Network Design for Railway Infrastructure by means of Linear Programming

Von der Fakultät für Bauingenieurwesen der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Jacob Christian Spönemann

Berichter: Universitätsprofessor Dr.-Ing. Dirk Vallée
Universitätsprofessor Dr. Ir. Arie M.C.A. Koster
Universitätsprofessor Dr. Berthold Vöcking

Tag der mündlichen Prüfung: 29.04.2013

Abstract

The network design problem for railway infrastructure NDRI aims to find a network of railway infrastructure which meets given traffic demands at the lowest possible design costs. This problem, for example, comes up in long-term infrastructure planning processes.

Railway infrastructure is represented by a network consisting of nodes and arcs. Nodes represent large stations, arcs lines connecting the stations. Traffic demand is represented by traffic flows consisting of train counts, source and sink nodes. Stations are assumed to be equipped with unbounded capacity to reduce complexity, so design issues are the network topology and the capacity of lines.

Using macroscopic models for infrastructure and operation and a timetable independent model for the capacity consumption of traffic flows, NDRI is modeled as a non-linear multicommodity flow problem on a complete multi-graph. The model is transformed to a mixed integer linear programming problem, called NDRI-MIP, using configurations similar to cutting patterns used in the widely known cutting stock problem.

Besides NDRI-MIP, another optimization model based on worst-case timetables, called NDRI-MIP$^{\mathrm{wc}}$, is introduced. A network designed by these means provides an upper bound on the infrastructure needed to satisfy the given traffic demand.

Both models use path flows instead of arc flows. This provides the opportunity to restrict the sets of routes, which are available to traffic flows, to sets which are reasonable in relation to practice. To improve the performance of the solution process some valid inequalities are presented and a branch-and-price approach for NDRI-MIP is introduced.

Largest instance which can be solved to optimality within a time frame of 24 hours consists of 25 nodes. The corresponding model is composed of 65,480 binary variables, 868 continuous variables, and 3,427 constraints. For optimality gaps up to 5% largest solvable instance consists of 40 nodes. It is composed of 182,130 binary variables, 4,976 continuous variables, and 9,071 constraints.

Contents

List of Variables

AC	Acquisition costs of an infrastructure element [€], page 21.
AC_{ballast}	Acquisition costs of ballast for one meter of track [$\frac{€}{\text{m}}$], page 125.
AC_{rail}	Acquisition costs of rails for one meter of track [$\frac{€}{\text{m}}$], page 125.
AC_{signal}	Acquisition costs of a signalling unit [€], page 125.
AC_{sleeper}	Acquisition costs of concrete sleepers for one meter of track [$\frac{€}{\text{m}}$], page 125.
AC_{switch}	Acquisition costs of a switch [€], page 125.
A	Set of arcs representing lines between stations, $A \subseteq N \times N$, page 13.
$A^{\text{m}}_{\text{in},n}$	Set of ingoing multi-arcs of node $n \in N$, page 52.
$A^{\text{m}}_{f_{\text{in},n}}$	Set of ingoing multi-arcs, which are available to the set of traffic flows $f_{\text{in},n}$, page 53.
A^{m}	Set of multi-arcs, page 41.
A^{opt}	Arcs of the optimal network graph, which is the solution of NDRI, page 34.
$A^{\text{m}}_{\text{out},n}$	Set of outgoing multi-arcs of node $n \in N$, page 52.
$A^{\text{m}}_{f_{\text{out},n}}$	Set of outgoing multi-arcs, which are available to the set of traffic flows $f_{\text{out},n}$, page 53.
cap_{ij}	Capacity of an arc $(i,j) \in A$, page 20.
cap_i	Capacity of a node $i \in N$, page 14.
$cc_{\text{ho},ij}$	Capacity consumption of homogeneous operation ($\lvert T \rvert = 1$) on arc $(i,j) \in A$, page 27.
$cc_{\text{td},ij}$	Total timetable dependent capacity consumption of all trains routed via the line $(i,j) \in A$, page 26.

$cc_{\mathrm{ti},ij}$	Expected total timetable independent capacity consumption of all trains routed via the line $(i, j) \in A$, page 25.
c_{ijs}	Configuration of a multi-arc $(i, j, s) \in A^{\mathrm{m}}$. Using a tuple of train counts of different train types c_{ijs} shows one way, how to fully exploit the capacity of (i, j, s), page 45.
$c_{t,ijs}$	Element of configuration c_{ijs}, page 45.
$cost_{i,j}$	Cost of an arc $(i, j) \in A$, page 20.
$cost_{ij,\mathrm{min}}$	Minimum costs of an arc (i, j), independent of the multi-arc which is selected, page 42.
$cost_i$	Cost of a node $i \in N$, page 14.
$\mathcal{C}'$	The pool of variables which is available for branch-and-price used to solve NDRI-MIP, page 64.
$\mathcal{C}_{ijs,\mathrm{all}}$	Set of all configurations of multi-arc (i, j, s), page 45.
$\mathcal{C}_{ijs}$	Set of all maximal configurations of multi-arc (i, j, s), page 45.
$\mathcal{C}$	Set of all *config*-variables of a NDRI instance, page 63.
$\mathcal{C}^{\mathrm{start}}$	Set of all *start*-configurations, cf. $\mathcal{C}_{ijs}^{\mathrm{start}}$, page 64.
$\mathcal{C}_{ijs}^{\mathrm{start}}$	Set of *start*-configurations of multi-arc (i, j, s) of NDRI-MIP, page 63.
d_t	Deceleration rate of a train of type t [km/h], page 22.
δ_{ij}	Indicator function. It indicates whether arc (i, j) is element of the path, which is passed to the function, page 38.
de_τ	Destination station of a traffic flow $\tau \in TD$, page 22.
D_{ij}	Distance between cities i and j, used in Lill's law of travel, page 79.
$\mathbf{E}$	$\mathbf{E}[X]$ denotes the expected value of random variable X, page 25.
$F_{t,ij}$	Traffic flow on arc (i, j) only consisting of trains of type $t \in T$, page 42.

f	Flow function. Returns the traffic flow which is routed via the path which is passed to f. Only used for homogenous traffic flows, page 38.
f_t	Train type dependent flow function. Returns the traffic flow of type t which is routed via the path which is passed to f, page 42.
G	Graph representing a network of railway infrastructure, page 13.
G^{c}	Complete network graph, page 41.
G^{opt}	Optimal network graph, solution of NDRI, page 34.
LCC_{ballast}	Life-cycle costs of ballast for one meter of track in $[\frac{€}{\mathrm{m \cdot a}}]$, page 125.
$LCC_{\mathrm{def},ij}$	Default life-cycle costs of an arc $(i,j) \in A$ without any overtaking station $[\frac{€}{\mathrm{a}}]$, page 21.
LCC_{pt}	Life-cycle costs of infrastructure elements required for a passing track $[\frac{€}{\mathrm{a}}]$, page 21.
LCC_{rail}	Rail life-cycle costs for one meter of track in $[\frac{€}{\mathrm{m \cdot a}}]$, page 125.
LCC_{signal}	Life-cycle costs of a signalling unit a block section has to be equipped with $[\frac{€}{\mathrm{a}}]$, page 123.
LCC_{sleeper}	Life-cycle costs of concrete sleepers for one meter of track in $[\frac{€}{\mathrm{m \cdot a}}]$, page 125.
LCC_{switch}	Life-cycle costs of a switch in $[\frac{€}{\mathrm{a}}]$, page 124.
LCC_{track}	Life-cycle costs per meter track $[\frac{€}{\mathrm{m \cdot a}}]$, page 123.
$l_{\mathrm{l},ij}$	Length of an arc $(i,j) \in A$, page 16.
$l_{\mathrm{bk},ij}$	Length of a block section [m], page 16.
$l_{\mathrm{os},ij}$	Average length of an overtaking section [km], page 16.
$l_{\mathrm{ps},ij}$	Length of the presignalling distance [m], page 16.
$l_{\mathrm{so},ij}$	Length of the safety overlap [m], page 16.
$l_{\mathrm{st},i}$	Length of a station block [m], page 15.
$l_{\mathrm{tr},t}$	Length of a train of type t [km/h], page 22.
L	Lifespan of an infrastructure element [a], page 21.

L_{ballast}	Lifespan of ballast [a], page 125.
L_{rail}	Lifespan of rails [a], page 125.
L_{signal}	Lifespan of a signalling unit [a], page 125.
L_{sleeper}	Lifespan of concrete sleepers [a], page 125.
L_{switch}	Lifespan of a switch [a], page 125.
lf_t	Load factor of train type t, page 78.
max_{cc}	Maximum of consumable arc capacity, in terms of available occupation time [min], (multi-)arc independent, page 24.
$max_{\text{cc},ij}$	Maximum of the consumable capacity of an arc (i,j), in terms of available occupation time [min], page 23.
$max_{\text{cc},ijs}$	Maximum of the consumable capacity of an multi-arc (i,j,s), in terms of available occupation time [min], cf. $max_{\text{cc},ij}$, page 69.
max_t	Maximum train count of trains of type t routable via an arc. Homogeneous operation is assumed for the derivation of that value, page 27.
$max_{t,ijs}$	Maximum train count of trains of type t routable via arc (i,j,s), cf. $max_{t,ij}$, page 45.
$max_{t,ij}$	Maximum train count of trains of type t routable via arc (i,j). Homogeneous operation is assumed for the derivation of that value, page 27.
MC	Maintenance costs of an infrastructure element $[\frac{€}{\text{a}}]$, page 21.
MC_{ballast}	Maintenance costs of ballast for one meter of track $[\frac{€}{\text{m}\cdot\text{a}}]$, page 125.
MC_{rail}	Maintenance costs of rails for one meter of track $[\frac{€}{\text{m}\cdot\text{a}}]$, page 125.
MC_{signal}	Maintenance costs of a signalling unit $[\frac{€}{\text{a}}]$, page 125.
MC_{sleeper}	Maintenance costs of concrete sleepers for one meter of track $[\frac{€}{\text{m}\cdot\text{a}}]$, page 125.
MC_{switch}	Maintenance costs of a switch $[\frac{€}{\text{a}}]$, page 125.

N	Set of nodes representing stations of the network of railway infrastructure, page 13.
nt_{ij}	Total number of trains on arc $(i, j) \in A$, page 25.
$n_{\mathrm{os},ij}$	Average number of overtaking stations, page 16.
$n_{\mathrm{tk},ij}$	Number of tracks of the open line, page 16.
$n_{t,ij}$	Number of trains of type t on arc $(i, j) \in A$, page 25.
$n_{t,\tau}$	Number of trains of type t of a traffic flow $\tau \in TD$, page 23.
P_i	Number of inhabitants of city i, used in Lill's law of travel, page 79.
pos_i	Position of a node, page 15.
$\mathbf{P}$	$\mathbf{P}[X = i]$ denotes the probability that value i is outcome of random variable X, page 24.
$p_{tu,ij}$	The probability of the occurrence of a succession of trains of types t and u competing for the capacity of arc (line) $(i, j) \in A$, page 24.
$\mathcal{P}^\tau$	Set of all directed *st*-*de*-paths available to traffic flow $\tau = (st_\tau, de_\tau, tr_\tau)$, page 38.
S_{ij}	Set of available stages of extension, page 16.
$sv_{\tau,t}$	Share of total traffic volume tv_t type t, which is assigned to traffic flow τ, page 79.
st_τ	Start station of a traffic flow $\tau \in TD$, page 22.
t_{op}	Observation period, page 22.
tv_t	Total traffic volume of train type t, i. e volume sold by corresponding business unit DB Mobility Logistics AG in 2009, page 78.
TD	Traffic demand represented by a set of traffic flows, page 23.
τ	Traffic flow, page 22.
tr_τ	Set of train counts of a traffic flow $\tau \in TD$, page 22.
$t_{\mathrm{t},ij}$	Set of different train types allowed on arc $(i, j) \in A$, page 16.
T	Set of train types, page 22.

T_{s}	Random variable modeling the service time. Random variates are corresponding minimum headway times, page 24.
v_t	Operating speed of train type t [km/h], page 22.
v_{ij}	Traffic volume between cities i and j, used in Lill's law of travel, page 79.
$v_{st,de,t}$	Traffic volume of train type t between stations st and de, page 79.
w_{ij}	Binary decision variable indicating the mix ratio of the traffic flow on arc (i,j), i.e. which train type (of two available train types) outnumbers the other, page 48.
x_{ijs}^{used}	Binary variable indicating whether multi-arc $(i,j,s) \in A^{\mathrm{m}}$ is included into the network graph, page 41.
x_{ij}^{used}	Binary variable indicating whether one multi-arc (i,j,s) of arc (i,j) has to be included into the network graph, page 40.
X_k	Random variable modeling the type of the kth train in the random succession of trains, page 24.
$Z_{tu,ij}$	Matrix of minimum headway times of arc $(i,j) \in A$, page 20.
$\bar{z}_{tu,ij}$	Mean minimum headway time of arc $(i,j) \in A$, page 25.
$z_{tu,ij}$	Minimum headway time of the succession of trains of types t and u competing for the capacity of arc $(i,j) \in A$, page 24.

1 Introduction

The introductory chapter starts with a motivation followed by an initial description of the network design problem for railway infrastructure. The chapter ends with a survey on related work and an introduction into the structure of this work.

1.1 Motivation

The problem to design a network of railway infrastructure is, for example, present in the strategic long-term infrastructure planning process done by railway infrastructure managers. In this regard, long-term means planning horizons of 10 up to 20 years. Objective of this planning process is the adaptation of railway infrastructure to future requirements and infrastructure manager's objectives, respectively. Since railway infrastructure managers offer available railway capacity to train operating companies, one of such future requirements could be to manage a higher number of train paths[1].

Ross [Ros01] divides the planning process into five phases. Essential aim of the first phase is the determination of criteria to be able to evaluate different infrastructure measures. These criteria can be derived from corporate goals like pushing up profits. This could be achieved, for example, by upgrading the infrastructure to sell more train paths or by dismantling of infrastructure capacity surpluses which reduces maintenance cost. Phases three, four, and five are dealing with the determination of implications of the planning process, the (monetary) evaluation of action alternatives, and the decision process. These phases are not discussed further here. Missing second phase is the *design phase* which is the focus of this work.

Input of this phase are the objectives of the precedent phase, output are new infrastructure designs fulfilling those objectives. The phase consists of two steps: step one is to route estimated future traffic flows (e.g. derived from forecasts for train path requests)

[1] A train path is that part of the capacity of the railway infrastructure which is necessary to schedule or run a train with a requested speed profile [HP08].

on the existing infrastructure and to identify bottlenecks or capacity surpluses. In step two the infrastructure has to be redesigned to meet those future traffic demands.

Network redesign is mainly based upon infrastructure manager's experience and is, as stated by Ross [Ros01], a *creative* process. Furthermore, the redesign is often restricted to local areas of the infrastructure network. This might lead to solutions which are not optimal with respect to the entire network. It is desirable to improve such a design practice and to achieve provable optimal design decisions. Why?

In general, railway infrastructure and transport infrastructure, respectively, have got following special characteristics [Abe72]. Due to its network structure, investments in transport infrastructure are indivisible[2] which implies high level capital intensity for such investments. In addition to that, railway infrastructure is, to a large degree, immovable and very long lasting. This is why provable optimal design decisions are desirable when investing in railway infrastructure as consequence of a long-term planning process.

This work presents an approach how to optimally solve such network design problem for railway infrastructure by considering it as a linear optimization problem.

1.2 Problem Description

The network design problem for railway infrastructure (NDRI) deals with the question: what is the design of a cost-optimal network of railway infrastructure, which has to fulfill a given traffic demand and certain capacity constraints?

Railway infrastructure is represented by a network consisting of large stations and lines connecting those stations. An input instance of NDRI consists of a traffic demand given by a set of estimated future traffic flows. Each traffic flow is defined by a pair of start and destination station and an amount of trains which has to be routed from the start to the destination station. The set of all start and destination stations defines exactly the set of all stations of the target network. So the stations of the network are implicitly given in advance.

The design process then deals with the determination of the network's topology and capacity to meet the given traffic demands at the lowest possible cost. Design decisions concerning the topology are only made with respect to lines, since the stations are given in advance as terminal points of traffic flows and, therefore, cannot be removed or

[2] A commodity or factor of production is called indivisible, if its use is forced to be above a minimum level, which usually results in a high share of fixed costs.

supplemented. So, the determination of the topology of the network is reduced to the question: which stations have to be connected to each other?

Design decisions concerning the capacity are affecting stations as well as lines. Dealing with both would lead to a problem with very high complexity. This is avoided here by just focussing on lines when dealing with questions concerning network capacity. Design of stations may be done in a subsequent step, which could be in the focus of future research. This approach is reasonable in relation to practice, too, where it is quite usual to dimension lines before stations. The reason lies in the fact that dimensioning of stations is in need of information, like traffic inflows and outflows, regarding lines converging at stations. As a consequence, here, stations are assumed to be equipped with unbounded capacity, which means that there is no restriction in the number of trains using a station at a time. So, the determination of network's capacity is reduced to the question: how much line capacity is required to route the traffic flows from their start to their destination station? In this regard, line capacity is expressed by different stages of extension of a line.

In order to prevent networks with oversized line capacities, a NDRI solution consists of a network which fulfills the traffic demand at lowest possible building costs. Since the stations are implicitly given in advance and, as mentioned before, are not involved in any design decisions, they are not included into cost calculations. So, building costs only arise if a line is included into the network during the design process.

1.3 Related Work

Network design problems are embedded in a wide range of practical problems and applications, especially in telecommunications and transportation planning. This results in many different models and solution approaches. Surveys are provided, for example, by Minoux [Min89], Pióro & Mehdi [PM04], and Magnanti & Wong [MW84].

In general network planning/design problems may combine tasks concerning dimensioning (capacity), routing of demands, and topology. Regarding communication networks, the compendium of Koster & Muñoz [KM10a] is dealing with all of such tasks separately as well as combined. On the one hand, there is, for example, the design of network topologies described by Koster & Muñoz [KM10b] in the first chapter of [KM10a]. The problem is just dealing with questions of network connectivity. Costs are associated with costs for establishing links between nodes. Since the focus is on the possibility to communicate, neither routing costs nor the capacity of links are considered. On the other hand, „classical“ network design problems like the *Optical Network Design* [KM10b] and the more complex *Two-layer Network Design Problem* [ORK+10] are presented. They combine the determination of topology, capacity, and demand routing.

