consists of setting time tables, choosing which routes the train takes and which platforms it will stop at, and finally the scheduling of stock and personnel. These phases can be seen as independent from one another, or be treated simultaneously. Traditionally train scheduling has been performed manually by train planners based on some simple rules and experience, which made it a time consuming and difficult task.

In train scheduling, the stations play a critical role. According to calculations made by the Swedish transport administration, passengers value time in transit between two methods of transportation 2.5 times as much as time spent in vehicle [20]. Due to the layout of most train stations, changing from one train to another can be difficult for those who are movement impaired. Hence we can optimize train routing for minimal walking distance between train changes to improve the passenger experience.

The routing of trains, especially inside stations, is a very complex problem. This is since the routing of one train highly affects the options for how we can route other trains. This is due to that each train needs a section of track ahead reserved, and during this reservation no other trains can occupy this section of track. Because of this, the delay of one train will affect the routing of other trains and create delay propagation.

The problem of assigning trains to routes and platforms inside a train station is known as the train platforming problem. Usually this is done in two phases. Firstly when the time table is set to assure that there is a fesable assignment of trains to routes. Secondarily the train platforming problem arises when one or more trains is delayed which creates a need to redo some or all platforming until all trains are running on time again.

There are many ways to model and optimize the train platforming problem (for more on the previous research see Chapter 3), in this thesis i will primarily use Integer Linear Programming.

1.1 Problem description

The primary objective of this thesis is to develop a mathematical model that can create practically possible assignments of trains to routes, and that can optimize the assignments for several different objectives. Such objectives include to minimize the number of platforms used, minimizing walking distance or assigning trains to platforms as close to the station entrance as possible.

Secondarily we want to compare the results when optimizing for the different objectives, and investigate how the different objectives are affected when optimizing for other objectives. The purpose of this is to see what properties can be optimized and to see at what expense.

For the sake of validation we present the results from an implementation of four different timetables for Linköping station, which is a quite typical mid-size railway station.

1.2 Approach and Method

To solve the train platforming problem we will use mathematical optimization, which is motivated by the large amount of previous research on similar problems that use optimization.

It turns out that the problem can be conveniently formulated as an integer linear program. The model will be implemented in AMPL and solved using the CPLEX solver

1.3 Delimitations

To restrict the problem to a feasible scope for the project a certain number of delimitations are needed. We will assume that we only control the train traffic inside the station, that is that we have a given direction of arrival and departure for each train that can not be changed. We will furthermore assume that the timetable is fixed and can't be changed, and that each train arrives and departs on time.

The goal is to develop a model that can work for any train station. To prove that the model works we will provide an implementation of the model at Linköping Central station. Furthermore, we will not consider shunting[1] decisions in the model, or in which order the trains are stored on the storage tracks.

1.4 Structure of the thesis

Chapter 2 contains a few preliminaries that are needed to understand the rest of the thesis. In chapter 3 we present a survey of some previous research on train platforming and other similar problems. The main contribution on the report is presented in Chapter 4, where the optimization model is presented, and an implementation for Linköping is presented in chapter 5. Chapter 6 contains the results of the implementation and finally the Conclusion and discussion is in chapter 7, along with suggestions for further research.

[1] Shunting is the process where rolling stock is sorted into complete trains

Chapter 2

Preliminaries

The purpose of this chapter is to give the necessary prerequisites to understand the results and to introduce the notations and conventions used in the rest of this thesis.

2.1 Optimization

The results of this thesis relies on the subject of mathematical optimization. For a more complete introduction to the subject, see the books Holmberg (2018) [11] or Lundgren et al. (2010)[16]. The following sections aims to introduce the notation that will be used in the rest of this thesis, and some results that are of special interest for the result.

Definition (Integer Linear Programming problem)

$$\min z = c^T x \tag{2.1}$$

$$\mathbf{A}x \leq b$$

$$x_i \in \mathbb{N}$$

The optimal value for the ILP is denoted as z^*_{IP}. If we replace the variables with $x_i \in \{0, 1\}$ we call the problem a binary integer problem.

Definition (LP Relaxation)

The LP relaxation of the integer problem in equation 2.1 is the problem

$$\min z = c^T x \tag{2.2}$$

$$\mathbf{A}x \leq b$$

$$x_i \geq 0$$

The optimal value for the LP-relaxation is denoted as z^*_{LP}.

Definition (Totally unimodular matrix) A matrix is said to be totally unimodular if and only if all the determinants of all its square submatrices are ± 1 or 0.

Definition (The integrality property) An IP-problem is said to have the integrality property if the optimal solution to the LP-relaxation is also a valid in the original IP-problem. In other words that is if $z^*_{IP} = z^*_{LP}$. A IP-problem has the integrality property if the matrix $\mathbf{A}$ is totally unimodular.

Defnition (Canonical linearisation). Given two binary variables x_1 and x_2, then we can transform the term $x_1 x_2$ to a linear term by introducing a new variable $w = x_1 x_2$. To make sure that w takes the right value based on x_1 and x_2 we introduce the constraints $x_1 \geq w$, $x_2 \geq w$, and $x_1 + x_2 - 1 \leq w$.

Chapter 3

Literature review

Multiple different research teams have worked on some version of the train platforming problem. However, each team have worked on quite different variants of the problem and have used different methods. The literature will be presented in three sections. The main focus is the first section which presents previous research on ILP and MILP models. Section 2 contains examples of other approaches for the TPP, and the third section is about research on similar problems to the TPP.

3.1 ILP and MILP methods

In three papers by Zwaneveld et al. from 1996, 1997 and 2001 ([24], [13] and [25]) the authors present a model to maximize the number of trains from a given timetable that can be assigned to a route through the station. Each platform and train combination is given a weight $\rho_{t,p}$ to model certain preferences for trains to stop at particular platforms.

The authors assume that each arrival and departure time, and the directions of arrival and departure are known a priori. The model used in the papers assume a cyclic timetable with length 60 minutes.

In the paper [24] the authors examine the computational complexity of the model, and conclude that if each train has more than three routes to choose from then the TPP is NP-complete. Furthermore Zwaneveld et. al. concluded that if each train has at most two routes to choose from, then the problem can be solved in $\mathcal{O}(|\mathcal{T}|^2)$, where $\mathcal{T}$ is the set of trains.

Finally in the paper [25] the authors show that their version of the TPP can be viewed as a weighted node packing problem. Each route and train combination is given a vertex and each vertex pair that correspond to the same train is an edge in the graph. Also each pair of incompatible assignments of trains to routes corresponds to an edge.

The authors introduce some preprocessing procedures and an algorithm for solving the weighted node packing problem, based on a branch and cut approach.

The model treated by Carprara et al. in [4] is a very general model with many things taken into consideration. Each train is only given an ideal arrival and departure time, and is allowed to shift from the ideal times by some set limit.

Furthermore a set of dummy platforms are introduced to handle potential trains over the capacity of the station. Each platform is associated with a cost, where the cost of the dummy platforms are significantly bigger than the costs of the ordinary platforms.

Each train is given a priority and a cost based on the priority of the train and how much the time is shifted from the ideal time given an assignment to a particular route.The model contains two types of incompatibilities between routes. There are those who are considered hard incompatibilities which are handled in the constrains of the model. There are also soft incompatibilities between routes, that can be violated at a certain cost.

To handle the costs of the soft incompatibilities Carprara et. al. get a square term in the objective function. In contrast to some other papers, the authors of [4] refrain from using the canonical linearisation, claiming that the resulting LP-relaxation would be too weak. Instead they make use of a set of clique inequalities from the graph of incompatible routes. This yields more constraints, but a tighter bound for the LP-relaxation.

The objective function used is a weighted sum of the costs of platforms, the costs associated with the used routes and the soft incompatibilities. The resulting model is too complex to solve explicitly, so the authors solve by branch and cut. The authors conclude that their model is able to improve the capacity of the three stations considered.

Petering et al [18] studied a combined timetabling and platforming problem. The objective was to create an cyclic timetable with as minimal cycle time, and to minimize journey times for each train. After implementing various preprocessing steps to reduce constraints and to make the model as efficient as possible, the authors present the results for several data sets with between 3 and 6 train lines.

Yet another approach is taken by Chakroborty and Vikram in [7]. They base their model around the assumption that arrival times of trains is only known one hour in advance. Furthermore the model takes into account that trains may need to be halted outside the station if all tracks happen to be occupied at the same time.

The authors formulate their model as a MILP-model with the objective to minimize trains that need to wait outside the station and maximizing assignments to preferred platforms. The authors conclude that for data sets up to 110 trains that the model has a short solve time.

In the paper by Akyol et. al. [1] the authors show that the TPP can be viewed as a parallel machine scheduling problem, and can be model using binary integer models used for the machine scheduling problem. Doing this lets the trains deviate from the original timetable, and the objective becomes to minimize this deviation. In a similar manner Zeng et. al [23] formulated the problem as a Job shop scheduling problem.

3.2 Other approaches

3.2.1 Graph theoretical approaches

One of the earliest studies of the train platforming problem was in a paper from 1998 by De Luca Cardillo and Mione [8] where they study the k L-list τ colourings of graphs. The main objective of the paper was to present an effective heuristic for finding a L-List τ colouring for a given k.

They do however present a representation of the train platforming problem that can be viewed as k L-list τ colouring of a graph. If we let the vertex set be the set of trains, and introduce some parameter I_{min}, corresponding to the minimum time between trains at a platform, then we can define the edges such that $(t_1, t_2) \in E$ if the arrival times differ by less than I_{min}.

By letting the set of colours equal the set of platforms and τ the set of incompatible assignments the problem can be viewed as a k L-list τ colouring.

Note that in this instance of the problem choosing a platform is the same as choosing a route. In other words, if we know where the train enters the station, where it exits the station, and which platform it stops at, then we also know the route of the train through the station. Furthermore the authors test the implementation of the heuristic for six scenarios, ranging from 41 to 242 trains, and from 5 to 21 tracks.

3.2.2 Heuristic approaches

Another variant of the problem was treated in a paper from 2003 by Carey and Carville [5].

One property that is unique for this model is that some platforms can be split into subplatforms where multiple trains can stop, one at each subplatform. This adds a lot of complexity in how assignment of a train to a given platform affects what other assignments are valid.

In contrast to most other papers, the goal of the authors in [5] is not to solve the train platforming problem using an optimization model, but instead to present an heuristic algorithm to solve the problem. The goal for the algorithm was not in the first place to solve the problem optimally, but instead to create a feasible solution, similar to those assignment created by train planners.

3.2.3 Real Time Dispatching

Most other papers considers solving the TPP in a planning context, long before the trains are supposed to run. Another situation where there is a need to route trains through stations is the real time situation. Since one delayed train affect the possible routings of other trains and might create further delays, dispatchers need to assign new departure times and routes in real time. This is known as the real time dispatching problem.

Such a situation is treated by Lamorgese and Mannino in [14]. The authors does not restrict the model to one line or one station, but instead formulate a model for the whole network. They proceed to present a model where the problem for the whole network can be decomposed into a problem for each station and one for each line. This model is solved similarly to how a Benders decomposition works, where the line problem acts as the master problem and the station acts as the slave problem.

3.2.4 Non-linear models and Max-plus automatas

In the paper [22] by Wang and Yue, the authors treat the platforming problem as a multi objective mixed integer non-linear problem, where the constraints are linear but one of the objective functions are not linear. Each train is given a preferred platform and the first objective is to maximmize the number of trains assigned to preferred platforms. The second objective function is to have a balanced use of the tracks. These objective functions are then weighted together. The authors found that for a timespan of three hours for a busy station the model was efficient at creating optimal assignments.

One of the later contributions on the topic of train platforming is the research presented by Besinovic and Goverde in [2]. In this paper the authors goal is to optimize for stability and minimize delay propagation. In contrast to many other recent papers the authors does not solve the optimization problem using integer problem programming, but instead uses a Max-plus automata.

3.3 Related research

3.3.1 Gate assignment at airports

A similar problem to the train platforming problem is the assignment of airplane to gates at an airport. One difference is that if need be not every plane needs to be assigned to a gate, instead planes can be assigned to load and unload passengers outside at the tarmac. Another difference to the TPP is that assignment of a plane to a given gate does not affect the possibility to assign other planes to other gates.

One of many examples where this problem has been studied is in the paper [10] by Ding et al, where the authors model the problem as a quadratic integer problem. In this model the authors use a multi objective optimization approach with two objective functions. The primary goal is to maximize the number of planes that are assigned to gates, and the secondary objective is to minimize the walking distance within the gates.

3.3.2 Allocation at bus stops

Another problem that is the somewhat similar is the allocation of bus lines at transit terminals. In contrast to the train platforming problem, the problem is to allocate whole lines (i.e. "all buses from Norrköping to Finspång arrive and depart from stop B2"). The constraints for the problem is very similar to the

TPP, with each line has to be assigned to one stop, we can't have two or more buses at the same stop at the same time.