Multicommodity Capacitated Network Design Problems especially *fixed-charge* ones are closely related to NDRI. They deal with routing of multicommodity flows and determination of arc capacity. Objective is to minimize routing and design costs. Routing costs are incurred whenever flow is routed through an arc. In addition to that, design costs, which may be fixed-charges, are being charged depending on the capacity which has to be installed on that arc. Gendron et al. [GCF98], on the one hand, give a survey of models for multicommodity capacitated network design problems and, on the other, present and compare several relaxation methods for efficient solution approaches of fixed-charge problems.

Apart from research on mathematical models for network design problems, there is some literature in the field of railway operations research which is dealing with the design/planning of railway infrastructure, too. Ross [Ros01] presents a detailed overview on a *Strategic Long-term Infrastructure Planning Process*. It covers different aspects from identification of capacity bottlenecks through to economical evaluation of infrastructure operations. Bendfeldt [Ben05] focusses on standardization of the design of railway station layouts. Stange [Sta08] introduces a method for optimized positioning of infrastructure elements on railway lines. Wieczorek [Wie06] describes a *Reverse Capacity Engineering* process. Thereby, stations and lines are dimensioned on the basis of a real timetable which is derived from detailed traffic demands.

1.4 Structure of this Work

The structure of the following chapters is three-part. First part consists of the next three chapters and deals with modeling issues. At first models for infrastructure, operation, and capacity are introduced (Chapter 2). After that mathematical models for NDRI using the models described before are presented (Chapter 3). Last chapter (Chapter 4) of this part shows some model refinements obtained by valid inequalities.

Second part focusses on solving, implementation and evaluation of mathematical models introduced before. This part encompasses Chapters 5 and 6. They deal with the setup of the branch-and-price framework, the implementation environment and two improvements of the solution process: the fixing of variables and determination of optimal Big-M values (Chapter 5). In addition to that Chapter 6 describes the evaluation framework and presents evaluation results.

Finally, last part which is equal to the last chapter (Chapter 7) summarizes key facts of this work and ends with concluding remarks.

2 Modeling Railway Infrastructure, Operation & Capacity

Designing networks of railway infrastructure is in need of a model for railway infrastructure and operation. Basically three modeling approaches can be distinguished: *microscopic*, *macroscopic*, and *mesoscopic* approaches. Sewcyk [Sew04] discerns the first two approaches as follows. In microscopic models infrastructure and operation is modeled in a fine-grained way which provides the opportunity of an accurate simulation of an individual operational event. This is necessary, for example, for a precise capacity analysis. These models can, for example, be applied in short-term and middle-term infrastructure planning processes where traffic demands may be available in a very detailed representations like timetables or sets of requested train paths. The granularity of microscopic models results in large volumes of data which limit the size of manageable study areas.

Macroscopic models do not provide the opportunity to focus on traffic as a set of single operational events. They are dealing with aggregated operational events like traffic flows which could be expressed as a number of trains using sections of the infrastructure within a certain time-frame. This representation of traffic makes a fine-grained model for the infrastructure dispensable. Infrastructure can be modeled in a more abstract and aggregated way, too. The level of abstraction ensures smaller volumes of data (in comparison to microscopic models) which allows an enlargement of the study area. Such macroscopic models, for example, fit well for long-term infrastructure planning processes, since they deal with forecasts of future traffic demands which could be far away from being as precise as a timetable and aim to examine entire networks.

There are models which are hybrids of microscopic and macroscopic models. Fellendorf et al. [FFV01] (road traffic), Kettner et al. [KSH04], and Sewcyk et al. [SRW07] (rail traffic) give examples of such hybrid approaches. These models are sometimes also referred to as *mesoscopic* models [Rad08].

The model used here is a macroscopic one and is defined regarding to Nießen &

Wendler [NW05] who give an example of a macroscopic model for the strategic long-term infrastructure planning. The question that needs to be answered first is: what needs to be modeled and what are the demands NDRI places on the model, respectively?

Central task in the network design process is the determination of line capacity which is required to route the given traffic flows to their destination stations. So, the choice of a method for the capacity assessment of lines considerably determines the choice of a model for infrastructure and operation.

2.1 Capacity

Besides the infrastructure elements itself, capacity of a railway line always depends on the operating programs executed on that line. A value for capacity by means of train counts separate from any kind of operation is not quantifiable[1]. This is because of the fact that the capacity consumption of single train depends on the preceding train using the same line. So for the determination of the capacity consumption of a mix of trains using a line one needs information about the succession of trains, i.e. the operating program. The fact that capacity consumption of one train depends on the preceding train is based on the concept of *minimum headway times*. Before this is shifted into focus, a fundamental principle of train movements has to be introduced: *train separation*.

2.1.1 Train Separation

In road traffic, separation of vehicles is based on the concept of relative braking distances. Vehicles follow each other in braking distance, which is enlarged by the distance a driver needs to react on the braking of the preceding vehicle. Such relative braking distance is, in general, shorter than the watching distance of the driver. If the preceding vehicle brakes, the succeeding driver can notice that by observing the flashing of the braking lights and, as a result, starts braking, too. In railway traffic that is not possible. Here, the relative braking distance, in most cases, exceeds the watching distance due to a steel wheel on steel rail system, which has, in comparison to road traffic, a significantly smaller coefficient of adhesion. This leads to the necessity of *train protection systems* which transmit movement authorities from the track to the trains. There are different

[1] This holds for any entity of railway infrastructure, like *sectional route nodes*, too. Sectional route nodes are the largest connected sub-networks of switching zones, in which usage of each route is mutual exclusive.

ways to implement such a system. An overview is given in [TV09]. In the current work, a train protection system which transmits the movement authority at discrete points (= fixed signals) is assumed, cf. Appendix A. Lines, which are of special interest here, are segmented by signals into block sections which have to be cleared by a train before the corresponding block signal in the beginning of the block section transmits the moving authority to a succeeding train. The segmentation of lines into block sections is of vital importance for the concept of minimum headway times. This is explained in the following.

2.1.2 Minimum Headway Time

A slight digression into the field of queuing theory is helpful to get an intuition of minimum headway times. A single track railway line can be modeled as a queuing system with a single service channel. A single supermarket checkout is a good example of such queuing system. The system simply spoken functions as follows: customers are arriving at the checkout, possibly have to wait in front of the checkout, and will be served after the preceding customer is served by the cashier. It is assumed, that the cashier not starts serving the next customer until the current customer has left the whole checkout area, which means that he has stowed away his purchase. Analogously, trains are arriving at the railway line, possibly have to wait in the upstream station and enter the track after the preceding train has left the track. That is, in both cases, an inaccurate model of the real-world. In fact, the cashier starts serving the next customer when the preceding one is still in the checkout area and is stowing away his purchase but will have finished doing that until the next customer is ready to stow away his own purchase. Regarding railway lines, it is the same situation. A train can enter the line before the preceding one has cleared all sections of the line, which is illustrated in the following sections. So the capacity consumption of a train or the service time, respectively, is not the total amount of time the train operationally uses the line, but that amount of time none other following train is allowed to start occupying sections of the line. In other words, the quantity which has to be determined to obtain the capacity consumption of a train using a certain line is the amount of time it takes at least before another train can follow. This quantity is called *minimum headway time* introduced by Happel in 1959 [Hap59].

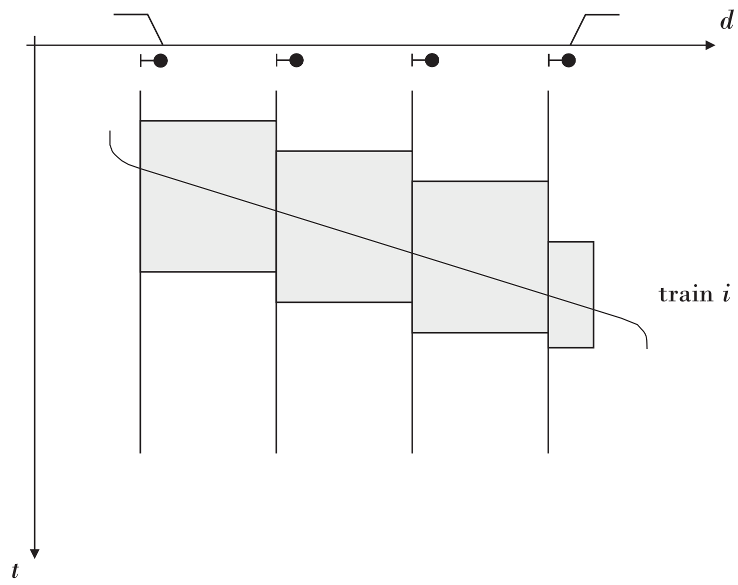

Figure 2.1: Time-distance diagram with a blocking time stairway for train i.

Blocking Time

To understand how the minimum headway time can be determined one has to understand how to express an operational use of a line. The operational use can be delineated by a so-called *blocking time* defined by Pachl [Pac08] in the following way:

> The blocking time is the total elapsed time a section of track (e.g. a block section, an interlocked route) is allocated exclusively to a train movement and therefore blocked for other trains.

The occupation of a line, which is partitioned into block sections, is modeled by a so-called *blocking time stairway* with blocking times for each block section, as pictured in Figure 2.1. A deeper insight into blocking time theory is given in Appendix A.1.

Determining the Minimum Headway Time

As already mentioned, the minimum headway time, denoted by z_{ij}, is the amount of time a train j has to wait at least when it wants to enter a track section of a line which is currently occupied by another train i. So, z_{ij} describes the occupation time or rephrasing the capacity consumption of train i in relation to successive train j. Using

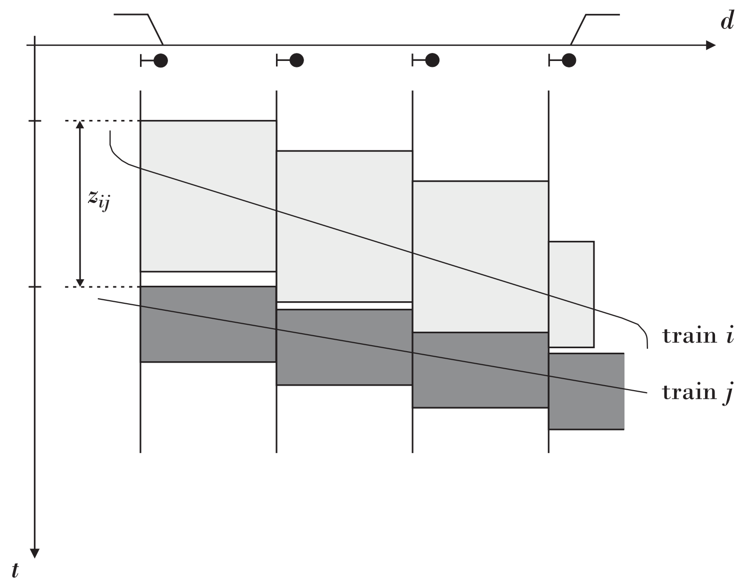

Figure 2.2: Minimum headway time z_{ij}.

the corresponding blocking time stairways of i and j, the minimum headway time of the ordered pair ij can be very clearly visualized, as shown in Figure 2.2. If train j wants to follow train i within the shortest time distance possible, the blocking time stairway of train j has to be shifted in time until there is no overlapping and there is at least one section with touching blocking times. The minimum headway time then can be read out from the first section. Furthermore, it is now clearly recognizable that train j is allowed to occupy the first block section before train i has left the last one.

The calculation of z_{ij} depends on properties of the type of trains i and j as well as on some properties of the line infrastructure itself.

2.1.3 Capacity Assessment of Lines

Minimum headway times have the advantage to combine properties of the trains running on the line and properties of the line's infrastructure. Therefore, they are used in this work to specify the *capacity profile* of a line.

A compact representation of that profile is a matrix of minimum headway times. Since minimum headway times have to be calculated pairwise, n train types, distinguished by parameters described later on, lead to a $n \times n$-matrix of minimum headway times (2.1).

The value at the ith row and jth column of that matrix represents the minimum headway time between a train of type i or a train of the ith type, respectively, and a following train of type j. In this work, train types are indicated by the use of identifiers t and u. Since minimum headway times are train type dependent values, they are indexed using such train type identifiers.

$$Z_{tu} = \begin{pmatrix} z_{tu} \end{pmatrix} = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nn} \end{pmatrix} \tag{2.1}$$

The matrix of minimum headway times or service times, respectively, then is used to determine the capacity consumption of a mix of trains. For a given succession, the capacity consumption is calculated by summing up the service times of each pair of trains succeeding each other. Two approaches to model successions of trains are introduced in Section 2.3.3.

To calculate the matrix of minimum headway times of a line, some parameters concerning the infrastructure and the trains using the line must be provided by corresponding models. The granularity of the infrastructure model as well as of the operation model must be chosen in a way such that the provided parameters yield adequate estimates of minimum headway times. Models and corresponding parameters are introduced in Section 2.2 and Section 2.3. A detailed insight into the calculation of minimum headway times using those parameters is given in Appendix A.2.

2.1.4 Alternative Capacity Assessment

A wide range of analytical methods for the capacity assessment of elements of infrastructure is based on the calculation of waiting times. In this regard, Gast [Gas87] introduces methods for the capacity assessment of lines, Wakob [Wak85] focuses on sectional route nodes, and Wendler [Wen02] on large stations and nodes, respectively. So-called *scheduled waiting times* arise during timetable construction, see Wendler [Wen99] and [Wen08]. They have to be distinguished from *unscheduled waiting times*, which arise during operation. Minimum headway times are of vital importance for the calculation of waiting times as well as for the capacity assessment used for NDRI. Chosen NDRI models for infrastructure and operation even enable the usage of waiting time for capacity assessment. Nevertheless, it is not the method of choice to simplify capacity calculations.

Scheduled Waiting Times

For a given set of requested train paths the timetable construction process aims to find a conflict-free positioning of these train paths. A conflict arises if at least two train paths compete for the same infrastructure elements at the same time. The conflict is resolved by the repositioning of the conflicting train paths. The scheduled waiting time is a measure for the repositioning effort which has to be made to construct a conflict-free timetable. There is a correlation between the utilization of infrastructure elements and scheduled waiting times, cf. Wendler [Wen08]. So, for a given admissible value of scheduled waiting times the capacity in terms of train counts can be specified.

Unscheduled Waiting Times

It is not uncommon that train operation suffers from (initial) delays. Such delays may trigger knock-on delays. Unscheduled waiting times result from knock-on delays and thus are a measure for the robustness of a timetable on a given infrastructure.

2.2 Infrastructure Model

In general, railway infrastructure encompasses the collectivity of *hardware* which is required to run trains. This contains rails, signals, switches, stations, balises, axle counters, and so on. In this work, as already mentioned, railway infrastructure is considered in a macroscopic way, focussing on parameters needed for reasonable estimations of capacity and cost.

Definition 2.1 (Network of Railway Infrastructure)**.** A network of railway infrastructure is represented by a directed graph $G := (N, A)$ with a set of nodes N and a set of arcs $\mathrm{A} \subseteq N \times N$. Each node $i \in N$ represents a station and each arc $(i, j) \in A$ a line connecting two stations $i, j \in N$.

Important properties of the network are specified by *attributes* assigned to each node and arc. These properties are *capacity* and *cost*. They are used by optimization models based on the infrastructure model. So, attributes serve as interface between the infrastructure model and optimization models and algorithms. Besides these attributes certain properties of the network of railway infrastructure are defined by *parameters*. They serve as input for the determination of cost and capacity, e.g., for reasonable estimations of minimum headway times. So, parameters can be considered as *internal* or *private* attributes.

2.2.1 Stations and Nodes

Following the definition taken out of UIC Code 406[2] [UIC04], stations are nodes of a network where at least two lines converge and where overtaking, crossing or direction reversals are possible, including marshalling yards. The macroscopic model used here uses a simplified definition, which distinguishes between two kinds stations: *terminal stations* and *overtaking stations*.

Terminal Station The terms *terminal station* and *station* describe the same entity of the infrastructure model and are therefore used synonymously. So terminal stations are the nodes N of a network of railway infrastructure $G = (N, A)$, as defined in Definition 2.1. They serve as start (source), destination (sink), and transshipment nodes of traffic flows. Overtaking, crossing and direction reversals are possible.

Overtaking Station In contrast to terminal stations, overtaking stations are not represented by nodes, because they do not serve as start or destination node of traffic flows. In an overtaking station a fast train is given the opportunity to pass a preceding slower train. These changes of successions of trains have an impact on the capacity of a line and the corresponding arc, respectively, cf. Section 2.2.2.

As mentioned before, station's properties are defined by a set of attributes assigned to the corresponding nodes. Explanatory notes regarding each attribute are given subsequent to the following definition.

Definition 2.2 (Attributes of a Node)**.** For a network of railway infrastructure $G = (N, A)$ and a node $i \in N$ the set of attributes $Attr_i$ of i is defined as follows: $Attr_i := \{cap_i,\, cost_i\}$, where

[2] Leaflet of the International Union of Railways which standardizes railway capacity analysis.

i) $cap_i \in \mathbb{R}^{\geqslant 0} \cup \{\infty\}$ specifies the capacity of node and

ii) $cost_i \in \mathbb{R}^{\geqslant 0}$ defines the cost of a node.

Capacity of a Node The capacity of a node cap_i specifies the operational capability of the corresponding station, which simply can be regarded as amount of trains, which is allowed to use a station at a time. Common approaches for the capacity assessment of a station use waiting times. This is a complex matter because a station has to be divided into switching zones (route nodes) and the station track group, which connects the route nodes. Waiting times then have to be separately calculated for both parts using different methods. Nießen [Nie08] and Nießen & Wendler [NW07] present methods focussing on route nodes. Potthoff [Pot70] and Fischer & Hertel [FH90] have worked on approaches usable in connection with station track groups. Even though such methods are applicable to NDRI they are, for reasons of complexity, not applied here. Instead, nodes are equipped with unbounded capacity (cf. Section 1.2):

$$\forall i \in N : cap_i := \infty. \tag{2.2}$$

Cost of a Node The cost of a node i represents construction costs which arise when building and running the corresponding station. Since the set of nodes is given in advance and remains unchanged (cf. Section 1.2), this value is set to zero for all nodes of the network:

$$\forall i \in N : cost_i := 0. \tag{2.3}$$

Definition 2.3 now defines parameters of a node, which are required for the calculation of minimum headway times.

Definition 2.3 (Parameter of a Node)**.** For a network of railway infrastructure $G = (N, A)$ and a node $i \in N$ the set of parameters of i is defined as follows: $Param_i := \{l_{\mathrm{st},i}, pos_i\}$, where

i) $l_{\mathrm{st},i} \in \mathbb{N}$ is the length of a station block [m] and

ii) $pos_i \subseteq \mathbb{R}^{\geqslant 0} \times \mathbb{R}^{\geqslant 0}$ determines the position of a node.

Station Block A *station block* is referred to as section between entry signal and depart signal of a station. For the sake of simplicity this value is assumed to be the same for all nodes in the network and it is fixed to a value which is reasonable in relation to practice.

$$\forall i \in N : l_{\mathrm{st},i} := 1{,}500 \text{ m}. \tag{2.4}$$

This value holds for overtaking stations of lines, too. Since the length of a station block is a constant for all stations, it is also referred to as just l_{st}.

Position of a Node The position pos_i of a node i is present as pair of coordinates and is used to derive the length of an arc, which is linking two nodes, see Section 2.2.2.

2.2.2 Lines and Arcs

Starting again with the UIC definition, a line is a link between two large nodes and usually consists of more than one line section. A line section is a part of a line, in which: the traffic mix and/or the number of trains, the infrastructure and signalling conditions do not change fundamentally. This description holds for this model too. Here, line sections are referred to as *overtaking sections*, which on the other hand are divided into block sections. The two kinds of segmentation are specified by line parameters. They are elements of the set of parameters needed, inter alia, for the estimation of minimum headway times.

Definition 2.4 (Parameter of an Arc)**.** For a network of railway infrastructure $G = (N, A)$ and an arc $(i, j) \in A$ the set of parameters of (i, j), is defined as follows:
$Param_{ij} := \{l_{\mathrm{l},ij},\ l_{\mathrm{bk},ij}, l_{\mathrm{ps},ij}, l_{\mathrm{so},ij},\ l_{\mathrm{os},ij}, S_{ij},\ n_{\mathrm{os},ij}, n_{\mathrm{tk},ij},\ t_{\mathrm{t},ij}\}$, where

i) $l_{\mathrm{l},ij} \in \mathbb{R}$ is the length of an arc [m],

ii) $l_{\mathrm{bk},ij} \in \mathbb{N}$ is the length of a block section [m],

iii) $l_{\mathrm{ps},ij} \in \mathbb{N}$ is the presignalling distance [m],

iv) $l_{\mathrm{so},ij} \in \mathbb{N}$ is the length of the safety overlap [m],

v) $l_{\mathrm{os},ij} \in \mathbb{R}$ is the average length of an overtaking section [km],

vi) S_{ij} is the set of available stages of extension,

vii) $n_{\mathrm{os},ij} \in \mathbb{R}$ is the average number of overtaking stations,

viii) $n_{\mathrm{tk},ij} \in \mathbb{N}$ is the number of tracks of the open line, and

ix) $t_{\mathrm{t},ij} \subseteq T$ is a set different train types.

Length of an Arc The length $l_{\mathrm{l},ij}$ of an arc $(i,j) \in A$ is defined as the distance between nodes $i, j \in N$, which represent the terminal stations of the corresponding line. It essentially depends on positions pos_i and pos_j of the corresponding nodes i and j (see Definition 2.3). Other parameters like topographic ones can also be included into the calculation. For the sake of simplicity, an Euclidean distance function is chosen here.

Block Section, Presignalling Distance & Safety Overlap A block section is a section of a track which usage is mutual exclusive, cf. Section 2.1.2. Figure 2.3 shows a line divided by block signal into block sections. The length of a block section $l_{\mathrm{bk},ij}$ is an important value for the calculation of minimum headway times and so an important parameter for the capacity of a line.

In the considered train protection system, the signal aspect of main signals, which may be block signals, is signalled in advance by so-called presignals. The distance between presignal and main signal is called presignalling distance, $l_{\mathrm{ps},ij}$, see Figure A.1.

The safety overlap is a feature of the train protection system. The safety overlap of the current block section is located in the beginning of the next block section. It has to be cleared by the current train movement before a succeeding train gets the permission to enter the block section. The safety overlap protects the current train movement against succeeding trains which may overrun a stop signal. In most cases, such „overrunning trains“ can be stopped within the safety overlap by automatic emergency brakings.

For the sake of simplicity the values of $l_{\mathrm{bk},ij}$, $l_{\mathrm{ps},ij}$, and $l_{\mathrm{so},ij}$ are assumed to be the same for all lines in the network and are fixed to an average values which are reasonable in relation to practice. So following equations hold:

$$\forall (i,j) \in A\colon l_{\mathrm{bk},ij} := 2{,}000 \text{ m}, \tag{2.5}$$

$$\forall (i,j) \in A\colon l_{\mathrm{ps},ij} := 1{,}000 \text{ m}, \tag{2.6}$$

$$\forall (i,j) \in A\colon l_{\mathrm{so},ij} := 200 \text{ m}. \tag{2.7}$$

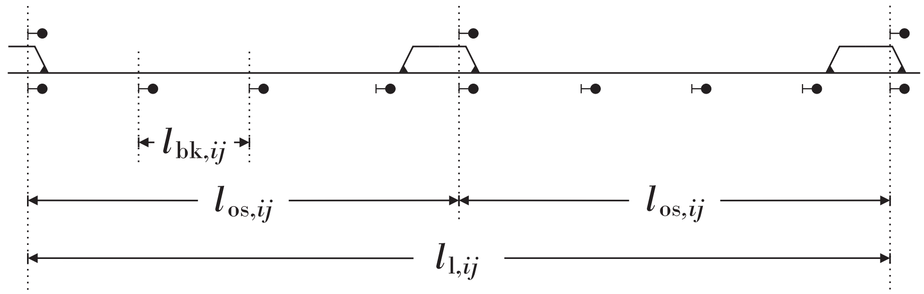

Figure 2.3: Parameters of an arc $(i,j) \in A$.

Since those values are constant for all lines they are also referred to as just l_{bk}, l_{ps}, and l_{so}.

Overtaking Sections and Stations To quantify minimum headway times of successive trains the line has to be divided into common track sections, which do not provide the opportunity to change the succession of trains. The perpetuation of the succession in such a section enforces the headway time to be at least the minimum headway time. This common track section is called *overtaking section*. Overtaking stations and terminal stations establish the possibility to change the succession of trains, so the length of an overtaking section corresponds to the distance between two stations adjacent to that section. The minimum headway time of a line as a whole then is the largest minimum headway time among all minimum headway times determined for each overtaking section.

In the macroscopic model used here, this approach is replaced by the use of an average length of overtaking sections specified by the value of $l_{\mathrm{os},ij}$, see Figure 2.3. Here, the variable domain of $l_{\mathrm{os},ij}$ is restricted to up to five different values. This holds for each $(i,j) \in A$.

$$l_{\mathrm{os},ij} \in \{i \mid i \in \{5, 10, 20, 50, 100\} \colon \frac{l_{\mathrm{l},ij}}{i} \geq 2\} \tag{2.8}$$

Lines without any overtaking station, i.e. $l_{\mathrm{os},ij} = l_{\mathrm{l},ij}$, are not considered here. This results in significant simplifications when it comes to capacity profiles of lines where $l_{\mathrm{os},ij}$ is of vital importance, cf. Appendix A.2. Due to the restriction to the five uniform stages of extension, only five capacity profiles have to be considered. Lifting the restriction would lead to up to $\frac{|A|(|A|-1)}{2}$ additional capacity profiles, i.e. matrices of minimum headway times.

Lines are classified by their average length of an overtaking section into so-called *stages of extension*. The pair $(l_{\text{os},ij}, n_{\text{tk},ij})$ uniquely defines the current stage of extension of arc (i, j). S_{ij} denotes the set of all stages of extension, which are considered here:

$$S_{ij} := \{i \mid i \in \{5, 10, 20, 50, 100\} \colon \frac{l_{\text{l},ij}}{i} \geq 2\} \times \{1, 2\}. \tag{2.9}$$

The average length of an overtaking station implicitly determines the average number of overtaking stations $n_{\text{os},ij}$ a line is equipped with:

$$n_{\text{os},ij} := \frac{l_{\text{l},ij}}{l_{\text{os},ij}} - 1. \tag{2.10}$$

Correction term -1 must be added because adding k overtaking stations in distances of $l_{\text{os},ij}$ results in $k + 1$ sections of length $l_{\text{os},ij}$.

Regarding line/arc capacity one can keep in mind: the greater the number of overtaking stations, the shorter the average distances between them, the shorter the minimum headway times and the greater the number of trains which can use the line during a certain time frame. This holds only for operating programs with different types of trains, since otherwise overtaking stations are useless.

Number of Tracks Here, a line can only be present as single-tracked or double-tracked line, which has an impact on the domain of the corresponding arc parameter:

$$\forall (i, j) \in A : n_{\text{tk},ij} \in \{1, 2\}. \tag{2.11}$$

In the usual terminology a single track railway line is used in both directions, whereas a double-tracked one consists of one track for each direction. In this model, which uses directed arcs as representation of lines (cf. Definition 2.1), the use of single track lines as well as the use of double track lines is restricted to one direction. So if a double track line in the common sense has to be established within the current model, one single track line for each direction has to be included into the network. Neglecting single track lines which can be used in both directions significantly simplifies the model. It is also appropriate in relation to practice, since single track lines in the classical sense are used for branch lines with low density of traffic. Such lines play a minor role in the considered long-term infrastructure planning process which focusses on main lines between large stations.

Train Types The set of train types $t_{t,ij} \in T$ defines which train types are allowed to access the line, where T is the set of train types considered here. The value of $t_{t,ij}$ may differ from line to line. Train types are introduced in Section 2.3.1.

Line's capacity and cost are defined by a set of attributes assigned to the corresponding arcs. These attributes are introduced in Definition 2.5. Again, explanatory notes regarding each attribute are given subsequent to the definition.

Definition 2.5 (Attributes of an Arc)**.** For a network of railway infrastructure $G = (N, A)$ and arc $(i, j) \in A$ the set of attributes $Attr_{ij}$ of (i, j) is defined as follows: $Attr_{ij} := \{cap_{ij}, cost_{i,j}\}$, where

i) cap_{ij} defines the capacity of (i, j). It consists of parts: a capacity profile $Z_{tu,ij} \in \mathbb{R}^{|t_{t,ij}| \times |t_{t,ij}|}$ and the number of tracks $n_{tk,ij}$.

ii) $cost_{ij} \colon \mathbb{R} \times \mathbb{N} \times \mathbb{N} \times \mathbb{R} \to \mathbb{R}$ defines the cost of the arc.

Capacity of an Arc The capacity of an arc is expressed by the corresponding capacity profile which is the matrix of minimum headway times, as introduced in Section 2.1. Thereby, minimum headway times are calculated pairwise for all train types $t, u \in t_{t,ij}$. The initial notation used for matrices of minimum headway times Z_{tu} is extended by arc indices ij. Besides fixed lengths for block sections and station blocks, the average length of an overtaking section $l_{os,ij}$ is of vital importance, cf. Appendix A.2. In addition to that, the number of tracks has a decisive influence on the capacity of an arc. The capacity profile remains unchanged regardless of the value of $n_{tk,ij}$, but the maximal capacity $max_{cc,ij}$, which is consumable by a mix of trains using the line, is doubled in case of a double track line ($n_{tk,ij} = 2$), cf. Section 2.3.3 and Equation (2.15).

Cost of an Arc The costs of an arc are defined as the costs to build and maintain the associated line. This is modeled by the use of *life-cycle costs* which are calculated according to the general formula:

$$LCC = \frac{AC}{L} + MC \quad [\text{€}/\text{a}]\,, \tag{2.12}$$

where AC are *acquisition costs* in euro, L is the *operational lifespan* in years and MC are the *maintenance costs* in euro per year.

Arc's costs are assumed to be load-independent since line's construction costs arise regardless of the number trains which use the line. A shortened life-span of infrastructure elements induced by heavy load is not taken into consideration here. Maintenance costs of railway infrastructure, which usually are load-dependent, are estimated by average values gained from experience.

To model load-dependent life-cycle costs of infrastructure elements, load-dependent arc costs have to be introduced and included into the mathematical optimization models, which are presented later on. This would significantly increase the complexity and in turn decrease the solvability of such models. This is not desirable. So, particularly with regard to the macroscopic model for infrastructure, it is reasonable to lower the level of precision to benefit from significant simplifications induced by load-independent life-cycle costs.

Life-cycle cost analysis of a line is focussing on three infrastructure elements: *tracks*, *(main) signals* and *switches*. The minimum costs of each arc (i, j) are its *default costs*, which denote the life-cycle costs of the corresponding line without any overtaking station. It is defined by the following function

$$LCC_{\mathrm{def},ij} \colon \mathbb{R} \times \mathbb{N} \to \mathbb{R} \tag{2.13}$$

which depends on length of the line (relevant to track costs) and the average length of a block section (relevant to signal costs). Each additional overtaking station creates additional (constant) costs LCC_{pt} for the passing track. So life-cycle costs of arc $(i, j) \in A$ are defined as follows:

$$cost_{ij}(l_{\mathrm{l},ij}, l_{\mathrm{bk}}, n_{\mathrm{tk},ij}, n_{\mathrm{os},ij}) := n_{\mathrm{tk},ij} \left(LCC_{\mathrm{def},ij}(l_{\mathrm{l},ij}, l_{\mathrm{bk}}) + n_{\mathrm{os},ij} \cdot LCC_{\mathrm{pt}} \right). \tag{2.14}$$

Appendix B provides detailed information of the life-cycle cost analysis as well as a sample calculation.

2.3 Operation Model

This section describes which types of trains are considered in this work, which train type parameters are needed to estimate minimum headway times, how traffic demand is defined, and how to determine capacity consumption of successions of trains by using such estimated capacity profiles.

2.3.1 Train Types

To model operation on lines, properties of trains have to be taken into account. Due to the granularity of the macroscopic model, it suffices to classify trains with similar properties into so-called *train types*.

Definition 2.6 (Parameters of a Train Type)**.** For all train types $t \in T$ the set of parameters $Param_t$—needed for the estimation of minimum headway times—is defined as follows: $Param_t := \{v_t,\, d_t,\, l_{\mathrm{tr},t}\}$, where

i) $v_t \in \mathbb{N}$ is the operating speed [km/h],

ii) $d_t \in \mathbb{R}$ the averaged deceleration rate [m/s^2], and

iii) $l_{\mathrm{tr},t} \in \mathbb{N}$ the length of train type t [m].

Here, the acceleration rate of a train type is not relevant to calculations of minimum headway times, cf. Appendix A.2.

2.3.2 Traffic Demand

The traffic load a network has to take is represented by a future traffic demand derived from forecasts. Such traffic demand consists of traffic flows. A traffic flow is simply represented by a number of trains which have to be routed directly or indirectly from a start to a destination station within an observation period t_{op}. It is not mandatory for all the trains of a single traffic flow to be routed via the same route, so flow splitting is possible. Of course each fraction retains its start and destination station.

Definition 2.7 (Traffic Flow)**.** For a network of railway infrastructure $G = (N, A)$ and a set of train types T the traffic flow τ is defined as $\tau := (st_\tau,\, de_\tau,\, tr_\tau)$, where

i) $st_\tau \in N$ is the start station,

ii) $de_\tau \in N$ is the destination stations, and

iii) $tr_\tau := \{n_{t,\tau} \in \mathbb{N} \mid t \in T\}$ is set of trains which have to be routed from st_τ to de_τ. $n_{t,\tau}$ is the number of trains of train type $t \in T$.

The set of all traffic flows of network G forms the traffic demand, which is denoted by TD.

2.3.3 Successions of Trains & Capacity Consumption

When routing a mix of trains via a line, a certain quality of service regarding the operation of the trains has to be ensured. This can be done by restricting the capacity consumption of the mix of trains to a threshold value. To do so, following question has to be answered first: how much capacity does a mix of trains consume when using a line?

As explained in Section 2.1, the capacity of an arc is defined by the capacity profile of the corresponding line. Such profile only specifies the capacity consumption of single train depending on the preceding train. To answer the question, which was asked in the beginning, one has to know in which succession trains enter the line. Basically, two cases must be differentiated in this context: the order of trains is fixed by a timetable or there exists no timetable or fixed succession, respectively. In both cases, it is assumed that trains succeed each other in a minimum time distance which is the minimum headway time. This is done following UIC Code 406 [UIC04]. It says regarding determination of consumption of line capacity that timetables have to be *compressed*...

> ... up to the minimum theoretical headway according to their timetable order, without recommending any buffer time.

Whenever the term *timetable* is mentioned here in connection with capacity consumption it refers to such compressed successions of trains, as shown in Figures 2.4 and 2.5.

On the basis of current practice, UIC Code 406 provides a guideline regarding the level of line capacity consumption, i.e. the occupation time, which ensures a satisfying quality of service. For mixed-traffic lines, on which this work is focussing, 60% of a daily period, respectively the observation period t_{op}, is specified to be an appropriate value. Calculated capacity consumptions are tested against this value which describes the maximum available time for operation of trains. It is denoted by $max_{\mathrm{cc},ij}$. It should be noted, that $max_{\mathrm{cc},ij}$ depends on the number of tracks of the considered line, cf. Section 2.2.2.

$$max_{\mathrm{cc},ij} := 0.6 \cdot t_{\mathrm{op}} \cdot n_{\mathrm{tk},ij} \tag{2.15}$$

If the maximum available time for operation of trains shall be addressed independent of a specific arc, it is denoted by max_{cc}. For the calculation of max_{cc} the maximum number of tracks for one direction, $n_{\mathrm{tk},ij} = 2$, is used, cf. explanatory notes of Equation (2.11).

$$max_{\mathrm{cc}} := 0.6 \cdot t_{\mathrm{op}} \cdot 2 \tag{2.16}$$

Coming back to the fixed or unfixed orders of trains, at first, a timetable independent method for the determination of capacity consumption is introduced followed by a timetable dependent one. At the end, the special case of homogeneous operation is described.