One example of a paper that studies this type of problem is Lindberg et. al. in [15]. The authors present an integer linear optimization model to minimize the walking distance. To handle simultaneous assignments the authors use the canonical linearisation. Furthermore the authors compare their optimization model with random assignments and with assignment based on non transferring passengers. The conclusion presented in [15] is that the optimization model presented can on average improve the walking distances by 13 %, compared to random assignments.

3.4 Summary and conclusion of literature review

In conclusion there have been many variants of the platforming problem studied, many more than the sample above. There are multiple different approaches to create valid assignments such as integer programming, graph colourings, automatas or certain algorithms.

Furthermore, most of the papers considered puts the objective as improving capacity of stations, or minimizing error propagation. The passenger perspective such as walking distance or optimizing for cross platform transits is rarely considered. We also see that the train platforming problems is very complex and gets difficult to solve quickly if the numbers of possible routes gets large. Most papers resort to heuristical approaches.

Chapter 4

Optimization model

In this chapter we propose an optimization model for the train platforming problem.

4.1 Properties of the station and parameters

For a given railway station we have some set of platforms $\mathcal{P}$, which are sections of track where trains can stop and passengers can board and alight the train. We also have a set $\mathcal{D}$ of directions, which can be tracks in and out from the station, or storage tracks where trains not in use are stored. The directions can also be tracks towards a shunting area of the station.

For example, see the fictitious train station in figure 4.1. We have that $\mathcal{P} = \{P1, P2, P3, P4\}$, and set of directions are $\mathcal{D} = \{D1, D2, D3, D4, D5, U1, U2\}$.

Due to some platforms being shorter than some of the longer trains there are restrictions on which train can stop at which platform. To model this compatibility we create the parameter $c_{p,t}$, where $c_{p,t} = 1$ if train t can stop at platform p, and $c_{p,t} = 0$ otherwise.

For each pair of a platform $p \in \mathcal{P}$ and a direction $d \in \mathcal{D}$ there is a set of possible routes $\mathcal{R}_{p,d}$ linking the two. We denote the set of all routes at the station as $\mathcal{R}$. We will assume that all routes are known in advance, either by generated by manual calculation or by implementation of some route generating algorithm. An example of such an algorithm can be found in [17].

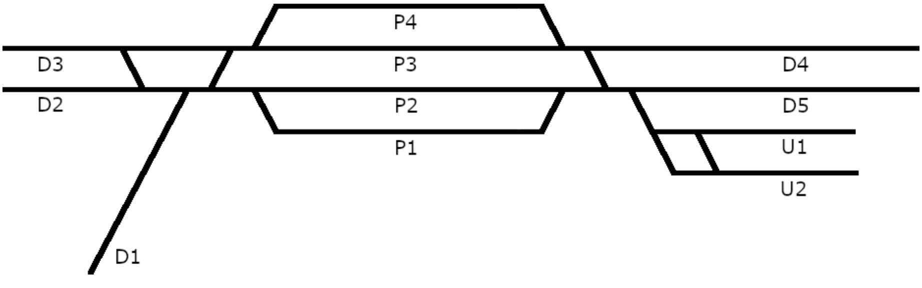


Figure 4.1: The layout of the tracks at a fictitious train station.

When a train arrives at the station a route is reserved from from the arrival point to the platform. During this time no other trains can use any section of track in this route. When the train has passed a section of track the section becomes no longer reserved. Similarly when a train leaves the platform a route is reserved from the platform to the point where the train exits the station.

In the model however the whole route is reserved for the train from the point it enters the station, until it stops at the platform, and then the whole outward route is reserved from when the train leaves the platform until the train has left the station.

Two routes is said to overlap if there is some section of tracks used by both routes. During the time that a route is reserved no other routes which overlap with the route can be used. To keep track of which routes overlap we let $a_{r,r'} = 1$ if the routes r and r' overlap, otherwise $a_{r,r'} = 0$. Take for example the routes r_1 from $D5$ to $P1$ and r_2 from $U1$ to $P2$. Since these two routes share a section of track we have $a_{r_1,r_2} = 1$

During the allotted time window we are to assign a set of trains $\mathcal{T}$ to platforms and routes. Each train $t \in \mathcal{T}$ has a given arrival time τ_t^a and a given departure time τ_t^d, which are the times that the train arrives to or departs from the platform. Furthermore we have an arrival direction $d_t^a \in \mathcal{D}$ and a departure direction $d_t^d \in \mathcal{D}$ for each train. The trains also has a number of passengers n_t that are to board or alight the train.

Π is the headway between trains at a given platforms, i.e. the time window required between two trains at the same platform for safety reasons. Ξ is the time it takes for the train to leave the station after leaving the platform, or the time it takes for the train from entering the station until it stops at the platform. In reality this depends on the route, the type of trains etc, but in the model we will assume the same headway for all train and route combinations.

We need to keep track of which trains are at the platforms simultaneously, to model the fact that each platform only can accommodate for one train at the time. To do this we introduce the set $\mathcal{E}_x \subseteq \mathcal{T} \times \mathcal{T}$, which consists of all pairs of trains that are at a platform during the same time.

In more detail we say that $(t_1, t_2) \in \mathcal{E}_x$ if $(\tau_{t_1}^a - \Pi/2, \tau_{t_1}^d + \Pi/2) \cap (\tau_{t_2}^a - \Pi/2, \tau_{t_2}^d + \Pi/2) \neq \emptyset$. This is equivalent to the arrival and departure times satisfying the inequality $(\tau_{t_1}^d - \tau_{t_2}^a + \Pi)(\tau_{t_2}^d - \tau_{t_1}^a + \Pi) < 0$. An example can be seen in figure 4.2.

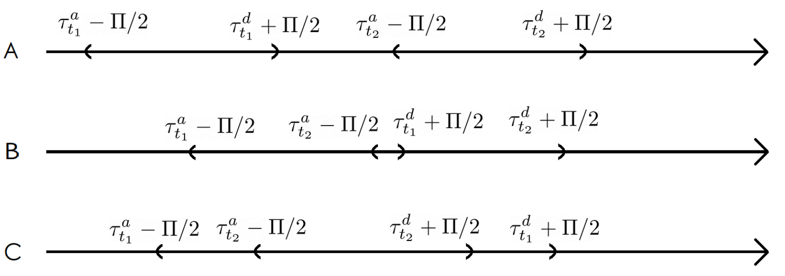


Figure 4.2: Three examples of how two time windows can relate to each other. In A the time windows does not overlap and hence $(\tau_{t_2}^d - \tau_{t_1}^a + \Pi)(\tau_{t_2}^d - \tau_{t_1}^a + \Pi) < 0$. In B we have a partial overlap and in C a total overlap. In both B and C we have that $(\tau_{t_2}^d - \tau_{t_1}^a + \Pi)(\tau_{t_2}^d - \tau_{t_1}^a + \Pi) \geq 0$. In B and C we have that the pair (t_1, t_2) belongs to $\mathcal{E}_x$, but this is not the case in A.

In a similar way we need to model incompatible assignments of routes for departure and arrival. This yields three cases, two arriving trains, two departing trains, and an arriving and a departing train.

The first two cases can be modelled in the same manner as the platform assignments. We create the sets $\mathcal{E}_y \subseteq \mathcal{T} \times \mathcal{T}$ and $\mathcal{E}_z \subseteq \mathcal{T} \times \mathcal{T}$, where $(t_1, t_2) \in \mathcal{E}_y$ or $(t_1, t_2) \in \mathcal{E}_z$ if (t_1, t_2) satisfy 4.1 respectively 4.2.

$$(\tau^a_{t_1} - \Xi, \tau^a_{t_1}) \cap (\tau^a_{t_2} - \Xi, \tau^a_{t_2}) \neq \emptyset \tag{4.1}$$

$$(\tau^d_{t_1}, \tau^d_{t_1} + \Xi) \cap (\tau^d_{t_2}, \tau^d_{t_2} + \Xi) \neq \emptyset \tag{4.2}$$

The conditions 4.1 and 4.2 are equivalent to (t_1, t_2) satisfying the inequalities 4.3 respectively 4.4.

$$(\tau^a_{t_2} - \tau^a_{t_1} + \Xi)(\tau^a_{t_1} - \tau^a_{t_2} + \Xi) < 0 \tag{4.3}$$

$$(\tau^d_{t_2} - \tau^d_{t_1} + \Xi)(\tau^d_{t_1} - \tau^d_{t_2} + \Xi) < 0 \tag{4.4}$$

To model the third case we need the set $\mathcal{E}_{yz} \subseteq \mathcal{T} \times \mathcal{T}$. In the contrast to the other cases the order of the pair is important, that is that $(t_1, t_2) \in \mathcal{E}_{yz}$ equivalent to $(t_2, t_1) \in \mathcal{E}_{yz}$. A pair of trains (t_1, t_2) is in $\mathcal{E}_{yz}$ if $(\tau^a_{t_1} - \Xi, \tau^a_{t_1}) \cap (\tau^d_{t_2}, \tau^d_{t_2} + \Xi) \neq \emptyset$. In other words t_1 is the arriving train and t_2 is the departing train. This condition is equivalent to the times satisfying the inequality $(\tau^a_{t_1} - \tau^d_{t_2})(\tau^d_{t2} - \tau^a_{t_1} + 2\Xi) < 0$.

4.2 Model

Based on the data given above we formulate the following model

$$\min f(x, y, z, w) \tag{4.5}$$

$$\sum_{p \in \mathcal{P}} c_{t,p} x_{t,p} = 1 \quad t \in \mathcal{T} \tag{4.6}$$

$$\sum_{r \in \mathcal{R}_{p,d^a_t}} y_{t,r} = x_{t,p} \quad p \in \mathcal{P}, t \in \mathcal{T} \tag{4.7}$$

$$\sum_{r \in \mathcal{R}_{p,d^d_t}} z_{t,r} = x_{t,p} \quad p \in \mathcal{P}, t \in \mathcal{T} \tag{4.8}$$

$$x_{t_1,p} + x_{t_2,p} \leq 1 \quad p \in \mathcal{P} \quad (t_1, t_2) \in \mathcal{E}_x \tag{4.9}$$

$$a_{r_1,r_2} y_{t_1,r_1} + a_{r_1,r_2} y_{t_2,r_2} \leq 1 \quad r_1, r_2 \in \mathcal{R} \quad (t_1, t_2) \in \mathcal{E}_y \tag{4.10}$$

$$a_{r_1,r_2} z_{t_1,r_1} + a_{r_1,r_2} z_{t_2,r_2} \leq 1 \quad r_1, r_2 \in \mathcal{R} \quad (t_1, t_2) \in \mathcal{E}_z \tag{4.11}$$

$$a_{r_1,r_2} y_{t_1,r_1} + a_{r_1,r_2} z_{t_2,r_2} \leq 1 \quad r_1, r_2 \in \mathcal{R} \quad (t_1, t_2) \in \mathcal{E}_{yz} \tag{4.12}$$

$$x_{t,p} \in \{0,1\} \quad y_{t,r} \in \{0,1\}, \quad z_{t,r} \in \{0,1\} \tag{4.13}$$

The model contains three sets of variables: $x_{t,p}$ which indicates if train t is assigned to platform p, $y_{t,r}$ indicating if train t is assigned to route r as an arrival route, and $z_{t,r}$ indicating if train t is assigned to route r as a departure route.

No particular objective function is specified here since the model is compatible with many different objective functions, more on that in the next section.

Constraint 4.6 assures that each train is assigned to one and only one compatible platform, while constraints 4.7 and 4.8 makes sure that if a train is assigned to a platform it is also assigned to a route to that platform.

Constraint 4.9 assures that if two trains can not be assigned to the same platform if their time windows overlap. Similarly the constraints 4.10 4.11, and 4.12 ensures that if two overlapping routes are used, then the time windows for when the routes are used does not overlap.

4.3 Objective functions

The model described above can be combined with numerous different objective functions. For the purpose of this thesis we want to compare objective functions that somehow relate to the passenger experience with objectives that improve different aspects of the railway operation. We have chosen four different objective functions that seek to minimize walking distance to station, total walking distance, switch usage and numbers of platforms in use.