A Timetable Independent Estimation

When designing a new network of railway infrastructure, intuitively, there does not exist a timetable for the trains. This is obvious because the routing of the trains is not precisely known in advance. In long-term infrastructure planning processes usually a random succession of trains is assumed to overcome this difficulty. Thereby, the service time of a train, i.e. the capacity consumption or minimum headway time, is modeled as stochastic process, see Wendler [Wen07]. To derive the mean capacity consumption of all trains the expected service time (= mean minimum headway time) has to be determined. For a network of railway infrastructure $G = (N, A)$, let $p_{tu,ij}$ be the probability of the occurrence of a succession of trains of type t and u competing for the capacity of line $(i, j) \in A$, T_{s} the random variable modeling the service time, and minimum headway times $z_{tu,ij}$ the random variates of T_{s}. Then the following holds:

$$\mathbf{P}[T_{\mathrm{s}} = z_{tu,ij}] = p_{tu,ij}. \tag{2.17}$$

To derive $p_{tu,ij}$, as shown in (2.18), let X_k be the random variable modeling the type of the kth train in the random succession of trains.

$$p_{tu,ij} = \mathbf{P}[X_{k+1} = u \wedge X_k = t] = \mathbf{P}[X_{k+1} = u \mid X_k = t] \cdot \mathbf{P}[X_k = t]. \tag{2.18}$$

Since the events X_{k+1} and X_k are not necessarily statistically independent, the conditional probability $\mathbf{P}[X_{k+1} = u \mid X_k = t]$ is required for the derivation. For the sake of simplicity during the solution process of the NDRI, it is assumed that there exists statistical independence between the mentioned events. Such a statistical independence is often assumed in practical applications. Thereby, it is assumed that the mix of trains does not change during the observation period, i.e. the arrival (at a service channel) of a train of a certain type does not change the share of that train type in the given mix ratio. This is equivalent to *drawing with replacement* in the *urn model* known from probability theory. Statistical independence of events X_{k+1} and X_k leads to the following representation of $p_{tu,ij}$ in (2.19).

$$\begin{aligned} p_{tu,ij} &= \mathbf{P}[X_{k+1} = u \wedge X_k = t] = \mathbf{P}[X_{k+1} = u] \cdot \mathbf{P}[X_k = t] \\ &= \frac{n_{t,ij}}{nt_{ij}} \cdot \frac{n_{u,ij}}{nt_{ij}} = \frac{n_{t,ij} \cdot n_{u,ij}}{nt_{ij}^2}, \end{aligned} \tag{2.19}$$

where $n_{t,ij}$ is the number of trains of type t on arc (i, j) and nt_{ij} the total number of trains on arc (i, j). Equation (2.20) then shows the desired expected service time $\mathbf{E}[T_s]$ respectively the mean minimum headway time $\bar{z}_{tu,ij}$.

$$\bar{z}_{tu,ij} = \mathbf{E}[T_\mathrm{s}] = \sum_t \sum_u p_{tu,ij} \cdot z_{tu,ij} = \sum_t \sum_u \frac{n_{t,ij} \cdot n_{u,ij}}{nt_{ij}^2} \cdot z_{tu,ij}. \tag{2.20}$$

Multiplying $\mathbf{E}[T_\mathrm{s}]$ with the total number of trains results in the expected total (timetable independent) capacity consumption $cc_{\mathrm{ti},ij}$ of all trains routed via the line (i, j):

$$cc_{\mathrm{ti},ij} = nt_{ij} \cdot \mathbf{E}[T_\mathrm{s}] = nt_{ij} \cdot \sum_t \sum_u \frac{n_{t,ij} \cdot n_{u,ij}}{nt_{ij}^2} \cdot z_{tu,ij}. \tag{2.21}$$

Using the matrix of minimum headway times, which can be calculated in advance, as already mentioned, the expected minimum headway time can be derived. This is equal to the expected service time, which corresponds to mean capacity consumption of a train. Multiplying this value with the total number of trains results in the expected capacity consumption of all trains routed via the line concerned.

A Timetable Dependent Derivation

As already mentioned, in long-term infrastructure planning processes capacity analysis usually is done using timetable independent approaches. In spite of that, a timetable dependent derivation of capacity consumption is presented here. This approach is used later on to derive a mathematical optimization model for NDRI with simplified capacity constraints.

The idea is to construct an artificial worst-case timetable, which induces a worst-case capacity consumption of trains. The network designed by this means can be considered as upper bound for the infrastructure needed. This upper bound property holds for any timetable dependent capacity consumption, which in turn creates timetable independence, too.

In the case of two different train types (a fast and a slow one), the worst-case timetable is constructed as follows: as far as possible a slow train should be followed by a fast train. Pairs of trains ordered in this way are resulting in largest minimum headway times and so are leading to worst-case capacity consumptions. Let $n_{1,ij}$ be the number of trains of type 1, a fast train type, $n_{2,ij}$ the number of slow trains of type 2. For the calculation of the total (timetable dependent) capacity consumption $cc_{\mathrm{td},ij}$, two cases must be distinguished:

$$\begin{aligned} & n_{1,ij} \geqslant n_{2,ij} \geqslant 0 : \\ & cc_{\mathrm{td},ij} := n_{2,ij} \cdot z_{21,ij} + (n_{2,ij} - 1) \cdot z_{12,ij} + (n_{1,ij} - n_{2,ij}) \cdot z_{11,ij} + z_{12,ij} \\ & \qquad\quad = (z_{21,ij} + z_{12,ij} - z_{11,ij}) \cdot n_{2,ij} + z_{11,ij} \cdot n_{1,ij}. \end{aligned} \tag{2.22}$$

$$\begin{aligned} & n_{2,ij} \geqslant n_{1,ij} \geqslant 0 : \\ & cc_{\mathrm{td},ij} := n_{1,ij} \cdot z_{21,ij} + n_{1,ij} \cdot z_{12,ij} + (n_{2,ij} - n_{1,ij} - 1) \cdot z_{22,ij} + z_{22,ij} \\ & \qquad\quad = (z_{21,ij} + z_{12,ij} - z_{22,ij}) \cdot n_{1,ij} + z_{22,ij} \cdot n_{2,ij}. \end{aligned} \tag{2.23}$$

where $z_{tu,ij}$ with $t, u \in \{1, 2\}$ are corresponding minimum headway times on arc (i, j). The derivation of these equations is illustrated by examples depicted in Figure 2.4 and Figure 2.5. In both cases last added minimum headway time (last addend in the first equation of (2.22) and (2.23), respectively) is that one between last and first train which arises from the assumption that the timetable repeats itself.

For $|T| > 2$ different train types it is nontrivial to derive the worst-case capacity consumption of a mix of trains. The problem seems to be a variant of the well-known *Traveling Salesman Problem* (TSP), which is *NP-hard*[3]. This would imply that there exists no way to efficiently calculate the worst-case capacity consumption, which in turn would lead to a rise in complexity of NDRI models using this worst-case approach. To figure out, whether there is a way to efficiently calculate the worst-case capacity consumption or if TSP is in fact polynomially reducible to this problem, is beyond the focus of this work. So, worst-case capacity consumption is considered only for $|T| \leqslant 2$.

Homogeneous Operation

If only trains of one single train type operate on a line, operation is referred to as *homogeneous operation*. This type of operation allows high frequencies of trains. This is, for example, observable in subway systems.

Derivation of the capacity consumption $cc_{\mathrm{ho},ij}$ becomes very easy since there is only one minimum headway time. For an arc $(i,j) \in A$ and $t_{\mathrm{t},ij} = t$ introduced types of capacity consumption coincide:

$$cc_{\mathrm{ho},ij} = cc_{\mathrm{ti},ij} = cc_{\mathrm{td},ij} = n_{t,ij} \cdot z_{tt,ij}. \tag{2.24}$$

Let $max_{t,ij}$ be the maximum train count of trains of type t, which can operate on the corresponding line with respect to the capacity available for consumption $max_{\mathrm{cc},ij}$. This maximum train count can only achieved by homogeneous operation, so it is derived as follows:

$$max_{t,ij} := \left\lfloor \frac{max_{\mathrm{cc},ij}}{z_{tt,ij}} \right\rfloor \tag{2.25}$$

If the maximum train count is addressed independent of a specific arc, $max_{\mathrm{cc},ij}$ instead of $max_{\mathrm{cc},ij}$ is used for the derivation.

$$max_{t} := \left\lfloor \frac{max_{\mathrm{cc}}}{z_{tt,ij}} \right\rfloor \tag{2.26}$$

It should be noted, that the value of $z_{tt,ij}$ stays the same for every arc (i,j), since changeable parameters of an arc like $n_{\mathrm{tk},ij}$ or $l_{\mathrm{os},ij}$ do not have an impact on minimum headway times of homogeneous train successions.

[3]The term NP-hard is explained in Section 3.1.1.

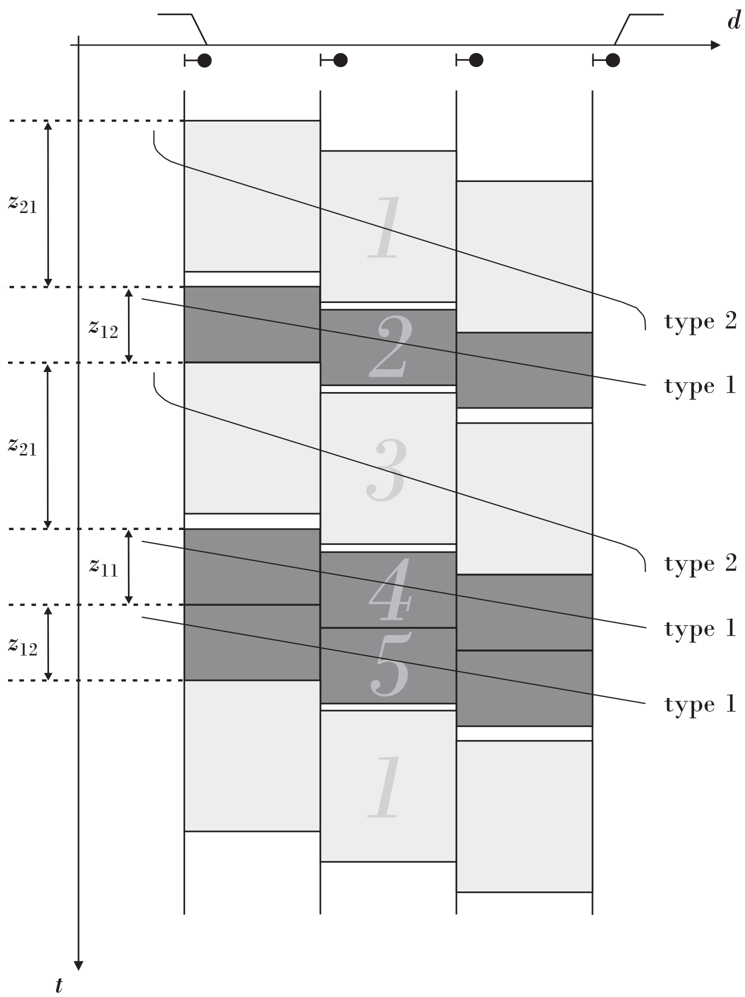

Figure 2.4: Capacity consumption $cc_{\text{td},ij}$ in case of $3 = n_{1,ij} \geqslant n_{2,ij} = 2$:
$$\begin{aligned} cc_{\text{td},ij} &= n_{2,ij} \cdot z_{21,ij} + (n_{2,ij} - 1) \cdot z_{12,ij} + (n_{1,ij} - n_{2,ij}) \cdot z_{11,ij} + z_{12,ij} \\ &= 2 \cdot z_{21,ij} + 2 \cdot z_{12,ij} + 1 \cdot z_{11,ij}. \end{aligned}$$

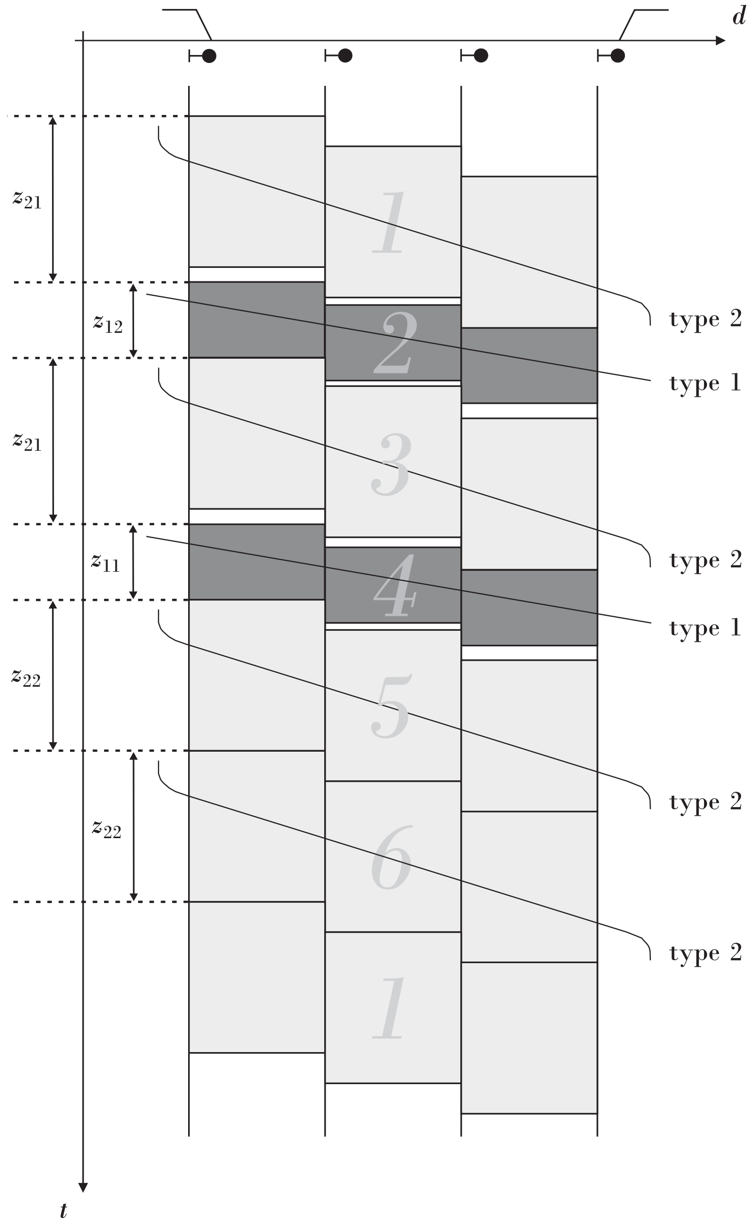

Figure 2.5: Capacity consumption $cc_{\mathrm{td},ij}$ in case of $4 = n_{2,ij} > n_{1,ij} = 2$:
$$cc_{\mathrm{td},ij} = n_{1,ij} \cdot z_{21,ij} + n_{1,ij} \cdot z_{12,ij} + (n_{2,ij} - n_{1,ij} - 1) \cdot z_{22,ij} + z_{22,ij}$$
$$= 2 \cdot z_{21,ij} + 2 \cdot z_{12,ij} + 2 \cdot z_{22,ij}.$$

3 Optimization Models for NDRI

This chapter introduces mathematical optimization models used to model and solve the network design problem for railway infrastructure. Before those models are described in detail, the problem description is recapitulated, cf. Section 1.2. This shall clarify which specific demands of the formulated problem are put on the structure of suitable mathematical models. Following this clarification, applied mathematical models are then introduced.

But first of all a short introduction into linear optimization problems is given.

3.1 Linear Programming

A linear programming problem (LP) is a mathematical optimization problem which consists, on the one hand, of a linear function which has to be maximized or minimized. It is usually denoted as *objective function*. On the other hand, there is a system of linear equations or linear inequalities which have to be fulfilled. Linear equations and inequalities are referred to as *linear constraints*. Equations (3.1)–(3.3) show a LP in *standard form*, cf. Chvátal [Chv83]. This form is also referred to as *canonical* or *symmetric* form.

$$\text{Maximize } \sum_{i=1}^{n} c_i x_i \tag{3.1}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i \qquad \forall i \in 1, \ldots, m, \tag{3.2}$$

$$x_j \geqslant 0 \qquad \forall j \in 1, \ldots, n, \tag{3.3}$$

The system of m constraints (3.2) often is stated using a vector-matrix notation:

$$A\mathbf{x} \leqslant \mathbf{b}, \tag{3.4}$$

where $A = (a_{ij})$ is a $m \times n$ – (coefficient) matrix, vector $\mathbf{x}$ consists of variables x_j and $\mathbf{b}$ of corresponding right-hand side values b_i. In style of that, constraints are referred to as *rows*, variables as *columns*.

If the domain of all variables x_j is integer, the problem is called *integer programming problem* (IP). If this applies only to some variables, it is called *mixed integer programming problem* (MIP).

3.1.1 Solving

Linear programming problems can be solved very efficiently, even in the case of very large problem instances. Algorithms with provable polynomial runtimes are, e.g., introduced by Khachiyan [Kha79] (*Ellipsoid Method*) and by Karmarkar [Kar84] (*Interior-Point Method*). A very efficient method is the *Simplex Algorithm* developed by Dantzig [Dan63]. Even though its exponential complexity in the worst-case, in relation to practice it turned out to be a very efficient algorithm.

Whereas an LP is efficiently solvable, this does not hold for (mixed) integer programming problems. They belong in general to the class of *NP-hard* problems. The term NP-hard is taken from *computational complexity theory*. A problem is NP-hard, if every problem which is in NP is *polynomial-time reducible* to it. That means, a NP-hard problem is at least as hard to solve as every problem which is element of complexity class NP. Since NP includes all *NP-complete* problems, which are supposedly not efficient solvable, an efficient algorithm to solve NP-hard problems is unknown. Best known deterministic algorithms for such problems have an exponential time complexity. For more information about hard problems and complexity theory see Hromkovič [Hro01] who provides an overview of algorithms for hard problems.

Methods to solve (M)IPs try to find an assignment of values of the integer variables which forms an optimal solution. One way to achieve that is to enumerate all possible assignments of values by a so-called explicit *enumeration tree*. This results even in the binary case even for a small set variables in a huge number of tree nodes. For n binary variables the enumeration tree consists of 2^n leaf nodes and in total of $2^{n+1} - 1$ nodes. Thereby, each leaf node corresponds to one value assignment of the n binary variables.

So, it is desirable not to explore the whole tree. This can be avoided by the calculation of bounds during tree building. With the help of these bounds it is possible to prune branches of the tree, so that they must not be explored. This describes a *branch-and-bound* (B&B) strategy, first introduced by Lang and Doig [LD60]. The most common *bound* is the linear programming relaxation of the (M)IP at the current node of the B&B-tree. Such relaxation can be obtained by dropping all integrality constraints. In the case of a maximization, problem the optimal solution of the relaxation provides an upper bound on every solution of the (M)IP. The B&B-tree is spanned by *branching*. Thereby, a variable which has a fractional value in the solution of the relaxation has to be selected. Using the fractional value, the domain of the variable is split into two parts each of them forming a subproblem of the current problem. Both subproblems are assigned to new nodes and are linked into the B&B-tree as child nodes of the current node.

3.1.2 Optimality Gap

The integer solution which is currently the best during B&B process is called *incumbent*. Its objective value serves as *primal bound* on the optimal solution. In case of a minimization problem it is an upper bound. The currently best lower bound which is obtained from relaxations is called *dual bound*. If both quantities, primal and dual bound, have the same value, an optimal solution is found and B&B terminates. In order to determine the quality of the current solution during the B&B and to determine the progress of the solution process, the so-called *optimality gap* can be consulted. The *(absolute) optimality gap* denotes the difference between primal and dual bound. If this difference is put in relation to the primal or the dual bound, one receives the *relative optimality gap*, which is a unit-free measure used to assess the current status of the solution process. In this work, the relative optimality gap is calculated according to the following formula:

$$\text{relative optimality gap} := \frac{|\text{primal bound} - \text{dual bound}|}{\text{primal bound}} \tag{3.5}$$

The relative optimality gap thus constitutes an upper bound on the relative error, which is a measure to assess the quality of the current solution in relation to the optimal solution.

For more information about integer programming and B&B see Wolsey [Wol98].

3.2 NDRI Problem Description

The problem description starts with the objective of NDRI. After that, characteristics are described by listing key parameters defining an input instance and by defining output information a solution of NDRI has to provide. This is done in an abstract way—using terms introduced in the last chapter—to reveal the structure of the problem.

Objective Objective of the network design problem for railway infrastructure is the determination of a cost-optimal network of railway infrastructure, which has to fulfill a given traffic demand and certain capacity constraints.

Input An input instance of NDRI consists of the following information:

- An empty network of railway infrastructure $G = (N, \emptyset)$ containing nodes $n \in N$ representing stations with unbounded capacity and zero life-cycle cost.
- A set of directed arcs $A \subseteq N \times N$, which represents constructible lines between the stations. Each arc $a \in A$ can be in one of up to ten stages of extension, defined by different values of the attributes capacity profile cap_{ij} and costs $cost_{ij}$ of including the corresponding line into the network.
- A traffic demand TD, defined as a set of traffic flows. Each flow consists of trains of different types $t \in T$. Train types are distinguished by a set parameters $Param_t$ which have an impact on their capacity consumption.
- A method to determine the capacity consumption of a mix of trains using a line and a quality parameter $max_{\mathrm{cc},ij}$, which limits the amount of capacity which is consumable. This includes a definition of how trains succeed one another.

Output The solution of NDRI is a graph $G^{\mathrm{opt}}= (N, A^{\mathrm{opt}})$ with $A^{\mathrm{opt}} \subseteq A$ and should cover following aspects:

- Topology: which stations have to be connected directly to each other? $\rightarrow$ Determination of a set of arcs $A^{\mathrm{opt}} \subseteq A$.
- Capacity: which stage of extension is associated to each arc $(i, j) \in A^{\mathrm{opt}}$? $\rightarrow$ Determination of cap_{ij} for each $(i, j) \in A^{\mathrm{opt}}$.

- Routing: which route is used by which train to reach its destination station?
- LCC: what are the optimal life-cycle costs of G^{opt}? $\rightarrow$ Summing up $cost_{ij}$ for all $(i,j) \in A^{\text{opt}}$.

As described, requirements for a solution of NDRI reveal the two-part structure of the problem. On the one hand, there is the network design problem which determines the topology of the network and the amount of capacity arcs have to be equipped with. On the other hand, there is the routing of traffic demands which determines the route of each train and which has to fulfill arc-induced capacity constraints. Let us first consider how the routing problem for traffic demands can be modeled and after that how the network design problem can be solved.

3.3 The Routing Problem

In contrast to NDRI, the routing (sub-)problem NDRI-R deals with a network graph which is given in advance, e.g., as output of a network design operation. The objective of NDRI-R is to find the best routing of each traffic flow from its start to its destination station. Thereby it has to be ensured that line capacity limitations of the given network graph are not exceeded. The quality of a routing is assessed by a function which assigns costs to the routing whenever trains are routed via a line.

To start with a simple model of NDRI-R, at first three restrictions are imposed:

R1: Only one traffic flow, $|TD| = 1$.

R2: Only one train type $T = \{t\}$.

R3: Only one stage of extension.

These restrictions will be lifted in the course of the modeling process. In addition to that, a timetable independent capacity consumption $cc_{\text{ti},ij}$ of lines (i,j) is assumed, cf. Equation (2.21). Due to restriction R2, which is tantamount to homogeneous operation, $cc_{\text{ti},ij}$ simplifies to:

$$cc_{\text{ti},ij} = cc_{\text{ho},ij} = n_{t,ij} \cdot z_{tt,ij}. \tag{3.6}$$

NDRI-R defined by this means is equivalent to the well-known *minimum cost flow problem*. NDRI-R can be stated as optimization problem analogous to the definition of the min-cost flow problem by Ahuja, Magnanti and Orlin [AMO93].

$$\text{Minimize} \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{3.7}$$

subject to

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i \qquad \forall i \in N, \tag{3.8}$$

$$x_{ij} \leqslant max_{t,ij} \qquad \forall (i,j) \in A, \tag{3.9}$$

$$x_{ij} \geqslant 0 \qquad \forall (i,j) \in A, \tag{3.10}$$

where $G = (N, A)$ is the given network of railway infrastructure and $\tau = (st_\tau, de_\tau, \{n_{t,\tau}\})$ the only traffic flow. x_{ij} is the number of trains using arc $(i,j) \in A$, c_{ij} denotes the cost per train using the corresponding arc, and for each node $i \in N$ the value of b_i is defined in the following way:

$$b_i := \begin{cases} n_{t,\tau} & \text{if } i = st_\tau, \\ -n_{t,\tau} & \text{if } i = de_\tau, \\ 0 & \text{else.} \end{cases} \tag{3.11}$$

In connection with flow problems, the constraint in equation (3.8) is called *flow conservation*. It ensures that the net flow of trains is zero for all nodes except for nodes representing start and destination station which work as source and sink nodes of the traffic flow. Routing costs c_{ij} per line and train are assumed to be constant values which are not further specified at this point. Equation (3.9) is called *capacity constraint*. For each line the capacity consumption of trains using a line $cc_{\mathrm{ti},ij} = x_{ij} z_{tt,ij}$ is forced to be less or equal than the maximum capacity consumption $max_{\mathrm{cc},ij}$, cf. Section 2.3.3.

Minimum cost flow problems with bounded integer capacities, like (3.7)–(3.10), always have *totally unimodular*[1] coefficient matrices. This results in the following nice property: such problems always have integral optimal solutions even if the integrality constraints are skipped or ignored, respectively. So, these problems can be solved as linear programming problems, which requires only polynomial time, cf. Section 3.1.1.

[1] A matrix is totally unimodular if each square submatrix has determinant 0 or ±1, cf. [AMO93].

3.3.1 Multicommodity Flow

To get rid of the artificial restriction R1, the NDRI-R model presented so far has to be extended to be able to deal with multiple traffic flows. Each traffic flow, also referred to as commodity, is then defined by its own start and destination station as well as by an amount of trains, cf. Definition 2.7. Restriction R2 and R3 remain in force. This leads to a special kind of network flow: the multicommodity flow (MCF). Following the MCF definition of Ahuja, Magnanti and Orlin [AMO93] a refined NDRI-R model is obtained:

$$\text{Minimize} \sum_{(i,j)\in A} \sum_{\tau \in TD} c_{ij} x_{ij}^{\tau} \tag{3.12}$$

subject to

$$\sum_{\tau \in TD} x_{ij}^{\tau} \leqslant max_{t,ij} \qquad \forall (i,j) \in A, \tag{3.13}$$

$$\sum_{j:(i,j)\in A} x_{ij}^{\tau} - \sum_{j:(j,i)\in A} x_{ji}^{\tau} = b_i^{\tau} \qquad \forall i \in N \text{ and } \forall \tau \in TD, \tag{3.14}$$

$$x_{ij}^{\tau} \geqslant 0 \qquad \forall (i,j) \in A \text{ and } \tau \in TD, \tag{3.15}$$

where $G = (N, A)$ is the network graph, TD is the set of traffic flows, x_{ij}^{τ} the number of trains of flow $\tau \in TD$ on arc (i, j) and c_{ij} the corresponding cost. General definitions for MCF are using commodity-specific costs c_{ij}^{τ}. That is dispensable for the current purpose. Since only one train type is used here (cf. R2), routing costs are assumed to be the same for all flows. Flow conservation constraint (3.14) and its right-hand side value b_i^{τ} (3.16) are defined analogous to (3.8) and (3.11) extended to fit for multiple traffic flows. For each node $i \in N$ and each traffic flow $\tau \in TD$ the value of b_i is defined in the following way:

$$b_i^{\tau} := \begin{cases} n_{t,\tau} & \text{if } i = st_{\tau}, \\ -n_{t,\tau} & \text{if } i = de_{\tau}, \\ 0 & \text{else.} \end{cases} \tag{3.16}$$

In contrast to minimum cost flow problems, the coefficient matrices of multicommodity flow problems are in general not totally unimodular. This holds for problem formulation (3.12)–(3.15), too. So, to preserve the integrality of flow variables x_{ij}^{τ}, integrality constraints have to be added.

Nevertheless, such integrality constraints are not included into the problem formulation because integral flows would render the problem to be NP-complete. In relation to practice, fractional trains obviously do not exist but with regard to the macroscopic model for infrastructure and operation, it is reasonable to lower the level of precision to benefit from the significant simplification induced by the use of non-integer variables.

3.3.2 Path Variables

Up to now network flows are defined on arcs. In the following, flows are defined on *paths*[2]. Path flows establish a direct connection to practice, since railway undertakings order railway infrastructure capacity in form of train paths, too. With the use of path flows it is easy to force the routing to use or avoid certain intermediate stations. Furthermore, it is easy to control, for example, journey times of traffic relations by neglecting paths which do not fit in this regard. In practice, similar restrictions lead to the sets of train paths which are of interest for infrastructure managers or railway undertakings, respectively.

In general, the transformation from arc flows to path flows suffers from an exponential number of available paths and in turn from an exponential number of path flow variables. This issue will be tackled later on in Section 6.1.6.

Switching from arc flows to path flows requires a function indicating whether a path contains a specific arc. For each arc (i, j) such function δ_{ij} is defined as follows:

$$\delta_{ij}(p) := \begin{cases} 1, & \text{if path } p \text{ contains arc } (i,j), \\ 0, & \text{else.} \end{cases} \tag{3.17}$$

This definition now permits to reformulate the model stated in (3.12)-(3.15) using path flows. For each traffic flow $\tau = (st_\tau, de_\tau, tr_\tau)$ let $\mathcal{P}^\tau$ denote the set of all (desired) directed *st*-*de*-paths, $\mathcal{P} := \bigcup\limits_{\tau \in TD} \mathcal{P}^\tau$ the set of all paths, and $f(p)$ the flow on path $p \in \mathcal{P}$. Arc flow x_{ij}^τ is transformed to path flow $f(\mathrm{p})$ as follows:

$$x_{ij}^\tau = \sum_{p \in \mathcal{P}^\tau} \delta_{ij}(p) f(p) \tag{3.18}$$

[2]A *path* in a directed network graph is a sequence of nodes and arcs without any repetition of nodes.

Equations (3.19)–(3.22) show the transformed optimization model.

$$\text{Minimize} \sum_{(i,j)\in A} \sum_{p\in\mathcal{P}} c_{ij}\delta_{ij}(p)f(p) \tag{3.19}$$

subject to

$$\sum_{p\in\mathcal{P}} \delta_{ij}(p)f(p) \leqslant max_{t,ij} \qquad \forall(i,j) \in A, \tag{3.20}$$

$$\sum_{p\in\mathcal{P}^\tau} f(p) = n_{t,\tau} \qquad \forall\tau = (st_\tau, de_\tau, \{n_{t,\tau}\}) \in TD, \tag{3.21}$$

$$f(p) \geqslant 0 \qquad \forall\tau \in TD \text{ and } p \in \mathcal{P}^\tau. \tag{3.22}$$

Now, the former flow conservation constraint (3.14) is transformed to a node independent representation of the traffic demand (3.21).

Before the routing model will be further developed, the idea behind the solution approach for the design problem is introduced.

3.4 The Design Problem

The design part NDRI-D of the network design problem for railway infrastructure deals with the questions which stations have to be connected directly to each other and which stage of extension fits best in terms of capacity and cost. Due to restriction R3, there is only one selectable type of arc with a fixed capacity profile and cost. So, at first, network design is only dealing with the determination of the minimum cost network topology.

The idea is to solve a routing problem on a *complete* network graph. Such complete network graph contains an arc for each pair of nodes. The *best* routing over such network graph then determines the design of the target network by including exactly those arcs into the network which are needed to route the given traffic demand and thus provide the necessary capacity.

Summed-up line life-cycle costs of included lines are interpreted as routing costs. Such new routing costs are load-independent, because a line is included into the network and built, respectively, regardless of the number of trains routed via the line. This type of routing cost definition ensures that the cost-optimal routing results in a cost-optimal network graph.

In the following the multicommodity flow model introduced in (3.19)–(3.22) is extended to fit for the design tasks described above.

3.5 Simple Model

To specify whether an arc is included into the network, *used*-variables $x_{ij}^{\text{used}} \in \{0,1\}$ (3.27) are introduced for each $(i,j) \in A$, where $G = (N, A)$ is the given complete network graph. Routing costs are replaced by life-cycle cost which arise if an arc is included into the network (3.23). Line life-cycle costs cost_{ij} are defined regarding Definition 2.5 and Equation (2.14). Equation (3.24) is the new capacity constraint which limits the capacity consumption of trains on line (i,j) to the maximum value permitted or zero, respectively, depending on the decision to include or not to include the corresponding arc into the network graph (i.e. assigning the value 1 or 0 to x_{ij}^{used}).

$$\text{Minimize} \sum_{(i,j)\in A} \text{cost}_{ij} x_{ij}^{\text{used}} \tag{3.23}$$

subject to

$$\sum_{p\in\mathcal{P}} \delta_{ij}(p) f(p) \leqslant max_{t,ij} x_{ij}^{\text{used}} \qquad \forall (i,j) \in A, \tag{3.24}$$

$$\sum_{p\in\mathcal{P}^\tau} f(p) = n_{t,\tau} \qquad \forall \tau = (st_\tau, de_\tau, \{n_{t,\tau}\}) \in TD, \tag{3.25}$$

$$f(p) \geqslant 0 \qquad \forall \tau \in TD \text{ and } p \in \mathcal{P}^\tau. \tag{3.26}$$

$$x_{ij}^{\text{used}} \in \{0,1\} \qquad \forall (i,j) \in A, \tag{3.27}$$

where $G = (N, A)$ is a network of railway infrastructure and t is a train type.

3.6 NDRI-NLP

Lifting restrictions R2 and R3 now requires subsequent refinements of the model. At first, R3 is lifted. Different stages of extension of a line, which are characterized by different capacity profiles, are represented by parallel arcs. This leads to a *complete multi-arc network graph.*

Definition 3.1 (Complete Multi-Arc Network Graph)**.** For a given set of nodes N let $A \subseteq N \times N$ be the set of all directed lines or station-to-station connections, respectively, whose potential construction is reasonable to the network designer[3]. The complete multi-arc network graph G^{c} is then defined as

$$G^{\mathrm{c}} := (N, A^{\mathrm{m}}),$$

where A^{m} is a set of multi-arcs. Since a line can be available in up to ten different stages of extension (cf. Equation 2.9), A^{m} contains up to ten parallel multi-arcs, each of them with a different combination of capacity profile and number of tracks, for each arc $(i, j) \in A$.

$$A^{\mathrm{m}} := \{(i, j, s) \mid (i, j) \in A, s \in S_{ij}\}.$$

The triple $(i, j, s) \in A^{\mathrm{m}}$ is called multi-arc. If a line needs to be addressed regardless of its stage of extension, the line is simply referred to as arc $(i, j) \in A$.