Denoting δ_p as the walking distance from the platform p to the station entrance and exit, and n_t the number of passengers that are to board or alight on the train t, then we can minimize the total walking distance for passengers that are leaving or entering the station. This gives the objective function f_1 seen in equation 4.14.

$$f_1(x) = \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} n_t \delta_p x_{t,p} \tag{4.14}$$

We can also take into account the number of passengers that are changing from one train to another. If we know the walking distances between platforms $\delta_{p,p'}$, and the number of passengers $n_{t,t'}$ switching from train t to t', then we can minimize the total walking distance as the objective function

However to do this we need the capacity to model the simultaneous assignment of trains $x_{t,p}x_{t',p'}$. To preserve the linearity of the problem we make use of the cannonical linearization and introduce the variable $w_{t,t',p,p'} = x_{t,p}x_{t',p'}$ and constraints 4.15, 4.16 and 4.17. This lets us model the total walking distance with the function 4.18, which is the same objective function as in [15].

$$w_{t,t',p,p'} \leq x_{t,p} \quad p, p' \in \mathcal{P}, t, t' \in \mathcal{T} \tag{4.15}$$

$$w_{t,t',p,p'} \leq x_{t',p'} \quad p, p' \in \mathcal{P}, t, t' \in \mathcal{T} \tag{4.16}$$

$$w_{t,t',p,p'} \geq x_{t,p} + x_{t',p'} - 1 \quad p, p' \in \mathcal{P}, t, t' \in \mathcal{T} \tag{4.17}$$

$$f_2(x, w) = \sum_{t\in\mathcal{T}} \sum_{t'\in\mathcal{T}} \sum_{p\in\mathcal{P}} \sum_{p'\in\mathcal{P}} \delta_{p,p'} n_{t,t'} w_{t,t',p,p'} + \sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}} n_t \delta_p x_{t,p} \tag{4.18}$$

One of the more sensitive elements of the railway are the switches where different tracks meet. Switches sometimes break and require money and time for upkeep. It would therefore be of interest to minimize switch usage.

If we denote the number of switches in a given route as s_r, then we can formulate the objective function to be the number of switches passed in total, which give the objective function

$$f_3(y, z) = \sum_{t\in\mathcal{T}} \sum_{r\in\mathcal{R}} s_r(y_{t,r} + z_{t,r}). \tag{4.19}$$

There might also be interest to minimize the numbers of platforms that are in use. If a platform could be removed, we could also remove several switches and sections of track, which would in turn mean less money spent on upkeep. A station with fewer platforms might also be less confusing from a passenger perspective.

To implement such an objective function we need a few additional variables and constraints. If we define a binary variable h_p to indicate if platform p is used or not, and add the constraint $x_{t,p} \leq h_p$, then we can formulate the objective function as

$$f_4(h) = \sum_{p\in\mathcal{P}} h_p. \tag{4.20}$$

Chapter 5

Implementation

5.1 Implementation at Linköping Station

To test our model we have implemented the model for Linköping central station. This is a mid-size station, typical for the Swedish railway network. Various types of traffic pass through the station, including commuter trains, cargo trains and long distance trains. Adjacent to the railway station there is a bus station for both local and regional bus traffic. Note that no cargo trains stop at the station, but there are several that pass through the station, and that this is modelled as the cargo trains having the same arrival- and departure time.

The layout for the station can be seen in figure 5.1. This is only an approximation, but it is this layout that will be used for the implementation. In reality passengers can exit the station at both sides, but in the model we will assume that the passengers exit and enter at the side which is closest to $P1$.

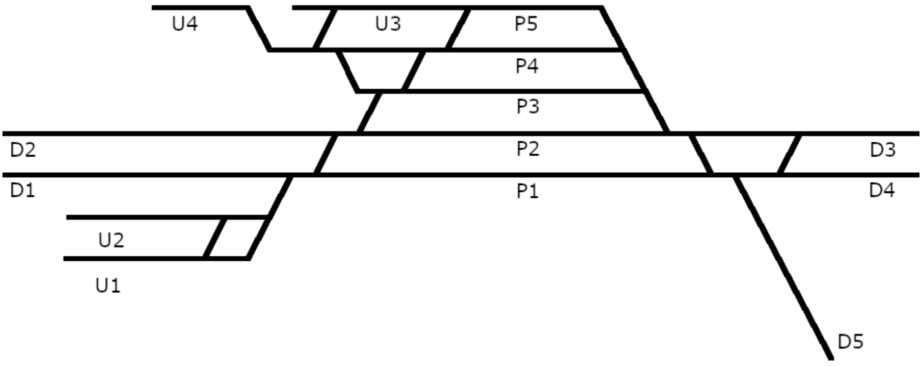


Figure 5.1: The layout of the tracks at Linköping central station

The station has three physical platforms, between $P5$ and $P4$, between $P3$ and $P2$, and adjacent to $P1$. Platforms 1-3 is long enough to accommodate all trains, where platforms 4 and 5 only are compatible with the shorter commuter trains. For detailed information regarding the platform compatibility in the implementation see appendix B.

The directions $D1, D2, D3, D4$ are the southern main line (Södra stambanan) and the direction $D5$ is the line towards Kalmar and Västervik (Stångådals-banan/ Tjustbanan). Besides the main directions we have four storage tracks for trains $U1, U2, U3$, and $U4$.

The possible routes for the station $\mathcal{R}$ were calculated by hand. This yielded 43 routes in total. On average each possible platform and direction combination has one possible route, with some combinations having more than one route. On the other hand, for example the combination $D2$ and $P1$ had no possible routes, since trains are not allowed to go backwards. The complete list of routes can be found in linköping.dat in appendix A.

5.2 Data

5.2.1 Timetables

The model is implemented for four different timetables denoted as by A-D. The timetables were based on the real timetable from April 1st 2021, which is a day with quite typical traffic. The real timetables can be found at [12] for the passenger trains and at [19] for the cargo trains.

A summary of the timetables can be found in table 5.1, and the full timetables can be found in Appendix B. The timetable D consists of all trains in a day, which is the longest time horizon that is interesting to optimize. This is due to that are very few or no trains that pass through the station during the night, which in turn leads to that platforming for one day have little to no impact on the platforming the next day.

Timetable	Start	Stop	$\lvert\mathcal{T}\rvert$
A	11:00	13:00	27
B	07:00	09:00	32
C	07:00	13:00	83
D	00:00	23:59	265

Table 5.1: A summary of the timetables used in the implementation.

For trains that arrive and depart the station we assume the convention of left hand traffic, and therefore trains arrive at $D2$ and $D4$, and depart at $D1$ and $D3$. Since Stångådalsbanan and Tjustbanan share a single track for the first part from Linköping trains both depart towards and arrive from $D5$.

No data was given about the departure direction and times of the trains that terminate at Linköping. Similarly no data of arrival time and directions was available for the trains that start at Linköping station. For those trains data was generated in the following manner.

- If a train terminates at Linköping and within 15 minutes a new train of the same type starts from Linköping then these two trains are assumed to be the same train.

- Otherwise we assume that the train stays five minutes at the station and departs to/arrives from some storage track. This storage track was arbitrarily chosen, with the same storage track for each train on the same line.

The arrival and departure directions of each train is included in appendix B. The capacity for each storage track is not considered, it is only assumed to be sufficient.

5.2.2 Passengers

The number of passengers where estimated in proportion to be half of the maximum number of passengers on each train type. Maximum number of passengers for each train can be found along with the timetables in Appendix B.

When optimizing for the objective function f_2 in equation 4.18, we need to know how many passengers will change from one train to another. This data was estimated as follows: if two trains depart more than one hour apart we assume that no passengers changes between these two trains.

Furthermore we assume that passengers are unlikely to change from one train to another of a similar type, that is for example changing from one commuter train on the line Linköping to Motala to another train on the same line. Besides these two restrictions we assume that passengers changes between all possible trains.

In the case for f_2 we assume that 40% of the maximum of passengers does not change trains, but instead arrive or depart from the station entrance. Furthermore we assume that 10% of max passengers changes to another train, equally distributed to all the available trains to change to.

The ratio between switching and non switching passengers were difficult to measure due to the ongoing coronavirus pandemic, and no data about travel before the pandemic was available to us. The ratio suggested above is a very rough estimate and we will compare different values later in a sensitivity analysis in section 6.5.

5.2.3 Linköping station

The walking distance between platforms are measured in a number of tracks needed to cross to get from the station entrance to the platform. Hence $\delta_1 = 0$, $\delta_2 = \delta_3 = 1$ and $\delta_4 = \delta_5 = 2$. By similar reasoning the values for $\delta_{p,p'}$ was approximated, the values can be found in Table 4.2.

$\delta_{p,p'}$	P1	P2	P3	P4	P5
P1	0	1	1	1	1
P2		0	0	1	1
P3			0	1	1
P4				0	0
P5					0

Table 5.2: The distances between platforms at Linköping station

In the implementation we assume that $\Pi = 1$ min and $\Xi = 1$ min.

Chapter 6

Results

The purpose of this chapter is to present the results of the implementation at Linköping which was presented in chapter 4. We will also investigate which is the maximal headway for which a valid solution exists.

The implementation was done in AMPL using the CPLEX solver, the code for which can be found in Appendix A. The code was run on a laptop with an 1.8 Ghz processor and 8 GB of internal memory. The AMPL presolve setting was enabled, which eliminated some variables and constraints for the purpose of reducing solve time.

6.1 Timetable A

The timetable A is based on a time window from 11:00 to 13:00, which is a period with relatively low traffic. The running times for optimizing for the four different objective functions, can be found in table 6.1. In this table Variables and Constraints denote the numbers before the presolve, while Variables* and Constraints* are after the presolve. All data in table 6.1 refer to the IP problem.

	Time	Variables	Constraints	Variables*	Constraints*
f_1^*	0.343s	2187	2815	553	2760
f_2^*	5.812s	20412	57490	14710	44763
f_3^*	0.047s	2187	2815	2148	2760
f_4^*	0.031s	2192	2950	557	2852

Table 6.1: Running times, variables and constraints.

Most notably is how f_2 is much more complex due to the added canonical linearisation. We are interested in how optimizing for one objective function affects the value of the other objective functions. A comparison of the values can be found in table 6.2. Comparisons between the LP and IP optimum can be found in table 6.3

	f_1^*	f_2^*	f_3^*	f_4^*
f_1	969.589	1050.6	1169.91	3148.05
f_2	1468.96	1404.62	1516.87	3490.22
f_3	212	193	152	324
f_4	5	4	4	3

Table 6.2: Comparison of values for the objective functions.

	f_1^*	f_2^*	f_3^*	f_4^*
LP optimum	969.589	1371.69	152	1.5
IP optimum	969.589	1404.62	152	3

Table 6.3: Comparison of values between the original problem and its LP relaxation.

Notably we see that LP optimum and IP optimum is the same for f_1 and f_3.

6.2 Timetable B

Timetable B is the trains between 07:00 and 09:00, which is a more traffic intense period than the one of timetable A.

	Time	Variables	Constraints	Variables*	Constraints*
f_1^*	0.125s	2592	7786	684	6995
f_2^*	13.376s	28192	84586	14820	48263
f_3^*	0.141s	2592	7786	2469	6995
f_4^*	0.063s	2597	7946	688	7092

Table 6.4: Running times, variables and constraints.

	f_1^*	f_2^*	f_3^*	f_4^*
f_1	1388.65	1464.01	1523.2	3187.59
f_2	2085.93	2034.77	2096.37	3797.96
f_3	232	237	189	318
f_4	5	4	4	3

Table 6.5: Comparison of values for the objective functions.

	f_1^*	f_2^*	f_3^*	f_4^*
LP optimum	1388.65	1987.3	186	2
IP optimum	1388.65	2034.77	189	3

Table 6.6: Comparison of values between the original problem and its LP relaxation.

Again we see that the LP and IP solutions coincide for f_1, but in contrast to A, the solutions are not the same for f_3. Also from table 6.2 we see that the values are the same for f_4 as the corresponding values in A.

6.3 Timetable C

Timetable C is longer than the previous two, and encompasses both timetable A, B and more trains.

	Time	Variables	Constraints	Variables*	Constraints*
f_1^*	0.14s	6723	12831	1555	11580
f_2^*	153.125s	178948	529506	93220	279591
f_3^*	0.219s	6723	12831	6422	11580
f_4^*	0.25s	6728	13246	1559	11822

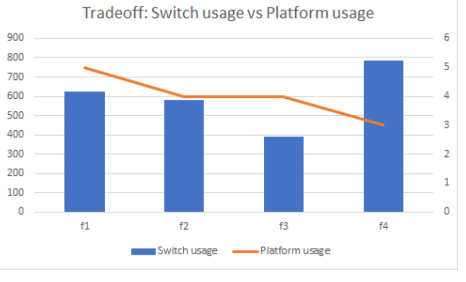
Table 6.7: Running times, variables and constraints.

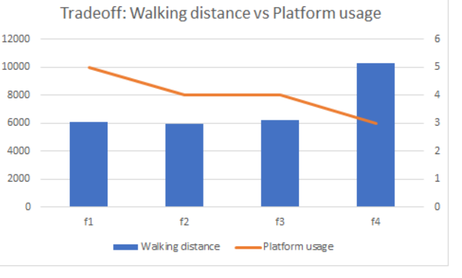
	f_1^*	f_2^*	f_3^*	f_4^*
f_1	4463.48	4676.78	4854.09	9295.75
f_2	6123.39	5976.67	6230.95	10824.9
f_3	623	581	490	783
f_4	5	4	4	3

Table 6.8: Comparison of values for the objective functions.

	f_1^*	f_2^*	f_3^*	f_4^*
LP optimum	4463.48	5927.8	487	2
IP optimum	4463.48	5976.67	490	3

Table 6.9: Comparison of values between the original problem and its LP relaxation.





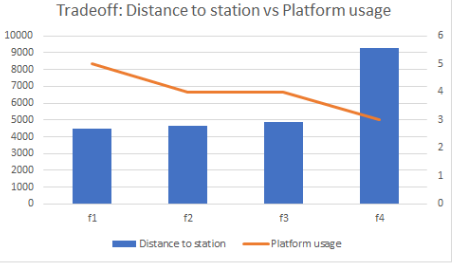


Figure 6.1: Each graph compares the value of one of the objective functions 1-3 (The blue staples) with the numbers of platforms used (f_4, the orange lines). Each staple represents optimizing for that objective function.

As we see in figure 6.1 the solutions we get from optimizing for f_1-f_3 are somewhat similar, while optimizing for f_4 gives much worse solutions in terms of f_1-f_3. Once again the solution for the LP relaxed problem is integer for f_1. We note that even for this larger dataset that the gap between the LP and IP solution seems to be quite small for all objective functions. If this is always the case, the LP solution can be used as a good upper bound on the IP objective function.

6.4 Timetable D

Timetable D takes the whole day into account, starting from midnight. As previously mentioned this is the longest time horizon that is interesting to optimize for due to the train traffic being sparse during the night.

	Time	Variables	Constraints	Variables*	Constraints*
f_1^*	0.266s	21303	39029	5644	36079
f_2^*	5439s	1750530	5226700	953296	2818750
f_3^*	0.312s	21303	39029	20421	36079
f_4^*	0.156s	21308	40344	5648	36843

Table 6.10: Running times, variables and constraints.

	f_1^*	f_2^*	f_3^*	f_4^*
f_1	13762.2	13945.4	14750.6	24410.3
f_2	18596	18274	18653.3	29251.6
f_3	1911	1752	1503	2386
f_4	5	4	4	4

Table 6.11: Comparison of values for the objective functions.

	f_1^*	f_2^*	f_3^*	f_4^*
LP optimum	13762.2	18066.7	1503	2
IP optimum	13762.2	18274	1503	4

Table 6.12: Comparison of values between the original problem and its LP relaxation.

Again we note that f_1 and f_3 is integral. It is also noteworthy that large number of variables when optimizing for f_2, which yielded the very long solve time.

6.5 Sensitivity analysis

We have this far seen that when optimizing for minimal walking distance that we get very similar results as when optimizing for proximity to station entrance and exit. This might be due to the fact that the fraction of passengers that switch from one train to another is not big enough to affect the assignments.

To study this we study five different scenarios, where 5%, 10%, 15%, 20% and 25% of maximum passengers are switching between train. Just as in the results above, the trains are assumed to always run at half of maximal passenger capacity. All these five scenarios are implemented for timetable C. The results can be found in table 6.13 and seen in figure 6.2. We see that the solutions

	f_1^*	f_2^*
5%		
f_1	5274.49	5381.14
f_2	6104.45	6031.09
10%		
f_1	4463.48	4676.78
f_2	6123.39	5976.67
15%		
f_1	3652.47	3972.42
f_2	6142.34	5922.26

	f_1^*	f_2^*
20%		
f_1	2841.46	3268.06
f_2	6161.28	5867.84
25%		
f_1	2030.45	2764.88
f_2	6180.23	5795.4

Table 6.13: The comparison of objective functions f_1 and f_2 for different levels of passengers that are switching from one train to another.

become less similar when the number of transiting passengers increases, but in general remain somewhat close (note that the axis in figure 6.2 does not start on 0).

6.6 Discussion

For the four timetables we implemented and for Linköping station there where some patterns and tendencies in the optimal solutions, and values for the objective functions.

Figure 6.2: The walking distance for the five different scenarios. The blue staples when optimizing for distance to station, the orange staples when optimizing for walking distance.

Firstly we note that $f^*_{1_{LP}} = f^*_{1_{IP}}$ for all of the four runs. It is not however the case that the constraint matrix is unimodular since the constraint matrix is the same for f_3, where $f^*_{3_{LP}} \neq f^*_{3_{IP}}$. These four results are not sufficient to show that the LP-relaxation is always integral, it might be the case four some other timetable or station that $f^*_{1_{LP}} \neq f^*_{1_{IP}}$.

From the sensitivity analysis in section 6.5 we note that the solutions for f_1^* and f_2^* were quite similar, even if a large fraction passengers switch trains. Under the assumption that most passengers does not switch trains, then we can use the objective function f_1, which is faster than f_2 to optimize for.

We see that the introduction of the canonical linearisation to model the function f_2 increases the complexity of the problem, and hence the solve time. This was most notable for the larger data sets.

In an earlier phase of the thesis a fifth objective function was considered. The quantity we sought to model was the number of crossing paths. By introduction of the three canonical linearisations $\lambda_{t,r,t',r'} = y_{t,r}y_{t',r'}$, $\mu_{t,r,t',r'} = z_{t,r}z_{t',r'}$ and $\nu_{t,r,t',r'} = y_{t,r}z_{t',r'}$, along with each new variables we introduced three sets of constraints similar to 4.15, 4.16 and 4.17. This gave the objective function

$$f_5(\lambda,\mu,\nu) = \sum_{t\in\mathcal{T}}\sum_{t'\in\mathcal{T}}\sum_{r\in\mathcal{R}}\sum_{r'\in\mathcal{R}} a_{r,r'}\left(\frac{\lambda_{t,r,t',r'}}{|\tau_t^a - \tau_{t'}^a|} + \frac{\mu_{t,r,t',r'}}{|\tau_t^d - \tau_{t'}^d|} + \frac{\nu_{t,r,t',r'}}{|\tau_t^a - \tau_{t'}^d|}\right). \quad (6.1)$$

This however proved immensely impractical. Solving for the optimal value of f_5 for the small timetable A resulted in around 1.8 million variables, and 5 million constraints after the presolve. The objective value could not be found after running the program for around 10 hours.

In conclusion we see that the introduction of one canonical linearisation is doable but significantly increases the number of variables and constraints, and therefore the running time. Using more than that in this context results in a program that is impossible to solve in a reasonable time frame, even for timetables that span a short timespan.

We also note that when optimizing for f_3 we got quite similar solutions as to those that optimize for f_2 and f_1. This is probably a result of the station layout at Linköping station. The routes through the stations that passes through the fewest switches is also the routes that stops closest to the station exit. Implementation for a station with different layout might not have this similarity.

Chapter 7

Conclusion

In this thesis we have presented an ILP-model to model viable assignments of trains to platforms and routes through a railway station. The model is shown to be compatible with at least four different objective functions to model for example walking distance or the number of platforms used.

To validate the model we presented an implementation for Linköping station for four different timetables. The running times for all these implementations was quick enough to be feasible to use, with exception for f_2 for the largest timetable. The solutions when optimizing for f_3 was quite close to the solution when optimizing for f_1 or f_2, which is probably due to the tracks closest to the station exit also being the tracks that passes through the least number of switches. This property is not universal, for example Norrköping central station does not have this property.

Before an optimization model similar to the one presented in this paper is used for train platforming there are several aspects that needs to be added to the model, but these results indicate that this type of model could be used to efficiently solve the TPP for at least small to medium sized stations.

It's possible that this model would work for larger stations, although this is not investigated in this thesis.

There seems to be a trade-off between the first three objective functions and number of platforms used. This might be due to the layout of Linköping station, to investigate this further implementation for other stations is needed.

The objective function f_2 had the longest running time, but we have seen that optimizing for f_1 gives a similar solution, assuming that most passengers doesn't switch to another train. Under the right conditions we could optimize for f_1 and get a solutions with good total walking distance.

7.1 Further research

In it's current form the model does not take into account trains that only pass through the station. This can be implemented as a train that arrives and departs at the same time from the platforms, but since these trains does not need to slow down they will reserve the track longer than necessary.

There is also need for more testing to validate the model. The model needs to be tested on larger stations to see if the running times are feasible. Furthermore the model also needs to be tested on stations with different features than Linköping, such as platforms that lie one after another on a single track, or tracks that terminate.

Depending on the results of the testing on different stations it could be interesting to modify the model to work at all stations. Little considerations were made when designing the model for computational efficiency, and it's not improbable that there are possible improvements in that regard.

Furthermore we have seen that in some instances that the LP-relaxation of the problem produces solutions that are very close to the IP solution. It would therefore be of interest to see if the model could be modified so that the LP optimal is always the IP optimal. If such a modification could be made, then it might be possible to save a some computation time, especially in the case of f_2 where the time difference for solving the LP and IP was significant. For reference the solve time for f_2 and timetable D was 5439s and the corresponding LP problem could be solved in 39s.

An approach that would be interesting to investigate is multi objective optimization, or weighted sums of objective functions. In reality we are almost never interested in minimizing just one property, but are instead interested in solutions that are "good" in all or most regards.