The routing-driven arc selection applied to multi-arcs now determines not only the network topology, but also the capacity of the lines. To achieve that, only one of the parallel multi-arcs may be selected. Its unique pair of capacity profile and number of tracks then determines the stage of extension of the corresponding line. The use of multi-arcs requires the introduction of new design decision variables x_{ijs}^{used} for each $(i, j, s) \in A^{\mathrm{m}}$. Decision variables x_{ij}^{used} are still in use. For each arc $(i, j) \in A$ these variables indicate whether node i has to be linked to node j independent of the implementation of that linkage, i.e. independent of the multi-arc selection. Furthermore, these variables are used to model minimum or basic costs $cost_{ij,\mathrm{min}}$ of an arc, defined in Equation (3.37). Minimum costs are used to improve the solution process or branch-and-bound, respectively. Once the decision is made to connect two nodes i and j variable x_{ij}^{used} is set to one. This induces minimum costs $cost_{ij,\mathrm{min}}$, see Equation (3.29). So independent of the selection of a corresponding multi-arc (i, j, s), one obtains a lower bound for the final costs of that connection.

Lifting restriction R2 results in a more sophisticated calculation of the capacity con-

[3]Reasonability may, for example, depend on geographical constraints.

sumption of a mix of trains using a line. Here, the timetable independent capacity consumption introduced in Equation (2.21) is used:

$$cc_{\mathrm{ti},ijs} = nt_{ij} \sum_{t \in T} \sum_{u \in T} \frac{n_{t,ij} \cdot n_{u,ij}}{nt_{ij}^2} z_{tu,ijs}. \tag{3.28}$$

Furthermore, different types of trains within a traffic flow require an extension of path flows. Now, $f_t(\mathrm{p})$ denotes the number of trains of type t on path p. The variable indicating the demand of trains is adapted, similarly. b_t^τ denotes the amount of trains of type t which have to be routed regarding traffic flow τ.

For complete multi-arc network graph $G = (N, A^{\mathrm{m}})$, a set of train types T, and a set of traffic flows TD, Equations (3.29)–(3.34) show the optimization model which results from lifting restriction R2 and R3.

$$\text{Minimize} \quad \sum_{(i,j) \in A} cost_{ij,\min} \cdot x_{ij}^{\mathrm{used}} + \sum_{(i,j,s) \in A^{\mathrm{m}}} (cost_{ijs} - cost_{ij,\min}) \cdot x_{ijs}^{\mathrm{used}} \tag{3.29}$$

subject to

$$nt_{ij} \sum_{t \in T} \sum_{u \in T} \frac{F_{t,ij} \cdot F_{u,ij}}{nt_{ij}^2} z_{tu,ijs} \leqslant max_{\mathrm{cc},ijs} x_{ijs}^{\mathrm{used}} + M \sum_{r \neq s} x_{ijr}^{\mathrm{used}} \qquad \forall (i,j,s) \in A^{\mathrm{m}}, \tag{3.30}$$

$$\sum_{s:(i,j,s) \in A^{\mathrm{m}}} x_{ijs}^{\mathrm{used}} = x_{ij}^{\mathrm{used}} \qquad \forall (i,j) \in A, \tag{3.31}$$

$$\sum_{p \in \mathcal{P}^\tau} f_t(p) = n_{t,\tau} \qquad \forall \tau \in TD, \forall t \in T \tag{3.32}$$

$$x_{ij}^{\mathrm{used}} \in \{0,1\} \qquad \forall (i,j) \in A, \tag{3.33}$$

$$x_{ijs}^{\mathrm{used}} \in \{0,1\} \qquad \forall (i,j,s) \in A^{\mathrm{m}}, \tag{3.34}$$

where

$$F_{t,ij} := \sum_{p \in \mathcal{P}} \delta_{ij}(p) f_t(p), \tag{3.35}$$

$$nt_{ij} := \sum_{t \in T} F_{t,ij}. \tag{3.36}$$

For each arc $(i,j) \in A$ variable $cost_{ij,\min}$ is defined as follows:

$$cost_{ij,\min} := \min\{cost_{ijs} \mid \forall s \colon (i,j,s) \in A^{\mathrm{m}}\}. \tag{3.37}$$

For each arc $(i, j) \in A$ there is a capacity constraint for each of the corresponding multi-arcs $(i, j, s) \in A^{\mathrm{m}}$. Since at most one multi-arc is selectable, only the capacity constraint which corresponds to the selected multi-arc should be in force (activated) whereas the remaining constraints should be idle (deactivated). As already introduced for the problem formulation (3.23)–(3.27), the activation of capacity constraints can be implemented using binary *used*-variables. A simple transfer of this concept to multi-arc network graphs would lead to infeasible models. Why?

Assuming that capacity constraints for each multi-arc (i, j, s) are, analogous to (3.24), defined in the following way:

$$cc_{\mathrm{ti},ijs} \leqslant max_{\mathrm{cc},ijs} x_{ijs}^{\mathrm{used}} \tag{3.38}$$

Furthermore, assuming that some traffic flow is routed via an arc (i, j), then the capacity consumption $cc_{\mathrm{ti},ijs}$ of all corresponding multi-arcs is a positive value ($cc_{\mathrm{ti},ijs} > 0$). Let $x_{ijs_0}^{\mathrm{used}} = 1$ and $x_{ijs_1}^{\mathrm{used}} = 0$ with $s_1 \neq s_0$. Corresponding capacity constraints then have the following form:

$$cc_{\mathrm{ti},ijs_0} \leqslant max_{\mathrm{cc},ijs_0}, \tag{3.39}$$

$$cc_{\mathrm{ti},ijs_1} \leqslant 0, \tag{3.40}$$

which is in conflict with $cc_{\mathrm{ti},ijs_1} > 0$. To overcome this, *used*-variables are used within a so-called *big-M approach*. The idea is that constraints are deactivated by making them redundant or dispensable for the problem formulation. This is achieved by adding a constant value M to the right-hand side of that constraint. M has to be chosen in a way that it serves as trivial upper bound for the capacity consumption. So, the activated constraint provides the desired upper bound whereas deactivated constraints just provide trivial upper bounds. To achieve that, M is linked to the decision variables in a way which ensures that M is activated if the constraint is deactivated and vice versa. The resulting set of capacity constraint is shown in (3.30). The following example shows, how this mechanism works:

Example 3.1 (Big M)**.** Let (i, j, s_0) and (i, j, s_1) the only two parallel multi-arcs of arc (i, j) and 6 the maximal value of capacity, which is consumable. Values 7 and 5 are assumed to be the capacity consumption of a given mix of trains if using (i, j, s_0) and (i, j, s_1), respectively. The crucial constraints can be stated as follows:

$$7 \leqslant 6 \cdot x_{ijs_0}^{\text{used}} + M \cdot x_{ijs_1}^{\text{used}}, \tag{3.41}$$

$$5 \leqslant 6 \cdot x_{ijs_1}^{\text{used}} + M \cdot x_{ijs_0}^{\text{used}}, \tag{3.42}$$

$$x_{ijs}^{\text{used}} + x_{ijs_1}^{\text{used}} = x_{ij}^{\text{used}}, \tag{3.43}$$

$$x_{ij}^{\text{used}}, x_{ijs_0}^{\text{used}}, x_{ijs_1}^{\text{used}} \in \{0, 1\}. \tag{3.44}$$

It is easy to see, that $x_{ij}^{\text{used}} = x_{ijs_1}^{\text{used}} = 1$ and $x_{ijs_0}^{\text{used}} = 0$ is the only valid assignment of values and that the system on inequalities is infeasible without the usage of M-terms. Here, the value of M has to be 7 or greater. The way how to determine suitable values for M is presented later on.

Because of the non-linear capacity constraint (3.30), Equations (3.29)–(3.34) constitute a non-linear optimization model of NDRI. Therefore, this model is denoted by NDRI-NLP. To benefit from powerful solvers for linear optimization problems, NDRI-NLP is transformed to a mixed-integer program NDRI-MIP.

3.7 NDRI-MIP

The idea behind the transformation from NDRI-NLP to NDRI-MIP is based on the well-known *cutting-stock problem*, which is described by Chvátal [Chv83] as follows.

> Materials such as papers, textiles, cellophane, and metallic foil are manufactured in rolls of large widths. These rolls, referred to as rows, are later cut into rolls of small widths, called finals. Each manufacturer produces raws of a few standard widths; the widths of the finals are specified by different customers and may vary widely. The cutting is done on machines by knives that slice through the rolls in much the same way as a knife slices bread. For example, a raw that is 100 inch wide may be cut into two finals with 31-inch widths and one final with a 36-inch width, with 2 inch left over going into waste. When a complicated summary of orders has to be filled, the most economical way of cutting the existing raws into desired finals is rarely obvious. The problem of finding such a way is known as the *cutting-stock problem*.

In this context, the concept of a *cutting pattern* is of importance. For a given width of raws and a given set of widths of finals, a cutting pattern describes one way to cut a raw into finals. Referring to the example given in the problem description by Chvátal, one cutting pattern of the 100-inch raw consists of two 31-inch finals and one 36-inch final. Another one would, for example, consist just of three 31-inch finals. An easy but extensive way to solve the cutting-stock problem is to list all cutting patterns and determine the number of raws which has to be cut according each cutting pattern to fulfill the given order of finals. The concept of cutting patterns shall now be transferred to NDRI with the objective to get rid of the non-linear capacity constraints. A cutting pattern in the context of NDRI is referred to as *configuration*.

Definition 3.2 (Configuration)**.** Given a complete multi-arc network graph $G^{\mathrm{c}} = (N, A^{\mathrm{m}})$ and a set of train types T. For an arc $(i, j, s) \in A^{\mathrm{m}}$

i) a *configuration* $c_{ijs} := (c_{1,ijs}, c_{2,ijs}, \ldots, c_{|T|,ijs}) \in \mathbb{N}^{|T|}$ is a $|T|$-tuple, which components fulfill the following property:

$$\sum_{t \in T} \sum_{u \in T} \frac{c_{t,ijs} \cdot c_{u,ijs}}{\sum_{t \in T} c_{t,ijs}} z_{tu,ijs} \leqslant max_{\mathrm{cc},ijs},$$

ii) $\mathcal{C}_{ijs,\mathrm{all}}$ is the set of all configurations, and

iii) $\mathcal{C}_{ijs} := \{c_{ijs} \in \mathcal{C}_{ijs,\mathrm{all}} \mid \nexists d_{ijs} \in \mathcal{C}_{ijs,\mathrm{all}} (d_{ijs} \neq c_{ijs}) \wedge (\forall t \in T (c_{t,ijs} \leqslant d_{t,ijs}))\}$ is the set of all *maximal* configurations. Maximal configurations describe the train/flow type mix ratios which fully utilize the capacity of the corresponding multi-arc.

The maximal configurations now are used to get rid of non-linear capacity constraints (3.30). If a multi-arc is included into the network, the corresponding *design*-variable x^{used}_{ijs} is set to one. In addition to that, a maximal configuration has to be selected. The components of the configuration bound the current traffic flow on the corresponding arcs separately for each train type. To identify whether a configuration is selected, binary *config*-variables $y^{\mathrm{used}}_{c_{ijs}}$ are introduced for each maximal configuration $c_{ijs} = (c_{1,ijs}, c_{2,ijs}, \ldots, c_{|T|,ijs}) \in \mathcal{C}_{ijs}$ of each arc $(i, j, s) \in A^{\mathrm{m}}$. Equation (3.46) shows the new capacity constraint, $max_{t,ijs}$ is the maximum train count of trains of type t routable via arc (i, j, s), cf. $max_{t,ij}$ in (2.25). Equation (3.48) ensures that only one configuration can be selected and that a selection is enabled if and only if the corresponding multi-arc is included into the network.

$$\text{Minimize} \sum_{(i,j)\in A} cost_{ij,\min} \cdot x_{ij}^{\text{used}} + \sum_{(i,j,s)\in A^{\text{m}}} (cost_{ijs} - cost_{ij,\min}) \cdot x_{ijs}^{\text{used}} \tag{3.45}$$

subject to

$$\sum_{p\in\mathcal{P}} \delta_{ij}(p) f_t(p) \leqslant \sum_{s:(i,j,s)\in A^{\text{m}}} \sum_{c_{ijs}\in\mathcal{C}_{ijs}} c_{t,ijs} y_{c_{ijs}}^{\text{used}} \qquad \forall (i,j)\in A, \forall t\in T, \tag{3.46}$$

$$\sum_{s:(i,j,s)\in A^{\text{m}}} x_{ijs}^{\text{used}} = x_{ij}^{\text{used}} \qquad \forall (i,j)\in A, \tag{3.47}$$

$$\sum_{c_{ijs}\in\mathcal{C}_{ijs}} y_{c_{ijs}}^{\text{used}} = x_{ijs}^{\text{used}} \qquad \forall (i,j,s)\in A^{\text{m}}, \tag{3.48}$$

$$\sum_{p\in\mathcal{P}^{\tau}} f_t(p) = n_{t,\tau} \qquad \forall \tau\in TD, \forall t\in T, \tag{3.49}$$

$$x_{ij}^{\text{used}} \in \{0,1\} \qquad \forall (i,j)\in A, \tag{3.50}$$

$$x_{ijs}^{\text{used}} \in \{0,1\} \qquad \forall (i,j,s)\in A^{\text{m}}, \tag{3.51}$$

$$y_{c_{ijs}}^{\text{used}} \in \{0,1\} \qquad \forall c_{ijs}\in\mathcal{C}_{ijs}. \tag{3.52}$$

The usage of maximal configurations to linearize the capacity constraint of NDRI-NLP creates two problems concerning an efficient solution process of the resulting mixed integer programming problem NDRI-MIP.

i) A large number of maximal configurations results in a high number of binary *config*-variables $y_{c_{ijs}}^{\text{used}}$. This results in large branch-and-bound trees, which in turn could result in high runtime complexities.

ii) Configuration variables induce a *symmetry* in the problem.

First problem is handled by a solution technique called *column generation*, presented in Chapter 5. Second problem is discussed in the following section. Last section of this chapter introduces a new linear model which avoids problems caused by the linearization of NDRI-NLP using configurations.

3.8 Symmetry

Besides their numerousness, configuration variables create another problem. They induce a *symmetry* in the problem in sense that for feasible solutions it is often possible to vary in the assignment of values to *config*-variables without changing feasibility or cost. Such solutions are known as *isomorphic* solutions. Especially in cases of small traffic flows, many maximal configurations can serve as upper bound of the current train type mix ratios. The consequence of this is an inefficiency of the branch-and-price algorithm, because many isomorphic solutions repeatedly appear while traversing the B&P-tree. A very nice overview about symmetry in integer linear programming, including among other topics symmetry detecting and symmetry breaking techniques, is given by Margot [Mar10]. One approach to get rid of the symmetry presented in the paper is the generation of *symmetry breaking inequalities* which are added to the problem formulation. In this regard, dynamic and static symmetry breaking inequalities are distinguished by Margot. Static symmetry breaking inequalities are added to the initial problem formulation to cut symmetric solutions. Dynamic symmetry breaking inequalities seek the same objective but are added during the solution process and may be not valid for the initial formulation. It is important to mention that symmetry breaking inequalities do affect the feasible region of the problem by cutting off symmetric feasible solutions.

The symmetry problem of NDRI-MIP has not been solved within this work, but seems to be a promising field of future research.

3.9 NDRI-MIP$^{\text{WC}}$

Up to this point, capacity consumption of trains operating on a line is modeled under the assumption of a timetable independent succession of trains. The linear program NDRI-MIP$^{\text{WC}}$, defined by Equations (3.53)–(3.62), is based on a timetable dependent derivation of line capacity consumption. The underlying succession of trains follows the principle of a worst-case timetable, cf. Section 2.3.3. So the network designed by means of worst-case capacity consumptions provides an upper bound on the infrastructure needed to satisfy the given traffic demand.

$$\text{Minimize} \sum_{(i,j)\in A} cost_{ij,\min} \cdot x_{ij}^{\text{used}} + \sum_{(i,j,s)\in A^{\text{m}}} (cost_{ijs} - cost_{ij,\min}) \cdot x_{ijs}^{\text{used}} \tag{3.53}$$

subject to

$$\alpha^0_{ijs} F_{t_0,ij} + \alpha^1_{ijs} F_{t_1,ij} \leqslant max_{\mathrm{cc},ijs} x^{\mathrm{used}}_{ijs} + M w_{ij} + M' \sum_{r \neq s} x^{\mathrm{used}}_{ijr} \quad \forall (i,j,s) \in A^{\mathrm{m}}, \tag{3.54}$$

$$\alpha^2_{ijs} F_{t_0,ij} + \alpha^3_{ijs} F_{t_1,ij} \leqslant max_{\mathrm{cc},ijs} x^{\mathrm{used}}_{ijs} + M''(1 - w_{ij}) + M''' \sum_{r \neq s} x^{\mathrm{used}}_{ijr} \quad \forall (i,j,s) \in A^{\mathrm{m}}, \tag{3.55}$$

$$F_{t_1,ij} \leqslant F_{t_0,ij} + M_{t_1} w_{ij} \quad \forall (i,j) \in A, \tag{3.56}$$

$$F_{t_0,ij} \leqslant F_{t_1,ij} + M_{t_0}(1 - w_{ij}) \quad \forall (i,j) \in A, \tag{3.57}$$

$$\sum_{s:(i,j,s) \in A^{\mathrm{m}}} x^{\mathrm{used}}_{ijs} = x^{\mathrm{used}}_{ij} \quad \forall (i,j) \in A, \tag{3.58}$$

$$\sum_{p \in \mathcal{P}^\tau} f_{t_0}(p) = n_{t_0,\tau} \quad \forall \tau \in TD, \tag{3.59}$$

$$\sum_{p \in \mathcal{P}^\tau} f_{t_1}(p) = n_{t_1,\tau} \quad \forall \tau \in TD, \tag{3.60}$$

$$x^{\mathrm{used}}_{ijs} \in \{0,1\} \quad \forall (i,j,s) \in A^{\mathrm{m}}, \tag{3.61}$$

$$x^{\mathrm{used}}_{ij}, w_{ij} \in \{0,1\} \quad \forall (i,j) \in A, \tag{3.62}$$

where

$$\alpha^0_{ijs} := z_{t_0 t_0, ijs} \tag{3.63}$$

$$\alpha^1_{ijs} := z_{t_1 t_0, ijs} + z_{t_0 t_1, ijs} - z_{t_0 t_0, ijs} \tag{3.64}$$

$$\alpha^2_{ijs} := z_{t_1 t_0, ijs} + z_{t_0 t_1, ijs} - z_{t_1 t_1, ijs} \tag{3.65}$$

$$\alpha^3_{ijs} := z_{t_1 t_1, ijs} \tag{3.66}$$

For NDRI-MIP$^{\mathrm{WC}}$, the number of train types is restricted to two different types t_0 and t_1 (w. l. o. g $v_{t_0} > v_{t_1}$), cf. Section 2.3.3. This results in two different instances of a worst-case timetable and two different calculation rules for the capacity consumption, see (2.22) and (2.23), respectively. For each arc, the choice of a calculation rule depends on the ratio of proportion of train types in current traffic flow. If $F_{t_0,ij} \geqslant F_{t_1,ij}$, capacity consumption is calculated on the basis of (2.22). Equation (2.23) is used if $F_{t_1,ij} \geqslant F_{t_0,ij}$. To model such decision process, decision variables w_{ij} are introduced in (3.62). In combination with a *big M*, those variables are used as „switches“ to enable and disable constraints which are involved in the current decision. If $w_{ij} = 0$, constraint (3.56) is enabled and forces the amount of trains of type t_1 to be less or equal than the amount of trains of type t_0 ($F_{t_0,ij} \geqslant F_{t_1,ij}$). On the other hand, inequality (3.57) is disabled, because it just provides a trivial upper bound on $F_{t_0,ij}$. M_{t_0} and M_{t_1} have to be sufficiently large to serve this purpose. In the same way capacity constraints (3.54) and (3.54) are enabled and disabled, respectively.