Appendix A

AMPL Code

A.1 Run file

```
#------------------------------------------------------------------
# SOLVER OPTIONS
#------------------------------------------------------------------

reset;                   # Resets everything
option solver './cplex'; # Choice of solver,
option display_eps 1e-6;
#------------------------------------------------------------------


#------------------------------------------------------------------
# FILES
#------------------------------------------------------------------

model TPP2.mod;

data TidtabellA.dat;
data Linkoping_ext.dat;

#------------------------------------------------------------------
# Generating passenger data
```

```
#-----------------------------------------------------------------


#Param A is used in the sensitivity analysis,
otherwise fixed at 10 percent
param A:=0.10;

for {t1 in TRAINS, t2 in TRAINS} {

if t1=t2 then{
let Changeing[t1,t2]:=0;
}

else if MaxPassengers[t1]=1 then{
let Changeing[t1,t2]:=0;
}

else if MaxPassengers[t2]=1 then{
let Changeing[t1,t2]:=0;
}

else if abs(dTime[t1]-dTime[t2])<60 then{

if  MaxPassengers[t1] <> MaxPassengers[t2] then{

let Changeing[t1,t2]:=1;
}

else let Changeing[t1,t2]:=0;
}
else let Changeing[t1,t2]:=0;

};


for{t1 in TRAINS}{

let n[t1]:=sum{t2 in TRAINS: t1<>t2}Changeing[t1,t2];
};
```

```
for{t1 in TRAINS, t2 in TRAINS} {

if Changeing[t1,t2]=0 then{

let PassengersT2T[t1,t2]:=0

}

else

let PassengersT2T[t1,t2]:=0.05*Changeing[t1,t2]*
(MaxPassengers[t1]/n[t1]+MaxPassengers[t2]/n[t2]);
};

for{t1 in TRAINS} {

let Passengers[t1]:=0.5*MaxPassengers[t1]
-sum{t2 in TRAINS:t1<>t2}PassengersT2T[t1,t2];
};
#-----------------------------------------------------------------
# Generating edge sets EDGES_X, EDGES_Y, EDGES_Z and E_YZ
#-----------------------------------------------------------------

let EDGES_X:={};
let EDGES_Y:={};
let EDGES_Z:={};
let EDGES_YZ:={};

for {t1 in TRAINS, t2 in TRAINS: t1<>t2} {

if (dTime[t2]-aTime[t1]+Headway_p)*(dTime[t1]-aTime[t2]+Headway_p)
 >0 then{

let EDGES_X:= EDGES_X union {(t1 , t2)};

}

if (aTime[t1]-aTime[t2]+Headway_r)*(aTime[t2]-aTime[t1]+Headway_r)
 >0 then{

let EDGES_Y:= EDGES_Y union {(t1, t2)};
```

```
}

if (dTime[t1]-dTime[t2]+Headway_r)*(dTime[t2]-dTime[t1]+Headway_r)
 >0 then{

let EDGES_Z:= EDGES_Z union {(t1,t2)};

}

if (aTime[t1]-dTime[t2])*(dTime[t2]-aTime[t1]+2*Headway_r)
 >0 then{

let EDGES_YZ:= EDGES_YZ union {(t1,t2)};

}

};

#----------------------------------
# TP_comp
#----------------------------------

for{t in TRAINS}{

if MaxPassengers = 309 then{

let TP_comp[t,1]:=1;
let TP_comp[t,2]:=1;
let TP_comp[t,3]:=1;
let TP_comp[t,4]:=0;
let TP_comp[t,5]:=0;

}


else


let TP_comp[t,1]:=1;
let TP_comp[t,2]:=1;
```

```
let TP_comp[t,3]:=1;
let TP_comp[t,4]:=1;
let TP_comp[t,5]:=1;

};
#------------------------------------------------------------------
# PROBLEM DEFINITIONS
#------------------------------------------------------------------

#-----------------------------------
# Distance to station
#-----------------------------------
problem Dist2Station: distance_to_station, x, y, z,
One_platform, xy_arrival, xz_departures,
platform_comp, route_comp_inin, route_comp_outout, route_comp_inout;
# Objective
# Constraints
# Variables

option relax_integrality 0;
option presolve 1;
#-----------------------------------
# Walking distance
#-----------------------------------
problem Total_Walking_Distance: walking_distance, x, y, z, w,
One_platform, xy_arrival, xz_departures,
platform_comp, route_comp_inin, route_comp_outout, route_comp_inout,
xw_relation_1, xw_relation_2, xw_relation_3;
# Objective
# Constraints
# Variables

option relax_integrality 0;
option presolve 1;
#-----------------------------------
# Switch usage
#-----------------------------------
problem Switches_usage: Switch_use, x, y, z,
One_platform, xy_arrival, xz_departures,
platform_comp, route_comp_inin, route_comp_outout, route_comp_inout;
# Objective
```

```
# Constraints
# Variables

option relax_integrality 0;
option presolve 1;
#-----------------------------------
# Platform usage
#-----------------------------------
problem Platform_usage: Platform_use, x, y, z, h, lambda, mu, nu,
One_platform, xy_arrival, xz_departures,
platform_comp, route_comp_inin, route_comp_outout, route_comp_inout,
xh_relation;
# Objective
# Constraints
# Variables

option relax_integrality 0;
option presolve 1;
#-----------------------------------
# Crossing paths
#-----------------------------------
problem CrossingPaths: Crossing_paths, x, y, z, lambda, mu, nu,
One_platform, xy_arrival, xz_departures,
platform_comp, route_comp_inin, route_comp_outout, route_comp_inout,
ylambda_relation_1, ylambda_relation_2, ylambda_relation_3,
zmu_relation_1, zmu_relation_2, zmu_relation_3,
yznu_relation_1, yznu_relation_2, yznu_relation_3;
# Objective
# Constraints
# Variables

option relax_integrality 0;
option presolve 1;
#-----------------------------------


#presolve_inteps >= 6.25e-06;



display card(TRAINS); #Displays the number of trains
```

```
solve Platform_usage;

display _objname, _obj;
display _objname, _obj>TPP.res;

display Switch_use >ALT.res;

display distance_to_station>ALT.res;



for{t1 in TRAINS, t2 in TRAINS, p1 in PLATFORMS, p2 in PLATFORMS} {

let w[t1,p1,t2,p2]:= x[t1,p1]*x[t2,p2];

}


display walking_distance> ALT.res;

for{t1 in TRAINS, t2 in TRAINS, r1 in ROUTES, r2 in ROUTES} {

let lambda[t1,r1,t2,r2] := y[t1,r1]*y[t1,r2];
let mu[t1,r1,t2,r2] := z[t1,r1]*z[t1,r2];
let nu[t1,r1,t2,r2] := y[t1,r1]*z[t1,r2];
}

display Crossing_paths ;

display {j in 1.._nvars: _var[j] > 0}
# Display only non-zero variables
(_varname[j], _var[j])>TPP.res;





display (_ampl_elapsed_time ) >Prop.res;
```

```
display (_total_solve_elapsed_time) >Prop.res;

display (_nvars) >Prop.res;
display (_snvars)>Prop.res;
display (_ncons)>Prop.res;
display (_sncons)>Prop.res;

display card(TRAINS) >Prop.res;
display card(PLATFORMS) >Prop.res;
display card(ROUTES) >Prop.res;

display card(EDGES_X) >Prop.res;
display card(EDGES_Y) >Prop.res;
display card(EDGES_Z) >Prop.res;
display card(EDGES_YZ) >Prop.res;
display EDGES_X > kanter.res;
display EDGES_Y > kanter.res;
display EDGES_Z > kanter.res;
display EDGES_YZ > kanter.res;

display PassengersT2T > byte.res;
display n > byte.res;
```

A.2 Mod file

```
#-------------------------------------------------------------
# MODEL FILE :  Train platforming problem (TPP)
#-------------------------------------------------------------

#-------------------------------------------------------------
# From Station data file
#-------------------------------------------------------------
set PLATFORMS;
set DIRECTIONS;
set ROUTES;
set SUB_ROUTES{PLATFORMS, DIRECTIONS};

param Distance{PLATFORMS};
param DistanceP2P{PLATFORMS, PLATFORMS};
param Overlap{ROUTES, ROUTES};
```

```
param Switches{ROUTES};

#-------------------------------------------------------------
# From timetable data file
#--------------------------------------------------------------

set TRAINS;

param aTime{TRAINS};
param dTime{TRAINS};
param aDir{TRAINS};
param dDir{TRAINS};
param MaxPassengers{TRAINS};


#--------------------------------------------------------------
# Generated in Run File
#--------------------------------------------------------------

set EDGES_X within {t1 in TRAINS, t2 in TRAINS: t1<>t2};
set EDGES_Y within {t1 in TRAINS, t2 in TRAINS: t1<>t2};
set EDGES_Z within {t1 in TRAINS, t2 in TRAINS: t1<>t2};
set EDGES_YZ within {t1 in TRAINS, t2 in TRAINS: t1<>t2};

param n{TRAINS};
param Changeing{TRAINS, TRAINS};
param PassengersT2T{TRAINS, TRAINS};
param Passengers{TRAINS};
param TP_comp{TRAINS, PLATFORMS};
#--------------------------------------------------------------
# Parameters
#--------------------------------------------------------------
param Headway_p := 1;
param Headway_r :=1;


#--------------------------------------------------------------
# Variables
#--------------------------------------------------------------

var x{TRAINS, PLATFORMS} binary;
```

```
var y{TRAINS, ROUTES} binary;
var z{TRAINS, ROUTES} binary;

var w{TRAINS, PLATFORMS, TRAINS, PLATFORMS} binary;

var h{PLATFORMS} binary;

var lambda{TRAINS, ROUTES, TRAINS, ROUTES} binary;
var mu{TRAINS, ROUTES, TRAINS, ROUTES} binary;
var nu{TRAINS, ROUTES, TRAINS, ROUTES} binary;



#--------------------------------------------------------------
# Objective functions
#--------------------------------------------------------------

minimize distance_to_station: sum{t in TRAINS, p in PLATFORMS}
Distance[p]*Passengers[t]*x[t,p];

minimize walking_distance:
sum{t in TRAINS, p in PLATFORMS, t2 in TRAINS, p2 in PLATFORMS}
DistanceP2P[p,p2]*PassengersT2T[t,t2]*w[t,p,t2,p2]
+sum{t in TRAINS, p in PLATFORMS}Distance[p]*Passengers[t]*x[t,p];

minimize Switch_use:
sum{t in TRAINS, r in ROUTES}Switches[r]*(y[t,r]+z[t,r]);

minimize Platform_use:
sum{p in PLATFORMS}h[p];

minimize Crossing_paths:
sum{t in TRAINS, r in ROUTES, t2 in TRAINS, r2 in ROUTES:t<>t2
and aTime[t]<>aTime[t2] and dTime[t]<>dTime[t2] and aTime[t]<>dTime[t2]}
Overlap[r,r2]*(lambda[t,r,t2,r2]/(abs(aTime[t]-aTime[t2]))
+mu[t,r,t2,r2]/(abs(dTime[t]-dTime[t2]))
+nu[t,r,t2,r2]/(abs(aTime[t]-dTime[t2])));

#--------------------------------------------------------------
# Constraints
#--------------------------------------------------------------
```

```
#xw  relations

subject to xw_relation_1{t1 in TRAINS, t2 in TRAINS, p1 in PLATFORMS,
p2 in PLATFORMS}: w[t1,p1,t2,p2]<=x[t1,p1];

subject to xw_relation_2{t1 in TRAINS, t2 in TRAINS, p1 in PLATFORMS,
p2 in PLATFORMS}: w[t1,p1,t2,p2]<=x[t2,p2];

subject to xw_relation_3{t1 in TRAINS, t2 in TRAINS, p1 in PLATFORMS,
p2 in PLATFORMS}: w[t1,p1,t2,p2]>=x[t1,p1]+x[t2,p2]-1;

#Each train is assigned to one and only one platform

subject to One_platform{t in TRAINS}:
sum{p in PLATFORMS}TP_comp[t,p]*x[t,p]=1;

# Relations between y, z and x

subject to xy_arrival{t in TRAINS, p in PLATFORMS}:
sum{r in SUB_ROUTES[p,aDir[t]]}y[t,r]=x[t,p];

subject to xz_departures{t in TRAINS, p in PLATFORMS}:
sum{r in SUB_ROUTES[p,dDir[t]]}z[t,r]=x[t,p];

# Platform compatibilty
subject to platform_comp{(t1,t2) in EDGES_X, p in PLATFORMS}:
x[t1,p]+x[t2,p]<=1;


# Route compatibility

subject to route_comp_inin{(t1,t2) in EDGES_Y, r1 in ROUTES,
r2 in ROUTES: Overlap[r1,r2]=1}: y[t1,r1]+y[t2,r2]<=1;

subject to route_comp_outout{(t1,t2) in EDGES_Z, r1 in ROUTES,
r2 in ROUTES: Overlap[r1,r2]=1}:z[t1,r1]+z[t2,r2]<=1;

subject to route_comp_inout{(t1,t2) in EDGES_YZ, r1 in ROUTES,
r2 in ROUTES: Overlap[r1,r2]=1}: y[t1,r1]+z[t2,r2]<=1;
```

```
# Variables h

subject to xh_relation{p in PLATFORMS, t in TRAINS}:
x[t,p] <= h[p];


# Variables lambda, mu and nu

subject to ylambda_relation_1{t1 in TRAINS, t2 in TRAINS,
r1 in ROUTES, r2 in ROUTES}: lambda[t1,r1,t2,r2]<=y[t1,r1];

subject to ylambda_relation_2{t1 in TRAINS, t2 in TRAINS,
r1 in ROUTES, r2 in ROUTES}: lambda[t1,r1,t2,r2]<=y[t2,r2];

subject to ylambda_relation_3{t1 in TRAINS, t2 in TRAINS,
r1 in ROUTES, r2 in ROUTES}: lambda[t1,r1,t2,r2]>=y[t1,r1]+y[t2,r2]-1;

subject to zmu_relation_1{t1 in TRAINS, t2 in TRAINS,
r1 in ROUTES, r2 in ROUTES}: mu[t1,r1,t2,r2]<=z[t1,r1];

subject to zmu_relation_2{t1 in TRAINS, t2 in TRAINS,
r1 in ROUTES, r2 in ROUTES}: mu[t1,r1,t2,r2]<=z[t2,r2];

subject to zmu_relation_3{t1 in TRAINS, t2 in TRAINS,
r1 in ROUTES, r2 in ROUTES}: mu[t1,r1,t2,r2]>=z[t1,r1]+z[t2,r2]-1;

subject to yznu_relation_1{t1 in TRAINS, t2 in TRAINS,
r1 in ROUTES, r2 in ROUTES}:nu[t1,r1,t2,r2]<=y[t1,r1];

subject to yznu_relation_2{t1 in TRAINS, t2 in TRAINS,
r1 in ROUTES, r2 in ROUTES}: nu[t1,r1,t2,r2]<=z[t2,r2];

subject to yznu_relation_3{t1 in TRAINS, t2 in TRAINS,
r1 in ROUTES, r2 in ROUTES}:nu[t1,r1,t2,r2]>=y[t1,r1]+z[t2,r2]-1;
```

A.3 Linköping specific data file

```
#--------------------------------------------------------------
# DATA FILE :  Linkoping station
#--------------------------------------------------------------


set PLATFORMS:= P1 P2 P3 P4 P5;

param Distance:=
P1 0
P2 1
P3 1
P4 2
P5 2;

set DIRECTIONS:= 1 ..9;


# 1 - D1 Södra stambanan, utgående tåg söderut
# 2 - D2 Södra stambanan, inkommande tåg söderifrån
# 3 - D3 Södra stambanan,utgående tåg norrut
# 4 - D4 Södra stambanan,inkommande tåg norrifrån

# 5 - D5 Stångådalsbanan/Tjustbanan både inkommande och utgående

# U1,U2,U3,U4 Uppställningplatser

param DistanceP2P
:P1 P2 P3 P4 P5:=
P1 0 1 1 1 1
P2 1 0 0 1 1
P3 1 0 0 1 1
P4 1 1 1 0 0
P5 1 1 1 0 0;


param:
ROUTES: Switches:=
P1D1 2
P1D3 4
```

```
P1D4 3
P1D5 2
P1U1 4
P1U2 3
P2D1 4
P2D2 2
P2D3 3
P2D4 5
P2D5 4
P2U1 6
P2U2 5
P2G 6
P3D1 6
P3D2 4
P3D3 4
P3D4 6
P3D5 5
P3U1 8
P3U2 7
P3U4 4
P4D1 8
P4D2 6
P4D3 5
P4D4 7
P4D5 6
P4U1 10
P4U2 9
P4U4 4
P5D1 9
P5D2 7
P5D3 5
P5D4 7
P5D5 6
P5U1 11
P5U2 10
P5U3 1
P5U4 4;

param Overlap
:P1D1 P1D3  P1D4 P1D5 P1U1 P1U2 P2D1  P2D2  P2D3 P2D4 P2D5 P2U1 P2U2
P3D1 P3D2 P3D3 P3D4 P3D5 P3U1 P3U2 P3U4 P4D1 P4D2 P4D3 P4D4 P4D5 P4U1
```

```
P4U2 P4U4 P5D1 P5D2 P5D3 P5D4 P5D5 P5U1 P5U2 P5U3 P5U4 :=
P1D1 1 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 1 0
0 0 0 1 1 0 0
P1D3  0 1 1 1 0 0 0 0 1 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P1D4 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0
0 1 1 0 0 0 0
P1D5 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0
0 1 1 0 0 0 0
P1U1 1 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 1 0
0 0 0 1 1 0 0
P1U2 1 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 0 1 0 0 0 0 1 1 0 1 0
0 0 0 1 1 0 0
P2D1  1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1
0 0 0 1 1 0 0
P2D2  0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1
0 0 0 1 1 0 0
P2D3 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P2D4 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P2D5 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P2U1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1
0 0 0 1 1 0 0
P2U2 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1
0 0 0 1 1 0 0
P3D1 1 0 0 0 1 1 1 1 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1
0 0 0 1 1 0 0
P3D2 0 0 0 0 0 0 1 1 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1
0 0 0 1 1 0 0
P3D3 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P3D4 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P3D5 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P3U1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1
0 0 0 1 1 0 0
P3U2 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 0 1 1
0 0 0 1 1 0 0
```

```
P3U4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 1
P4D1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 0
P4D2 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 0
P4D3 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P4D4 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P4D5 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P4U1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 0
P4U2 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 0
P4U4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 1
P5D1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 0
P5D2 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 0
P5D3 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P5D4 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P5D5 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0
1 1 1 0 0 0 0
P5U1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 0
P5U2 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1
0 0 0 1 1 0 0
P5U3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0
P5U4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 1;


set SUB_ROUTES[P1,1] := P1D1;
set SUB_ROUTES[P1,2] := ;
set SUB_ROUTES[P1,3] := P1D3;
```

```
set SUB_ROUTES[P1,4] := P1D4;
set SUB_ROUTES[P1,5] := P1D5;
set SUB_ROUTES[P1,6] := P1U1;
set SUB_ROUTES[P1,7] := P1U2;
set SUB_ROUTES[P1,8] := ;
set SUB_ROUTES[P1,9] := ;


set SUB_ROUTES[P2,1] := P2D1;
set SUB_ROUTES[P2,2] := P2D2;
set SUB_ROUTES[P2,3] := P2D3;
set SUB_ROUTES[P2,4] := P2D4;
set SUB_ROUTES[P2,5] := P2D5;
set SUB_ROUTES[P2,6] := P2U1;
set SUB_ROUTES[P2,7] := P2U2;
set SUB_ROUTES[P2,8] := ;
set SUB_ROUTES[P2,9] := ;

set SUB_ROUTES[P3,1] := P3D1;
set SUB_ROUTES[P3,2] := P3D2;
set SUB_ROUTES[P3,3] := P3D3;
set SUB_ROUTES[P3,4] := P3D4;
set SUB_ROUTES[P3,5] := P3D5;
set SUB_ROUTES[P3,6] := P3U1;
set SUB_ROUTES[P3,7] := P3U2;
set SUB_ROUTES[P3,8] := ;
set SUB_ROUTES[P3,9] := P3U4;

set SUB_ROUTES[P4,1] := P4D1;
set SUB_ROUTES[P4,2] := P4D2;
set SUB_ROUTES[P4,3] := P4D3;
set SUB_ROUTES[P4,4] := P4D4;
set SUB_ROUTES[P4,5] := P4D5;
set SUB_ROUTES[P4,6] := P4U1;
set SUB_ROUTES[P4,7] := P4U2;
set SUB_ROUTES[P4,8] := ;
set SUB_ROUTES[P4,9] := P4U4;

set SUB_ROUTES[P5,1] := P5D1;
set SUB_ROUTES[P5,2] := P5D2;
set SUB_ROUTES[P5,3] := P5D3;
```

```
set SUB_ROUTES[P5,4] := P5D4;
set SUB_ROUTES[P5,5] := P5D5;
set SUB_ROUTES[P5,6] := P5U1;
set SUB_ROUTES[P5,7] := P5U2;
set SUB_ROUTES[P5,8] := P5U3;
set SUB_ROUTES[P5,9] := P5U4;
```

Appendix B

Timetables

B.1 Timetable A

Train	τ_t^a	τ_t^d	d_t^a	d_t^d	Passengers	$c_{t,1}$	$c_{t,2}$	$c_{t,3}$	$c_{t,4}$	$c_{t,5}$
KP8559	78	83	U3	D5	144	1	1	1	1	1
KP8554	38	43	D5	U3	144	1	1	1	1	1
KP8587	62	67	U2	D5	144	1	1	1	1	1
KP8584	51	56	D5	U2	144	1	1	1	1	1
OT8727	15	16	D3	D2	240	1	1	1	1	1
OT8729	45	46	D3	D2	240	1	1	1	1	1
OT8731	75	76	D3	D2	240	1	1	1	1	1
OT8733	105	106	D3	D2	240	1	1	1	1	1
OT8726	14	15	D1	D4	240	1	1	1	1	1
OT8728	44	45	D1	D4	240	1	1	1	1	1
OT8730	74	75	D1	D4	240	1	1	1	1	1
OT8732	104	105	D1	D4	240	1	1	1	1	1
SJ527	1	2	D3	D2	309	1	1	1	0	0
ST3941	33	34	D3	D2	309	1	1	1	0	0
SJ529	57	58	D3	D2	309	1	1	1	0	0
SJ530	56	57	D1	D4	309	1	1	1	0	0
ST300	85	86	D1	D4	309	1	1	1	0	0
SJ532	112	113	D1	D4	309	1	1	1	0	0
IC10295	94	96	D3	D4	309	1	1	1	0	0
SJ2119	49	59	D3	D4	309	1	1	1	0	0
SJ2123	109	129	D3	D4	309	1	1	1	0	0

GT47795	4	4	D4	D1	0	1	1	1	1	1
GT42784	24	24	D4	D1	0	1	1	1	1	1
GT66248	28	28	D2	D3	0	1	1	1	1	1
GT4421	62	62	D4	D1	0	1	1	1	1	1
GT27783	70	70	D4	D1	0	1	1	1	1	1
GT44969	99	99	D4	D1	0	1	1	1	1	1

Timetable, based on the real timetable from April 1 2021, from 11:00 to 13:00. In the table, arrival and departure times are given in minutes after 11:00.

B.2 Timetable B

Train	τ_t^a	τ_t^d	d_t^a	d_t^d	Passengers	$c_{t,1}$	$c_{t,2}$	$c_{t,3}$	$c_{t,4}$	$c_{t,5}$
KP8555	76	81	U3	D5	144	1	1	1	1	1
KP8550	20	25	D5	U3	144	1	1	1	1	1
KP8552	102	107	D5	U3	144	1	1	1	1	1
KP8583	53	58	U3	D5	144	1	1	1	1	1
KP8580	27	32	D5	U3	144	1	1	1	1	1
OT8811	3	4	D4	D1	240	1	1	1	1	1
OT8711	15	16	D4	D1	240	1	1	1	1	1
OT8813	30	31	D4	D1	240	1	1	1	1	1
OT8713	45	46	D4	D1	240	1	1	1	1	1
OT8815	64	65	D4	D1	240	1	1	1	1	1
OT8715	75	76	D4	D1	240	1	1	1	1	1
OT8817	90	91	D4	D1	240	1	1	1	1	1
OT8717	105	106	D4	D1	240	1	1	1	1	1
OT8710	14	15	D2	D3	240	1	1	1	1	1
OT8712	29	30	D2	D3	240	1	1	1	1	1
OT8814	54	55	D2	U1	240	1	1	1	1	1
OT8800	64	65	D2	D3	240	1	1	1	1	1
OT8714	74	75	D2	D3	240	1	1	1	1	1
OT8816	94	95	D2	D3	240	1	1	1	1	1
OT8716	104	105	D2	D3	240	1	1	1	1	1
SJR2103	49	54	D4	U1	309	1	1	1	0	0
SJS521	56	58	D4	D1	309	1	1	1	0	0
SJS523	117	119	D4	D1	309	1	1	1	0	0

SJR2107	109	114	D4	U2	309	1	1	1	0	0
SJR2116	0	5	U1	D3	309	1	1	1	0	0
SJS522	57	59	D2	D3	309	1	1	1	0	0
SJR2120	64	69	U2	D3	309	1	1	1	0	0
GT66222	7	7	D2	D3	0	1	1	1	1	1
GT61993	22	22	D4	D1	0	1	1	1	1	1
GT4431	36	36	D4	D1	0	1	1	1	1	1
GT40290	86	86	D2	D3	0	1	1	1	1	1
GT60174	113	113	D2	D3	0	1	1	1	1	1

B.3 Timetable C

Train	τ_t^a	τ_t^d	d_t^a	d_t^d	Passengers	$c_{t,1}$	$c_{t,2}$	$c_{t,3}$	$c_{t,4}$	$c_{t,5}$
KP8555	76	81	U3	D5	144	1	1	1	1	1
KP8557	199	204	U3	D5	144	1	1	1	1	1
KP8559	317	323	U3	D5	144	1	1	1	1	1
KP8550	20	25	D5	U3	144	1	1	1	1	1
KP8552	102	107	D5	U3	144	1	1	1	1	1
KP8554	278	289	D5	U3	144	1	1	1	1	1
KP8583	53	58	U3	D5	144	1	1	1	1	1
KP8585	182	187	U3	D5	144	1	1	1	1	1
KP8587	302	307	U3	D5	144	1	1	1	1	1
KP8580	27	32	D5	U3	144	1	1	1	1	1
KP8582	140	145	D5	U3	144	1	1	1	1	1
KP8584	291	296	D5	U3	144	1	1	1	1	1
OT8811	3	4	D4	D1	240	1	1	1	1	1
OT8711	15	16	D4	D1	240	1	1	1	1	1
OT8813	30	31	D4	D1	240	1	1	1	1	1
OT8713	45	46	D4	D1	240	1	1	1	1	1
OT8815	64	65	D4	D1	240	1	1	1	1	1
OT8715	75	76	D4	D1	240	1	1	1	1	1
OT8817	90	91	D4	D1	240	1	1	1	1	1
OT8717	105	106	D4	D1	240	1	1	1	1	1
OT8719	135	136	D4	D1	240	1	1	1	1	1
OT8721	165	166	D4	D1	240	1	1	1	1	1
OT8723	195	196	D4	D1	240	1	1	1	1	1
OT8725	225	226	D4	D1	240	1	1	1	1	1