The derivation of suitable values for $M_{t_0}, M_{t_1}, M, M', M''$, and M''' is addressed in Section 5.3.

3.10 Conclusions & Next Steps

After the introduction of models for railway infrastructure, operation, and capacity the network design problem for railway infrastructure is modeled as non-linear multi-commodity flow problem on a complete multi-graph using timetable independent capacity consumption. To linearize the model, configurations are introduced. Despite the difficulties concerning symmetries and the problem size, which are induced by the config-variables, NDRI-MIP has the advantage to be a very flexible model. On the one hand, the capacity function which models the capacity consumption is interchangeable. NDRI-MIP only deals with configurations independent of the way they are created. So instead of expected service times, every other measure may be used, e.g., scheduled and unscheduled waiting times, which are state of the art of analytic models of railway operations research in Germany. On the other hand, the number of different train types can be increased without fundamentally changing the model. However, the number of config-variables will increase significantly.

NDRI-MIP$^{\mathrm{WC}}$ represents a different approach to obtain a linear model for NDRI: worst-case timetables. Problems caused by configurations are avoided. A disadvantage lies in the problem formulation since worst-case timetables are hard coded into the capacity constraints. Changes of timetables implicate changes of the constraints. Here, two different train types are used, which leads to two different worst-case timetables and in turn to two capacity constraints for each arc. Worst-case timetables for $|T| > 2$ are not considered here, cf. Section 2.3.3.

Advantages and disadvantages of both models are evaluated in detail later on in Chapter 6. Besides results of the evaluation, this chapter encompasses a description of the evaluation framework and a problem size analysis, too. Beforehand, Chapters 4 and 5 cover different aspects concerning model refinements, implementation, and solving of NDRI models NDRI-MIP and NDRI-MIP$^{\mathrm{WC}}$. In order to solve problem instances of these models, state-of-the-art solvers for mixed integer programming problems are used.

They are shortly introduced in Section 5.4. Such solvers are very powerful since they are well developed over decades. Nevertheless, for large numbers of variables, especially for integer ones, solvers are pushed to their limits. As already addressed in Section 3.7, the NDRI-MIP model is faced with the problem to be composed of a large number of binary *config*-variables. The problem is handled by column generation, which is introduced in Section 5.1.1 of Chapter 5. Besides MIP solution techniques that chapter is focussing on *fixing of variables* and the derivation of suitable *big-M* values. Another more fundamental problem for MIP solvers, even for small problem instances, is the so-called *integrality gap*, which denotes the ratio between the optimal solution of an integer problem and the optimal solution of the relaxation of that problem. Large gaps imply bad bounds during branch-and-bound, which in turn do imply an inefficient search for integer solutions. To reduce that gap, *valid inequalities* are added to the problem formulations of NDRI-MIP and NDRI-MIP$^{\mathrm{WC}}$. Chapter 4 covers that topic.

4 Valid Inequalities

In (mixed) integer linear programming a *valid inequality* or (*valid*) *cut* is a constraint, which tightens the feasible region of the LP relaxation to achieve a better approximation of the (M)IP feasible region, which itself remains unaffected by the cut. Mostly, cuts are generated using so-called *cutting plane algorithms*. Input is the linear relaxation of an integer program which is at first solved to optimality. If the solution is integer, the algorithm terminates. Otherwise a *separation problem* has to be solved, i.e. finding an inequality which separates the optimal LP solution from the IP feasible region. Whereas the embedding of a cutting plane algorithm requires elaborate modifications of the branch-and-bound algorithm, i.e. developing a *branch-and-cut* algorithm, valid inequalities can be added *a priori* or *offline* to the problem formulation, too, cf. Wolsey [Wol98]. Such an offline approach is used in this work. A branch-and-cut algorithm or, to be more precisely, a branch-and-price-and-cut algorithm could be topic of future research. In contrast to the *online* cut generation of cutting plane algorithms, the usage of an offline approach provides the possibility to tighten the problem formulation before branch-and-bound is applied. This allows the use of high-end solver software, which usually does not provide an interface to modify the embedded branch-and-bound algorithms. The offline method has the disadvantage that cuts are not added on demand. So to achieve the tightening effect of the online method, probably a huge number of valid inequalities have to be added, which in turn could lead to a significant blow-up of solving times. The next sections are dealing with cuts, which are added to the NDRI models.

4.1 Single Node Cuts

Single node cuts which are added to both models NDRI-MIP and NDRI-MIP$^{\mathrm{WC}}$ are used to describe the following finding.

Without knowing their routing, the set of given traffic flows implicitly contains some information about the number of arcs which at least have to be included into the network. This comes from the fact that nodes need a minimum number of adjacent arcs to be able to accomplish their tasks as sources and sinks of traffic flows.

The goal of these cuts is to improve bounds on the objective value of the MIP, i.e. the design costs, which are gained by LP relaxations during branch-and-bound. The way in which this is achieved by single node cuts is very intuitive. The number of arcs which is required at least is expressed by the minimum number *used*-variables which have to take the value one. This in turn determines minimum design costs, since the objective function only consists of *used*-variables and corresponding cost coefficients are positive.

Let $G^{\mathrm{c}} = (N, A^{\mathrm{m}})$ be a complete network graph and $n \in N$ a single node. A single node cut specifies either the minimum number of ingoing arcs $a_{\mathrm{in},n}$ or outgoing arcs $a_{\mathrm{out},n}$ of n, which are required to route ingoing and outgoing sets of traffic flows $f_{\mathrm{in},n}$ and $f_{\mathrm{out},n}$.

Depending on the determination of $f_{\mathrm{in},n}$ and $f_{\mathrm{out},n}$, two types of single node cuts are distinguished here.

i) Numbers of ingoing and outgoing arcs a_{in,de_τ} and a_{out,st_τ} are determined separately <u>for each traffic flow</u> $\tau = (st_\tau, de_\tau, tr_\tau) \in TD$, which means that

$$f_{\mathrm{in},de_\tau} := \{\tau\} \text{ and } f_{\mathrm{out},st_\tau} := \{\tau\}.$$

Corresponding cuts are denoted by MIS (<u>m</u>inimum <u>i</u>ngoing arcs required for a <u>s</u>ingle flow) and MOS (<u>m</u>inimum <u>o</u>utgoing arcs required for a <u>s</u>ingle flow).

ii) Numbers of ingoing and outgoing arcs $a_{\mathrm{in},n}$ and $a_{\mathrm{out},n}$ are determined separately <u>for each node</u> n. Traffic flows with $de = n$ and $st = n$, respectively, are aggregated to sets of in-flows and out-flows:

$$f_{\mathrm{in},n} := \{(st_\tau, de_\tau, tr_\tau) \in TD \mid de_\tau = n\} \text{ and } f_{\mathrm{out},n} := \{(st_\tau, de_\tau, tr_\tau) \in TD \mid st_\tau = n\}.$$

Corresponding cuts are denoted by MIM (<u>m</u>inimum <u>i</u>ngoing arcs required for <u>m</u>ultiple flows) and MOM (<u>m</u>inimum <u>o</u>utgoing arcs required for <u>m</u>ultiple flows).

Before MIS & MOS and MIM & MOM are introduced in detail, an important quantity is introduced. Let $A^{\mathrm{m}}_{\mathrm{in},n}$,$A^{\mathrm{m}}_{\mathrm{out},n} \subseteq A^{\mathrm{m}}$ be the sets of ingoing and outgoing multi-arcs of node $n \in N$. The sets of ingoing and outgoing arcs, which are available to sets of traffic flows $f_{\text{in},n}$ and $f_{\text{out},n}$ are then defined by (4.1) and (4.2). An arc is available to a set of ingoing or outgoing traffic flows of a node if it is element of the ingoing or outgoing arcs and element of some path of at least one traffic flow.

$$A^{\text{m}}_{f_{\text{in},n}} := \bigcup_{\tau \in f_{\text{in},n}} \{(i,j,s) \in A^{\text{m}}_{\text{in},n} \mid \exists p \in \mathcal{P}^{\tau} \colon \delta_{ij}(p) = 1\} \tag{4.1}$$

$$A^{\text{m}}_{f_{\text{out},n}} := \bigcup_{\tau \in f_{\text{out},n}} \{(i,j,s) \in A^{\text{m}}_{\text{out},n} \mid \exists p \in \mathcal{P}^{\tau} \colon \delta_{ij}(p) = 1\} \tag{4.2}$$

4.1.1 MIS & MOS

In the following, the derivation of the MIS cut is described in detail. The derivation of the MOS cut is handled in line with that and cause of that is not described here.

For each traffic flow of the traffic demand MIS specifies the number of multi-arcs which have to be included into the network, i.e. their corresponding *used*-variables have to be set to 1, to be able to route at least the current flow into the sink node.

$$\text{MIS:} \qquad \forall (st_\tau, de_\tau, tr_\tau) \in TD\colon \sum_{(i,j,s) \in A^{\text{m}}_{f_{\text{in},de_\tau}}} n_{\text{tk},ijs} \cdot x^{\text{used}}_{ijs} \geqslant a_{\text{in},de_\tau} \tag{4.3}$$

The reason why the *used*-variables are multiplied by the number of tracks is explained later on. To determine the value of $a_{\text{in},n}$ an integer optimization problem has to be solved. This is described in following for both models NDRI-MIP and NDRI-MIP$^{\text{WC}}$.

For NDRI-MIP the minimum number of ingoing arcs a_{in,de_τ} which have to be included into the network for each traffic flow $\tau = (st_\tau, de_\tau, tr_\tau)$ is derived by solving (3.45)–(3.52) with the difference that the underlying complete network graph is restricted to arcs $A^{\text{m}}_{f_{\text{in},de_\tau}}$ and the corresponding set of nodes. The traffic demand is restricted to the given traffic flow τ.

Let $x^{\text{used}*}_{ijs}$ be the used variables of an optimal solution of the restricted problem, then $a_{\text{in},n}$ is defined by

$$a_{\text{in},n} := \sum_{(i,j,s) \in A^{\text{m}}_{f_{\text{in},n}}} n_{\text{tk},ijs} \cdot x^{\text{used}*}_{ijs} \tag{4.4}$$

Factor $n_{\text{tk},ijs}$ is added to handle effects which occur from the fact that a multi-arc (i,j,s) either represents a single or double tracked line. For the same reason each addend of the MIS cut is multiplied with such a factor, too. Assuming that $a_{\text{in},n}$ is calculated without the use of $n_{\mathrm{tk},ijs}$ and the optimal solution consists of only one double tracked line. This would lead to $a_{\mathrm{in},n} = 1$. Which in turn implies that the MIS inequality is satisfied for a single tracked line, even if it is to small to route the current in-flow. That would not change the feasibility of the problem formulation but the MIS cut would be less tight.

For NDRI-MIP$^{\mathrm{WC}}$ the value of a_{in,de_τ} is determined by solving (3.53)–(3.62) with same restrictions described before in case of NDRI-MIP.

4.1.2 MIM & MOM

The derivation of MIM and MOM cuts is explained using the MOM cut as an example. The derivation of MIM cuts is handled in line with that, but is not presented here.

The MOM cut specifies the minimum number of required outgoing arcs of a node with respect to all out-flows. MOM cuts have the same shape as MOS cuts. They differ in the value of $f_{\mathrm{out},n}$, cf. enumeration in the beginning of Section 4.1, and, as a result of that, they differ in the fact that they are included into the problem formulation just once for each node.

$$\text{MOM:} \qquad \forall n \in N\colon \sum_{(i,j,s)\in A^{\mathrm{m}}_{f_{\mathrm{out},n}}} n_{\mathrm{tk},ijs} \cdot x^{\mathrm{used}}_{ijs} \geqslant a_{\mathrm{out},n} \tag{4.5}$$

The derivation of $a_{\mathrm{out},n}$ follows the same principles used for MIS and MOS introduced in Section 4.1.1.

4.1.3 MIS & MOS versus MIM & MOM

The question may be raised why both kinds of single node cuts are used here, because there seems to be redundancy in the information they provide. Generally, that is not the case.

MIS and MOS or MIM and MOM cuts both improve the feasible region of the LP relaxation, but depending on the very own structure of a problem instance either MIS and MOS or MIM and MOM cuts lead to better improvements, cf. Figures 4.1 and 4.2. Before the solving starts, it is not decidable which type of cut fits better for the current problem instance. So, it is reasonable to add both cut types to the problem formulation.

Figure 4.1 shows an example in which the MOS cuts provide a better approximation of the IP feasible region than the MOM cut. Figure 4.2 shows the opposite case. Except of the demand structure, both cases are based on the same problem instance which

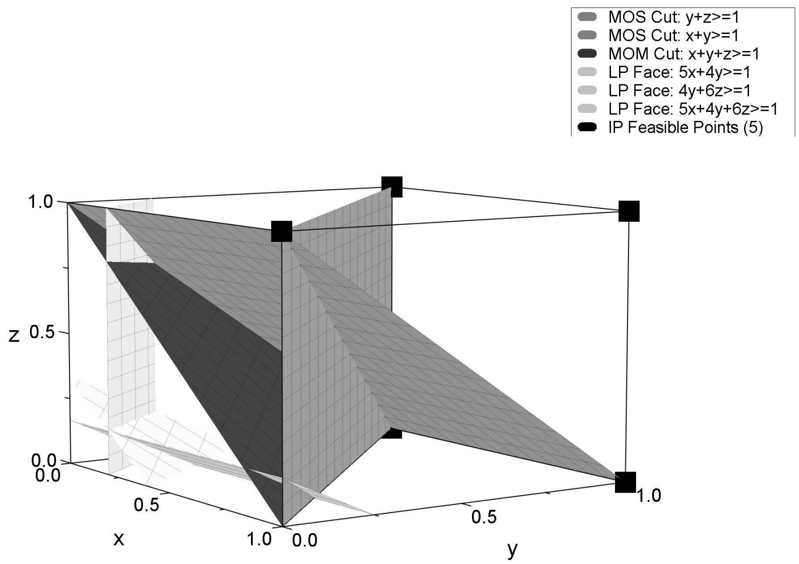

Figure 4.1: The MOS cuts (medium grey) better approximate the IP feasible region, located in the upper right corner, than the MOM cut (dark grey).

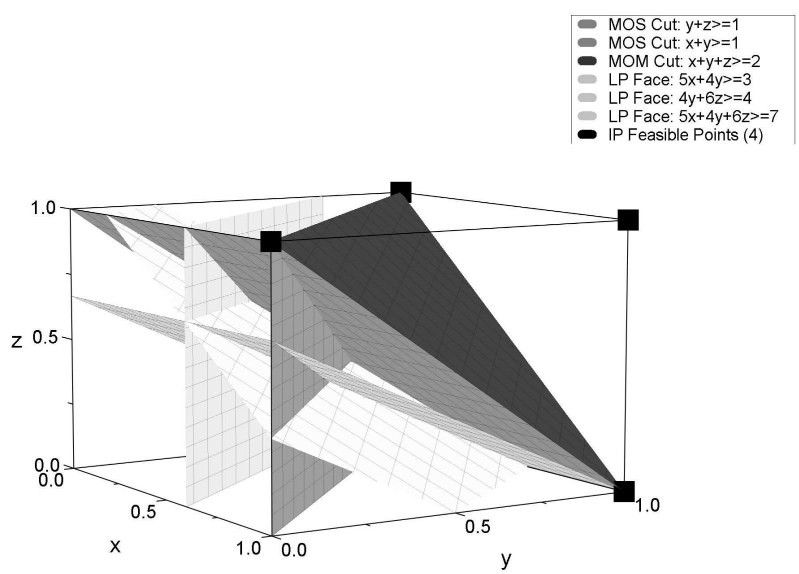

Figure 4.2: The MOM cut (dark grey) better approximates the IP feasible region, located in the upper right corner, than the MOS cuts (medium grey).

is reduced to three design decisions concerning arcs. There are three binary decisions variables x, y, and z, each of them indicates whether or not the corresponding arc has to be included into the network. Depending on the demand structure, four (Figure 4.2) or five (Figure 4.1) value assignments, respectively, form the set of (integer) feasible solutions. They are located at the cube corners and highlighted by black squares. In both figures, the feasible region of the LP relaxation is bounded by the faces of cube and three additional faces. These three faces are displayed as planes in bottom left corner shaded in light grey. The feasible region of the LP relaxation is located right above these planes. The two MOS cuts are shaded in medium grey, the MOM cut in dark grey. It is easy to see, that the MOM cut in Figure 4.2 provides a better approximation of the IP feasible region, which is located in the upper right corner, than the MOS cuts. Figure 4.1 shows the other way round.