OT8727	255	256	D4	D1	240	1	1	1	1	1
OT8729	285	286	D4	D1	240	1	1	1	1	1
OT8731	315	316	D4	D1	240	1	1	1	1	1
OT8733	345	346	D4	D1	240	1	1	1	1	1
OT8710	14	15	D2	D3	240	1	1	1	1	1
OT8712	29	30	D2	D3	240	1	1	1	1	1
OT8814	54	55	D2	U1	240	1	1	1	1	1
OT8800	64	65	D2	D3	240	1	1	1	1	1
OT8714	74	75	D2	D3	240	1	1	1	1	1
OT8816	94	95	D2	D3	240	1	1	1	1	1
OT8716	104	105	D2	D3	240	1	1	1	1	1
OT8718	134	135	D2	D3	240	1	1	1	1	1
OT8720	164	165	D2	D3	240	1	1	1	1	1
OT8722	194	195	D2	D3	240	1	1	1	1	1
OT8724	224	225	D2	D3	240	1	1	1	1	1
OT8726	254	255	D2	D3	240	1	1	1	1	1
OT8728	284	285	D2	D3	240	1	1	1	1	1
OT8730	314	315	D2	D3	240	1	1	1	1	1
OT8732	344	345	D2	D3	240	1	1	1	1	1
SJR2103	49	54	D4	U1	309	1	1	1	0	0
SJS521	56	58	D4	D1	309	1	1	1	0	0
SJS523	117	119	D4	D1	309	1	1	1	0	0
SJR2107	109	114	D4	U2	309	1	1	1	0	0
SJR2116	0	5	U1	D3	309	1	1	1	0	0
SJS522	57	59	D2	D3	309	1	1	1	0	0
SJR2120	64	69	U2	D3	309	1	1	1	0	0
SJS525	175	178	D4	D1	309	1	1	1	0	0
SJS527	239	242	D4	D1	309	1	1	1	0	0
ST3941	271	274	D4	D1	309	1	1	1	0	0
SJS529	295	298	D4	D1	309	1	1	1	0	0
SJS524	114	117	D2	D3	309	1	1	1	0	0
SJS526	174	177	D2	D3	309	1	1	1	0	0
SJS528	232	235	D2	D3	309	1	1	1	0	0
SJS530	294	297	D2	D3	309	1	1	1	0	0
ST3940	323	326	D2	D3	309	1	1	1	0	0
SJS532	350	353	D2	D3	309	1	1	1	0	0
SJR2111	169	172	D4	U4	309	1	1	1	0	0
SJR2115	229	232	D4	U4	309	1	1	1	0	0

SJR2119	289	292	D4	U4	309	1	1	1	0	0
SJR2123	349	352	D4	U4	309	1	1	1	0	0
SJR2124	126	129	U4	D3	309	1	1	1	0	0
SJR2128	186	189	U4	D3	309	1	1	1	0	0
SJR2132	246	249	U4	D3	309	1	1	1	0	0
SJR2136	306	309	U4	D3	309	1	1	1	0	0
GT66222	7	7	D2	D3	0	1	1	1	1	1
GT61993	22	22	D4	D1	0	1	1	1	1	1
GT4431	36	36	D4	D1	0	1	1	1	1	1
GT40290	86	86	D2	D3	0	1	1	1	1	1
GT60174	113	113	D2	D3	0	1	1	1	1	1
GT47866	143	143	D4	D1	0	1	1	1	1	1
GT47867	145	145	D4	D1	0	1	1	1	1	1
GT48967	203	203	D4	D1	0	1	1	1	1	1
GT66239	218	218	D4	D1	0	1	1	1	1	1
GT47795	244	244	D4	D1	0	1	1	1	1	1
GT42783	264	264	D4	D1	0	1	1	1	1	1
GT66248	268	268	D2	D3	0	1	1	1	1	1
GT4421	302	302	D4	D1	0	1	1	1	1	1
GT27783	310	310	D4	D1	0	1	1	1	1	1
GT44969	339	339	D4	D1	0	1	1	1	1	1

B.4 Timetable D

Train	τ_t^a	τ_t^d	d_t^a	d_t^d	Passengers	$c_{t,1}$	$c_{t,2}$	$c_{t,3}$	$c_{t,4}$	$c_{t,5}$
KP8581	335	340	U3	D5	144	1	1	1	1	1
KP8583	473	478	U3	D5	144	1	1	1	1	1
KP8585	602	607	U3	D5	144	1	1	1	1	1
KP8587	722	727	U3	D5	144	1	1	1	1	1
KP8589	841	846	U3	D5	144	1	1	1	1	1
KP8591	962	967	U3	D5	144	1	1	1	1	1
KP8593	1087	1092	U3	D5	144	1	1	1	1	1
KP8595	1204	1209	U3	D5	144	1	1	1	1	1
KP8580	447	452	D5	U3	144	1	1	1	1	1
KP8582	560	565	D5	U3	144	1	1	1	1	1
KP8584	711	716	D5	U3	144	1	1	1	1	1
KP8586	831	836	D5	U3	144	1	1	1	1	1

KP8588	951	956	D5	U3	144	1	1	1	1	1
KP8590	1071	1076	D5	U3	144	1	1	1	1	1
KP8592	1191	1196	D5	U3	144	1	1	1	1	1
KP8594	1311	1316	D5	U3	144	1	1	1	1	1
KP8553	322	327	U3	D5	144	1	1	1	1	1
KP8555	496	501	U3	D5	144	1	1	1	1	1
KP8557	619	624	U3	D5	144	1	1	1	1	1
KP8559	718	723	U3	D5	144	1	1	1	1	1
KP8561	861	866	U3	D5	144	1	1	1	1	1
KP8563	980	985	U3	D5	144	1	1	1	1	1
KP8565	1101	1106	U3	D5	144	1	1	1	1	1
KP8567	1216	1221	U3	D5	144	1	1	1	1	1
KP8550	440	445	D5	U3	144	1	1	1	1	1
KP8552	522	527	D5	U3	144	1	1	1	1	1
KP8554	698	673	D5	U3	144	1	1	1	1	1
KP8556	818	823	D5	U3	144	1	1	1	1	1
KP8558	938	943	D5	U3	144	1	1	1	1	1
KP8560	1060	1065	D5	U3	144	1	1	1	1	1
KP8562	1180	1185	D5	U3	144	1	1	1	1	1
KP8564	1298	1303	D5	U3	144	1	1	1	1	1
OT8703	315	316	D4	D1	240	1	1	1	1	1
OT8705	345	346	D4	D1	240	1	1	1	1	1
OT8707	375	376	D4	D1	240	1	1	1	1	1
OT8809	387	388	D4	D1	240	1	1	1	1	1
OT8709	405	406	D4	D1	240	1	1	1	1	1
OT8811	423	424	D4	D1	240	1	1	1	1	1
OT8711	435	436	D4	D1	240	1	1	1	1	1
OT8813	450	451	D4	D1	240	1	1	1	1	1
OT8713	465	466	D4	D1	240	1	1	1	1	1
OT8815	484	485	D4	D1	240	1	1	1	1	1
OT8715	495	496	D4	D1	240	1	1	1	1	1
OT8817	510	511	D4	D1	240	1	1	1	1	1
OT8717	525	526	D4	D1	240	1	1	1	1	1
OT8719	555	556	D4	D1	240	1	1	1	1	1
OT8721	585	586	D4	D1	240	1	1	1	1	1
OT8723	615	616	D4	D1	240	1	1	1	1	1
OT8725	645	646	D4	D1	240	1	1	1	1	1
OT8727	675	676	D4	D1	240	1	1	1	1	1

OT8729	705	706	D4	D1	240	1	1	1	1	1
OT8731	735	736	D4	D1	240	1	1	1	1	1
OT8733	765	766	D4	D1	240	1	1	1	1	1
OT8735	795	796	D4	D1	240	1	1	1	1	1
OT8737	825	826	D4	D1	240	1	1	1	1	1
OT8739	855	856	D4	D1	240	1	1	1	1	1
OT8741	885	886	D4	D1	240	1	1	1	1	1
OT8743	915	916	D4	D1	240	1	1	1	1	1
OT8845	930	931	D4	D1	240	1	1	1	1	1
OT8745	945	946	D4	D1	240	1	1	1	1	1
OT8847	965	966	D4	D1	240	1	1	1	1	1
OT8747	975	976	D4	D1	240	1	1	1	1	1
OT8849	990	991	D4	D1	240	1	1	1	1	1
OT8749	1005	1006	D4	D1	240	1	1	1	1	1
OT8851	1028	1029	D4	D1	240	1	1	1	1	1
OT8751	1035	1036	D4	D1	240	1	1	1	1	1
OT8853	1050	1051	D4	D1	240	1	1	1	1	1
OT8753	1065	1066	D4	D1	240	1	1	1	1	1
OT8855	1083	1084	D4	D1	240	1	1	1	1	1
OT8755	1098	1099	D4	D1	240	1	1	1	1	1
OT8757	1128	1129	D4	D1	240	1	1	1	1	1
OT8759	1155	1156	D4	D1	240	1	1	1	1	1
OT8761	1185	1186	D4	D1	240	1	1	1	1	1
OT8863	1215	1216	D4	D1	240	1	1	1	1	1
OT8765	1245	1246	D4	D1	240	1	1	1	1	1
OT8867	1275	1276	D4	D1	240	1	1	1	1	1
OT8769	1305	1306	D4	D1	240	1	1	1	1	1
OT8871	1335	1336	D4	D1	240	1	1	1	1	1
OT8773	1365	1366	D4	D1	240	1	1	1	1	1
OT8775	1395	1396	D4	D1	240	1	1	1	1	1
OT8777	1425	1426	D4	D1	240	1	1	1	1	1
OT8702	314	315	D2	D3	240	1	1	1	1	1
OT8704	344	345	D2	D3	240	1	1	1	1	1
OT8706	374	375	D2	D3	240	1	1	1	1	1
OT8808	389	390	D2	D3	240	1	1	1	1	1
OT8708	404	405	D2	D3	240	1	1	1	1	1
OT8710	434	435	D2	D3	240	1	1	1	1	1
OT8812	449	450	D2	D3	240	1	1	1	1	1

OT8712	464	465	D2	D3	240	1	1	1	1	1
OT8814	474	475	D2	U1	240	1	1	1	1	1
OT8800	484	485	D2	D3	240	1	1	1	1	1
OT8714	494	495	D2	D3	240	1	1	1	1	1
OT8816	509	510	D2	D3	240	1	1	1	1	1
OT8716	524	525	D2	D3	240	1	1	1	1	1
OT8718	554	555	D2	D3	240	1	1	1	1	1
OT8720	584	585	D2	D3	240	1	1	1	1	1
OT8722	614	615	D2	D3	240	1	1	1	1	1
OT8724	644	645	D2	D3	240	1	1	1	1	1
OT8726	674	675	D2	D3	240	1	1	1	1	1
OT8728	704	705	D2	D3	240	1	1	1	1	1
OT8730	734	735	D2	D3	240	1	1	1	1	1
OT8732	764	765	D2	D3	240	1	1	1	1	1
OT8734	794	795	D2	D3	240	1	1	1	1	1
OT8736	824	825	D2	D3	240	1	1	1	1	1
OT8738	854	855	D2	D3	240	1	1	1	1	1
OT8740	884	885	D2	D3	240	1	1	1	1	1
OT8742	914	915	D2	D3	240	1	1	1	1	1
OT8744	944	945	D2	D3	240	1	1	1	1	1
OT8846	954	955	D2	D3	240	1	1	1	1	1
OT8746	974	975	D2	D3	240	1	1	1	1	1
OT8848	989	990	D2	D3	240	1	1	1	1	1
OT8748	1004	1005	D2	D3	240	1	1	1	1	1
OT8850	1014	1015	D2	D3	240	1	1	1	1	1
OT8750	1034	1035	D2	D3	240	1	1	1	1	1
OT8852	1054	1055	D2	D3	240	1	1	1	1	1
OT8752	1064	1065	D2	D3	240	1	1	1	1	1
OT8754	1094	1095	D2	D3	240	1	1	1	1	1
OT8756	1124	1125	D2	D3	240	1	1	1	1	1
OT8758	1154	1155	D2	D3	240	1	1	1	1	1
OT8760	1184	1185	D2	D3	240	1	1	1	1	1
OT8762	1214	1215	D2	D3	240	1	1	1	1	1
OT8764	1244	1245	D2	D3	240	1	1	1	1	1
OT8766	1274	1275	D2	D3	240	1	1	1	1	1
OT8768	1304	1305	D2	D3	240	1	1	1	1	1
OT8770	1334	1335	D2	D3	240	1	1	1	1	1
OT8772	1364	1365	D2	D3	240	1	1	1	1	1

OT8774	1394	1395	D2	D3	240	1	1	1	1	1
OT8776	1424	1425	D2	D3	240	1	1	1	1	1
SJS519	415	418	D4	D1	309	1	1	1	0	0
SJS521	475	478	D4	D1	309	1	1	1	0	0
SJS523	536	539	D4	D1	309	1	1	1	0	0
SJS525	596	599	D4	D1	309	1	1	1	0	0
SJS527	659	662	D4	D1	309	1	1	1	0	0
ST3941	691	694	D4	D1	309	1	1	1	0	0
SJS529	715	718	D4	D1	309	1	1	1	0	0
SJS531	779	782	D4	D1	309	1	1	1	0	0
SJS533	835	838	D4	D1	309	1	1	1	0	0
ST3931	858	861	D4	D1	309	1	1	1	0	0
SJS535	899	902	D4	D1	309	1	1	1	0	0
SJS537	955	958	D4	D1	309	1	1	1	0	0
SJIC207	1037	1040	D4	D1	309	1	1	1	0	0
SJS539	1016	1019	D4	D1	309	1	1	1	0	0
SJS541	1073	1076	D4	D1	309	1	1	1	0	0
ST3943	1088	1091	D4	D1	309	1	1	1	0	0
SJS513	1104	1107	D4	D1	309	1	1	1	0	0
SJS543	1136	1139	D4	D1	309	1	1	1	0	0
SJS545	1195	1198	D4	D1	309	1	1	1	0	0
ST3935	1223	1226	D4	D1	309	1	1	1	0	0
SJS547	1259	1262	D4	D1	309	1	1	1	0	0
SJS549	1312	1315	D4	D1	309	1	1	1	0	0
SJS520	411	414	D2	D3	309	1	1	1	0	0
SJS512	454	457	D2	D3	309	1	1	1	0	0
SJS522	477	480	D2	D3	309	1	1	1	0	0
SJS524	537	540	D2	D3	309	1	1	1	0	0
SJS526	597	600	D2	D3	309	1	1	1	0	0
SJS528	655	658	D2	D3	309	1	1	1	0	0
SJS530	717	720	D2	D3	309	1	1	1	0	0
ST3940	746	749	D2	D3	309	1	1	1	0	0
SJS532	773	776	D2	D3	309	1	1	1	0	0
SJS534	837	840	D2	D3	309	1	1	1	0	0
SJS536	893	896	D2	D3	309	1	1	1	0	0
ST3930	935	938	D2	D3	309	1	1	1	0	0
SJS538	957	960	D2	D3	309	1	1	1	0	0
SJS540	1017	1020	D2	D3	309	1	1	1	0	0

SJS542	1077	1080	D2	D3	309	1	1	1	0	0
SJS544	1137	1140	D2	D3	309	1	1	1	0	0
ST3942	1158	1161	D2	D3	309	1	1	1	0	0
ST3900	1236	1239	D2	D3	309	1	1	1	0	0
SJIC204	1204	1207	D2	D3	309	1	1	1	0	0
SJS546	1197	1200	D2	D3	309	1	1	1	0	0
SJS548	1258	1261	D2	D3	309	1	1	1	0	0
SJS506	1280	1283	D2	D3	309	1	1	1	0	0
ST3934	1295	1298	D2	D3	309	1	1	1	0	0
SJS550	1318	1321	D2	D3	309	1	1	1	0	0
SJS508	1397	1400	D2	D3	309	1	1	1	0	0
SJR2103	469	472	D4	U4	309	1	1	1	0	0
SJR2107	529	532	D4	U4	309	1	1	1	0	0
SJR2111	589	592	D4	U4	309	1	1	1	0	0
SJR2115	649	652	D4	U4	309	1	1	1	0	0
SJR2219	709	712	D4	U4	309	1	1	1	0	0
SJR2123	769	772	D4	U4	309	1	1	1	0	0
SJR2127	829	832	D4	U4	309	1	1	1	0	0
SJR2131	889	892	D4	U4	309	1	1	1	0	0
SJR2135	949	952	D4	U4	309	1	1	1	0	0
SJR2139	1009	1012	D4	U4	309	1	1	1	0	0
SJR2143	1069	1072	D4	U4	309	1	1	1	0	0
SJR2151	1189	1192	D4	U4	309	1	1	1	0	0
SJR2155	1249	1252	D4	U4	309	1	1	1	0	0
SJR2159	1309	1312	D4	U4	309	1	1	1	0	0
SJR2108	306	309	U4	D3	309	1	1	1	0	0
SJR2112	366	369	U4	D3	309	1	1	1	0	0
SJR2116	422	425	U4	D3	309	1	1	1	0	0
SJR2120	486	489	U4	D3	309	1	1	1	0	0
SJR2124	546	549	U4	D3	309	1	1	1	0	0
SJR2128	606	609	U4	D3	309	1	1	1	0	0
SJR2132	666	669	U4	D3	309	1	1	1	0	0
SJR2136	726	729	U4	D3	309	1	1	1	0	0
SJR2140	786	789	U4	D3	309	1	1	1	0	0
SJR2144	839	842	U4	D3	309	1	1	1	0	0
SJR2148	906	909	U4	D3	309	1	1	1	0	0
SJR2152	965	968	U4	D3	309	1	1	1	0	0
SJR2156	1026	1029	U4	D3	309	1	1	1	0	0

SJR2160	1085	1088	U4	D3	309	1	1	1	0	0
SJR2164	1146	1149	U4	D3	309	1	1	1	0	0
SJR2168	1207	1210	U4	D3	309	1	1	1	0	0
GT15184	10	10	D2	D3	0	1	1	1	1	1
GT42230	66	66	D2	D3	0	1	1	1	1	1
GT4411	71	71	D4	D1	0	1	1	1	1	1
GT42701	109	109	D4	D1	0	1	1	1	1	1
GT15178	161	161	D2	D3	0	1	1	1	1	1
GT40239	188	188	D4	D1	0	1	1	1	1	1
GT9848	225	225	D2	D3	0	1	1	1	1	1
GT15149	235	235	D4	D1	0	1	1	1	1	1
GT42778	253	253	D2	D3	0	1	1	1	1	1
GT15145	256	256	D4	D1	0	1	1	1	1	1
GT15143	278	278	D4	D1	0	1	1	1	1	1
GT62036	300	300	D2	D3	0	1	1	1	1	1
GT44232	308	308	D2	D3	0	1	1	1	1	1
GT48034	335	335	D2	D3	0	1	1	1	1	1
GT44908	354	354	D2	D3	0	1	1	1	1	1
GT45300	367	367	D2	D3	0	1	1	1	1	1
GT62195	412	412	D4	D1	0	1	1	1	1	1
GT66222	427	427	D2	D3	0	1	1	1	1	1
GT61993	442	442	D4	D1	0	1	1	1	1	1
GT4431	456	456	D4	D1	0	1	1	1	1	1
GT40290	506	506	D2	D3	0	1	1	1	1	1
GT60174	533	533	D2	D3	0	1	1	1	1	1
GT47866	563	563	D4	D1	0	1	1	1	1	1
GT47867	565	565	D4	D1	0	1	1	1	1	1
GT48967	623	623	D4	D1	0	1	1	1	1	1
GT66239	638	638	D4	D1	0	1	1	1	1	1
GT47795	664	664	D4	D1	0	1	1	1	1	1
GT42783	684	684	D4	D1	0	1	1	1	1	1
GT66248	688	688	D2	D3	0	1	1	1	1	1
GT4421	722	722	D4	D1	0	1	1	1	1	1
GT27783	730	730	D4	D1	0	1	1	1	1	1
GT44969	759	759	D4	D1	0	1	1	1	1	1
GT15219	802	802	D4	D1	0	1	1	1	1	1
GT42123	811	811	D4	D1	0	1	1	1	1	1
GT66943	815	815	D4	D1	0	1	1	1	1	1

GT84954	849	849	D2	D3	0	1	1	1	1	1
GT66266	868	868	D2	D3	0	1	1	1	1	1
GT66265	871	871	D2	D3	0	1	1	1	1	1
GT40291	940	940	D4	D1	0	1	1	1	1	1
GT66270	980	980	D2	D3	0	1	1	1	1	1
GT44259	982	972	D2	D3	0	1	1	1	1	1
GT15190	1103	1103	D2	D3	0	1	1	1	1	1
GT44289	1164	1164	D4	D1	0	1	1	1	1	1
GT66423	1202	1202	D4	D1	0	1	1	1	1	1
GT40292	1225	1225	D2	D3	0	1	1	1	1	1
GT27328	1261	1261	D2	D3	0	1	1	1	1	1
GT44968	1262	1262	D2	D3	0	1	1	1	1	1
GT42702	1280	1280	D2	D3	0	1	1	1	1	1
GT44907	1291	1291	D4	D1	0	1	1	1	1	1
GT66258	1325	1325	D2	D3	0	1	1	1	1	1
GT66231	1327	1327	D4	D1	0	1	1	1	1	1
GT61992	1342	1342	D2	D3	0	1	1	1	1	1
GT44977	1344	1344	D4	D1	0	1	1	1	1	1
GT66216	1347	1347	D2	D3	0	1	1	1	1	1
GT44451	1350	1350	D4	D1	0	1	1	1	1	1
GT4342	1370	1370	D2	D3	0	1	1	1	1	1
GT44101	1380	1380	D4	D1	0	1	1	1	1	1
GT66260	1384	1384	D2	D3	0	1	1	1	1	1
GT44228	1415	1415	D2	D3	0	1	1	1	1	1
GT26505	1439	1439	D4	D1	0	1	1	1	1	1

Bibliography

[1] Emine Akyol, Josef Volek, Tugba Sarac, and Josef Bulicek. A new scheduling approach to train platforming problem. *Perner's Contacts*, 12, 2017.

[2] Nikola Besinovic and Rob M. P. Goverde. Stable and robust train routing in station areas with balanced infrastructure capacity occupation. *Public transportation*, 11(1):211–236, 2019.

[3] John A Bondy and Uppaluri Siva Ramachandra Murty. *Graph Theory*. Springer, 2008.

[4] Alberto Caprara, Laura Galli, and Paolo Toth. Solution of the train platforming problem. *Transportation science*, 45(2):246–257, 2011.

[5] Malachy Carey and Sinead Carville. Scheduling and platforming trains at busy complex stations. *Transportation research parta a*, 37(3):195–224, 2003.

[6] Alberto Carprara, Leo Kroon, Michele Monaci, Marc Peeters, and Paolo Toth. Passenger railway optimization. In Cynthia Barnhart and Gilbert Laporte, editors, *Handbooks in Operations Research and Management Science*, chapter 3, pages 129–187. Elsevier, 2007.

[7] Partha Chakroborty and Durgesh Vikram. Optimum assignment of trains to platforms under partial schedule compliance. *Transportation Research Part B: Methodological*, 42(2):169–184, 2008.

[8] Dorotea De Luca Cardillo and Nicola Mione. k l-list tau colouring of graphs. *European journal of Operational research*, 106(1):160–164, 1998.

[9] Reinhard Diestel. *Graph Theory*. Springer, 2010.

[10] H. Ding, A. Lim, B. Rodrigues, and Y. Zhu. The over-constrained airport gate assignment problem. *Computers and Operations Research*, 32(1):1867–1880, 2005.

[11] Kaj Holmberg. *Optimering.* Liber, 2018.

[12] Samtrafiken i Sverige AB. Resrobot. `https://tagtidtabeller.resrobot.se/`, 2021.

[13] Leo G Kroon, H. Edwin Romeijn, and Peter J Zwaneveld. Routing trains through railway stations: complexity issues. *European journal of Operational research*, 98(3):485–498, 1997.

[14] Leonardo Lamorgese and Carlo Mannino. An exact decmoposition approach for the real-time train dispatching problem. *Operations Research*, 63(1):48–64, 2015.

[15] Terese Lindberg, Fredrik Johansson, and Anders Peterson. An integer optimization model for allocation of bus lines to the stops of a bus terminal. 2021.

[16] Jan Lungdgren, Mikael Rönnqvist, and Peter Värbrand. *Optimeringslära.* Studentlitteratur, 2010.

[17] Sam Olsson. Optimization model for selection of switches at railway stations. Master's thesis, Linköping University, June 2021.

[18] Matthew E. H. Petering, Mojtaba Heytar, and Dietrich R. Bergmann. Mixed-integer programming for railway capacity analysis and cyclic, combined train timetabling and platforming. *Transportation Science*, 50, 2016.

[19] Railit. Railit tracker. `https://tracker.railit.se/#/login`, 2021.

[20] Trafikverket. Asek, analysmetod och samhällsekonomiska kalkylvärden. -, -, 2020.

[21] Trafikverket. Underhåll av väg och järnväg. https://www.trafikverket.se/resa-och-trafik/underhall-av-vag-och-jarnvag/, May 2021.

[22] Hanxiao Wang and Yue Yixiang. Optimizing train-platforming plans at arrival-departure tracks of high-speed railway stations. 2020.

[23] Qiongfang Zeng, Yinggul Zhang, and Dingyou Lei. An experimental analysis on dispatching rules for the train platforming problem in busy complex passenger stations. *Promet traffic and transportation*, 29, 2017.

[24] Peter J Zwaneveld, Leo G. Kroon, H. Edwin Romeijn, Marc Salomon, Stéphanie Dauzère-Pérès, Stan P.M. van Hoesel, and Harrie W Ambergen. Routing trains through railway stations: Model formulation and algorithms. *Transportation science*, 30(3):181–194, 1996.

[25] Peter J Zwaneveld, Leo G Kroon, and Stan P.M. van Hoesel. Routing trains through a railway station based on a node packing model. *European journal of Operational research*, 128(1):14–33, 2001.