4.2 Connected-Cut

Besides single node cuts, there is another type of valid inequality which can be stated by exploiting the structure of the traffic demand. If the structure of the traffic demand implies that each node of the final network graph is reachable from every other node, i.e. the graph is *strongly connected*, then the network consists of at least as many arcs as nodes. This comes from the fact that the most economical way (which means minimizing the number of arcs required) to achieve strong connectivity is a cyclic path visiting each node once. For n nodes such cycle, called *Hamiltonian cycle*, consists of n arcs. So if one has figured out that the final network graph will consist of only one strongly connected component, the following valid inequality can be added to both problem formulation NDRI-MIP and NDRI-MIP$^{\mathrm{WC}}$.

$$\sum_{(i,j)\in A} x_{ij}^{\mathrm{used}} \geqslant |N| \tag{4.6}$$

The question remains, how to figure out whether the final network graph will consist of only one strongly connected component without knowing the final design. The idea is to create a network graph out of the given nodes and the traffic demand. After that, strongly connected components are identified using *Tarjan's Algorithm* [Tar71] which is based on a depth-first search. Its running time complexity is linear in the number of nodes and arcs. If it turns out that there is only one strongly connected component, valid inequality (4.6) is added to the problem formulation. The input graph $G^{\text{Tar}} = (N, A^{\text{Tar}})$ for Tarjan's Algorithm is created as follows. For each traffic flow a directed arc is included into G^{Tar}:

$$A^{\text{Tar}} := \{(i,j) \in N \times N \mid (i,j,tr_\tau) \in TD\}. \tag{4.7}$$

Each traffic flow $(st_\tau, de_\tau, tr_\tau) \in TD$ implies the existence of at least one st_τ, de_τ -path in the final optimal graph G^{opt}, which is unknown. Such existence is represented by an arc (st_τ, de_τ) in G^{Tar}. Paths in G^{Tar} correspond to sequences of paths which have to exist in G^{opt}. Reachability therefore can be transferred from G^{Tar} to G^{opt}, independent of the final design. So if G^{Tar} consist of only one strongly connected component, G^{opt} consist of only one strongly connected component, too.

The question may be raised why strong connectivity is not included as a requirement into the problem formulation. In relation to practice, this would be reasonable, because infrastructure managers would like to keep a certain amount of flexibility regarding their managed networks to be able to offer (at least in theory) each possible station-to-station connection to railway undertakings. Nevertheless, the requirement that each station is reachable from every other station is not included into the problem formulation because the number of constraints would significantly increase, which in turn would limit the solvability and increase the solving times.

4.3 Flow Bounds

This section presents valid inequalities for NDRI-MIP$^{\text{WC}}$, which serve as upper bounds for $F_{t,ij}$,i.e. the flow of trains of type $t \in \{t_0, t_1\}$[1], which is routed via arc (i,j):

$$F_{t,ij} := \sum_{p \in \mathcal{P}} \delta_{ij}(p) f_t(p) \tag{4.8}$$

4.3.1 Simple Upper Bounds

Simple upper bounds independent of any binary decision variables are introduced in this section. The idea is to assign to each train its minimum capacity consumption, which is the minimum headway time of the corresponding train type during homogeneous

[1] As already introduced in Section 3.9, NDRI-MIP$^{\text{WC}}$ is restricted to two train types.

operation. So for a train of type t the minimum capacity consumption is $z_{tt,ij}$. In this way the minimum capacity consumption of flow $F_{t,ij}$ is $F_{t,ij} \cdot z_{tt,ij}$. Together with the maximum of consumable capacity max_{cc} following bound is derived and added to the problem formulation of NDRI-MIP$^{\mathrm{WC}}$ for each arc (i,j) and train type t:

$$F_{t,ij} z_{tt,ij} \leqslant max_{\mathrm{cc}} \Leftrightarrow F_{t,ij} \leqslant \frac{max_{\mathrm{cc}}}{z_{tt,ij}} = max_t. \tag{4.9}$$

Analogous to that an additional valid inequality for the total flow on arc (i,j) is added to the model:

$$\sum_{t \in T} F_{t,ij} z_{tt,ij} \leqslant max_{\mathrm{cc}}. \tag{4.10}$$

Since these bounds are calculated under the assumptions of homogeneous operation and maximum available arc capacity (cf. definition of max_{cc} in (2.16)), the bound is not very tight. But it has the advantage to provide information independent of the value assignment of problem variables x_{ij}^{used}, x_{ijs}^{used}, and w_{ij}, which in turn ensures that its expressiveness is not affected by relaxations of that variables.

4.3.2 γ Bound

Valid inequalities (4.11) and (4.12), which are included into the problem formulation for each arc $(i,j,s) \in A^{\mathrm{m}}$, are calculated depending on problem variables x_{ijs}^{used} and w_{ij}.

$$F_{t_0,ij} \leqslant \gamma_{ijs} + (1 - w_{ij})(max_{t_0,ijs} - \gamma_{ijs}) + (1 - x_{ijs}^{\mathrm{used}})(max_{t_0} - \gamma_{ijs}) \tag{4.11}$$

$$F_{t_1,ij} \leqslant \gamma_{ijs} + w_{ij}(max_{t_1,ijs} - \gamma_{ijs}) + (1 - x_{ijs}^{\mathrm{used}})(max_{t_1} - \gamma_{ijs}) \tag{4.12}$$

The basic idea behind these bounds is that the information provided by these variables leads to tighter upper bounds of the traffic flow routed via an arc (i,j,s). Table 4.1 shows which bounds are enabled by different value assignments of the corresponding decision variables.

If the current arc is included into the network (cases iii) and iv) of Table 4.1), the upper bound of $F_{t,ij}$ is at most $max_{t,ijs}$. This is a tighter bound in comparison to (4.9), since max_t is calculated under the assumption that capacity of a double track line is available, which means $max_{t,ijs} \leqslant max_t$. The bound $max_{z,ijs}$ is further refined by the value assignment of w_{ij}. If $w_{ij} = 0$ (case iii) of Table 4.1), inequality $F_{t_1,ij} \leqslant F_{t_0,ij}$

case	$x^{\mathbf{used}}_{ij s}$	w_{ij}	**rhs** (4.11)	**rhs** (4.12)
i)	0	0	$max_{t_0,ijs} + max_{t_0} - \gamma_{ij}$	max_{t_1}
ii)	0	1	max_{t_0}	$max_{t_1,ijs} + max_{t_1} - \gamma_{ijs}$
iii)	1	0	$max_{t_0,ijs}$	γ_{ijs}
iv)	1	1	γ_{ijs}	$max_{t_1,ijs}$

Table 4.1: Right-hand side values for Equations (4.11) and (4.12).

is brought into force, cf. Equation (3.56). As a result, the maximal value of $F_{t_1,ij}$ is obtained if $F_{t_1,ij} = F_{t_0,ij}$. This value is represented by γ_{ijs}, which is defined in (4.3.2).

$$\gamma_{ijs} := \frac{max_{\mathrm{cc},ijs}}{z_{t_1 t_0,ijs} + z_{t_0 t_0,ijs}} \tag{4.13}$$

Several evaluations revealed a speed-up of the solution process if the presented flow bounds were added, which supports the decision to include them into the problem formulation. A separate and comprehensive evaluation of the flow bounds is not done within this work.

5 Solving & Implementation

This chapter covers different aspects concerning solving and implementation of the NDRI models NDRI-MIP and NDRI-MIP$^{\text{WC}}$. At the beginning, column generation is introduced, a solution technique which is used to solve NDRI-MIP instances without the need to deal with their whole bunches of variables. Second section deals with the fixing of variables to values which can be determined in a pre-processing step before the actual solving process starts. This reduces the number of variables the solver has to explore. Third section is focussing on the determination of suitable big-M values for the NDRI-MIP$^{\text{WC}}$ model. Such values have to be chosen with care since they have to be sufficiently large but not arbitrary large to avoid extended LP feasible regions, which may slow down the solution process. Last section shortly presents the environment used to implement NDRI models and to solve NDRI instances. This encompasses a brief introduction of the used solver software and the computer systems the solvers are executed on.

5.1 Solving NDRI-MIP

NDRI-MIP instances come along with a high number of binary *config*-variables. A problem instance with a 40 nodes network graph, for example, consists of 182,130 binary *config*-variables. To cope with the high number of binary variables a technique known as *column generation* or *pricing*, respectively, is applied. General information concerning these techniques and its combination with branch-and-bound approaches, called *branch-and-price*, is provided by the following section. After that, details of the corresponding implementation are introduced. The last two subsections of this section deal with *farkas pricing* and its implementation. Farkas pricing is used to deal with infeasible linear programming problems which could arise during branch-and-price.

5.1.1 Column Generation & Branch-and-Price

The column generation technique is used to solve linear programs which have to cope with a large number of variables and columns, respectively[1]. The aim is to find optimal solutions without explicitly examining the whole set of variables. Starting point of the iterative process is the so-called *master problem* (MP) which is transformed to a *restricted master problem* (RMP), i. e the master problem where the set of considered variables is restricted to a *meaningful* subset of the original variables. The RMP then is solved to optimality. In a second step remaining variables are checked regarding their potential to improve the optimal solution. The problem of finding such candidate variables is called *subproblem* or *pricing problem*. Found variables are added to the restricted master problem, which, again, has to be solved to optimality. Both steps have to be repeated until there is no variable which is worth adding to the RMP. At that stage, the optimal solution of the RMP is an optimal solution of the MP.

Example 5.1. Let the master problem be a linear program defined by (5.1)–(5.3), where constraint matrix A is represented by a set of columns: $A := \{a_j \mid j \in J\}$. The restricted master problem is obtained by using just a subset of columns $A' := \{a_j \mid j \in J' \subseteq J\}$.

$$\text{Minimize} \sum_{j \in J} c_j x_j \tag{5.1}$$

subject to

$$Ax \geqslant b, \tag{5.2}$$

$$x_j \geqslant 0 \qquad \forall j \in J, \tag{5.3}$$

Let π be the optimal dual solution of the RMP. The pricing problem (5.4) aims to find a column which improves the optimal solution of the RMP, i.e. the column a_j with least negative *reduced cost*[2]: $c_j - \pi^t a_j$. This leads to the following optimization problem which is the subproblem or pricing problem, respectively.

$$z^* := \min\{c_j - \pi^t a_j \mid j \in J, a_j \in A\} \tag{5.4}$$

[1]In linear programs variables usually are represented by corresponding columns of the constraint matrix.
[2]Further information about dual variables, reduced cost, and *Duality Theory* in general can be obtained from Bazaraa et al. [BJS10].

If $z^* < 0$, the corresponding column is added to the RMP, which then has to be solved to optimality, again. Otherwise, the current optimal primal solution of the RMP optimally solves the MP, too.

In case of (mixed) integer linear programs, the column generation approach has to be combined with branch-and-bound techniques (cf. Section 3.1.1) where column generation is applied to the problem relaxations arising at each B&B-node. Such combined approach is known as *branch-and-price* (B&P). Further information about column generation, especially for the case of integer programming, can be obtained from a compendium by Desaulniers et al. [DDS05]. It includes a nice didactic introduction to the use of the column generation technique written by Desrosiers and Lübbecke [DL05]. The description of the column generation technique given above is based on that introduction.

5.1.2 Implementation of B&P

In order to solve NDRI-MIP instances using a branch-and-price approach, SCIP[3] is used as solver. Within SCIP a so-called *variable pricer* has to be implemented. For this purpose, SCIP provides a callback mechanism. The callback method PRICERREDCOST is of central importance. It is called whenever the LP relaxation of the current B&B–node is feasible. Inside this method one can implement how to solve the pricing problem to find variables which improve the objective value. Before the implementation of the pricing problem is presented in detail, first it is explained how the RMP is created.

Let $G^{\mathrm{c}} = (N, A^{\mathrm{m}})$ the complete network graph of an instance of NDRI-MIP and $\mathcal{C}$ the set of all *config*-variables of that instance. The RMP is created in the following way. For each multi-arc $(i, j, s) \in A^{\mathrm{m}}$ the set of corresponding configurations $\mathcal{C}_{ijs}$ and *config*-variables, respectively, is restricted to a set $\mathcal{C}_{ijs}^{\mathrm{start}}$ of so-called *start*-configurations. $\mathcal{C}_{ijs}^{\mathrm{start}}$ consists of one *start*-configuration for each train type. A *start*-configuration of train type t is referred to as that configuration with the maximal value in the t-th component. So *start*-configurations of a multi-arc represent maximal train counts of each train type during homogenous operation.

$$\mathcal{C}_{ijs}^{\mathrm{start}} := \bigcup_{t \in T} \{c_{ijs} \in \mathcal{C}_{ijs} \mid c_{t,ijs} \text{ is maximal}\}. \tag{5.5}$$

[3]SCIP is part of the open source optimization suite which is used here to solve NDRI-MIP problem instances using branch-and-price. An introduction is given in Section 5.4.

For $|T| = 2$, which is chosen for the evaluation of the NDRI models, the cardinality of $\mathcal{C}_{ijs}^{\text{start}}$ is 2, too. That is easy to see when looking at the construction rule of $\mathcal{C}_{ijs}$ which ensures the uniqueness each component of every configuration.

Let $\mathcal{C}^{\text{start}}$ be the set of all start-configurations:

$$\mathcal{C}^{\text{start}} := \bigcup_{(i,j,s)\in A^{\text{m}}} \mathcal{C}_{ijs}^{\text{start}}. \tag{5.6}$$

The pool of variables $\mathcal{C}'$, which is available during pricing, is then defined as:

$$\mathcal{C}' := \mathcal{C} \setminus \{y_{c_{ijs}}^{\text{used}} \mid c_{ijs} \in \mathcal{C}^{\text{start}}\}. \tag{5.7}$$

Following Equation (5.4), the pricing problem for NDRI-MIP LP relaxations is outlined in (5.8). Since *config*-variables are not directly linked to costs, cf. Equation (3.45), the cost coefficient is represented by the value zero. Each variable $y_{c_{ijs}}^{\text{used}}$ has only $|T| + 1$ nonzero values in the coefficient matrix. One for each capacity constraint (3.46) of the current arc. The remaining one for the „only one configuration per multi-arc“ constraint (3.48). For the derivation of the reduced cost, a column composed in the described way has to be multiplied with the dual multipliers $\pi_1, \ldots, \pi_{|T|}$ and π_0 which are associated to the constraints mentioned before and which can be retrieved by the function SCIPgetDualsolLinear. The negative algebraic sign of the coefficients $c_{k,ijs}$ has its roots in the logical representation of the model used by SCIP and the LP file format, respectively. Thereby the left-hand side of constraints contains every term associated to problem variables, whereas the right-hand side consists of a scalar.

$$z^* := \min\{0 - \begin{pmatrix}\pi_1 & \pi_2 & \ldots & \pi_{|T|} & \pi_0\end{pmatrix} \cdot \begin{pmatrix} -c_{1,ijs} \\ -c_{2,ijs} \\ \vdots \\ -c_{|T|,ijs} \\ 1 \end{pmatrix} \mid \forall c_{ijs} \colon y_{c_{ijs}}^{\text{used}} \in \mathcal{C}'\} \tag{5.8}$$

If $z^* < 0$ the corresponding variable $y_{c_{ijs}}^{\text{used}}$ is included into the RMP and removed from the pool of variables $\mathcal{C}'$. The solving loop is continued until PRICERREDCOST is called again. Since $\mathcal{C}'$ is completely known in advance and its cardinality is a manageable number, the subproblem is solved by the explicit performance of $|\mathcal{C}'|$ reduced cost calculations.

For a 20-node example the number of *config*-variables reduces from 35,612 for the MP to 2,232 for the RMP. The optimal solution of the corresponding MIP requires another 5,188 variables which are added by (farkas) pricing. More evaluation data is presented in Section 6.3.

5.1.3 Infeasibility Handling

During branch-and-price the situation may arise that the current RMP is infeasible. This does not necessarily mean that the MP is infeasible too, since RMP just uses a subset of the variables. If the RMP is infeasible, one needs a mechanism to render the problem feasible again. The method used here is called *Farkas Pricing*, cf. Lübbecke [Lüb10], and is introduced within the SCIP framework [Ach09], which provides a callback and methods to implement it.

The idea of Farkas Pricing is based on Farkas' Lemma, see Bazaraa et al. [BJS10]. The lemma can be used to proof a fundamental theorem within the *Theory of Duality*. Bazaraa et al. [BJS10] denote the theorem *Fundamental Theorem of Duality*:

Theorem 5.1 (Fundamental Theorem of Duality)**.** For a primal and corresponding dual linear programming problem, exactly one of the following statements is true:

i) Both problems have an optimal solution.

ii) One problem has an unbounded optimal objective value, in which case the other problem must be infeasible.

iii) Both problems are infeasible.

So if the RMP of current node in the branch-and-price tree is infeasible, the dual problem is unbounded (if it is not infeasible). In such case, there exists a *dual ray* π^* which is a vector that indicates the direction of the unboundedness.

If a constraint which removes the unboundedness of the dual RMP in direction of the dual ray can be found, the feasibility of the primal RMP is restored. Potential coefficients of such a constraint correspond to the columns of the (primal) MP. Their inner products with the dual ray have to be positive to fit for the described purpose.

Using the notation introduced in the previous section this leads to an optimization problem which is similar to the pricing problem stated in Equation (5.4).

$$\max\{\pi^* a_j \mid j \in J, a_j \in A\} \tag{5.9}$$

5.1.4 Implementation of Farkas Pricing

For the implementation of farkas pricing SCIP provides the callback PRICERFARKAS which is executed if the current LP turns out to be infeasible. The values of the dual ray, if it exists, are accessible by the method SCIPgetDualfarkasLinear. Since (5.9) is equivalent to

$$\min\{-\pi^* a_j \mid j \in J, a_j \in A\} \tag{5.10}$$

farkas pricing is analogous to the reduced cost pricing described before. Using a generic pricing approach one can use the same pricing routine with the same problem (5.8). The difference is in the values assigned to $\pi_1, \ldots, \pi_{|T|}$ and π_0, which are in the case of reduced cost pricing values of the dual solution and in the case of farkas pricing values of the dual ray.

5.2 Fixing of Variables

Depending on the given traffic demand and the set of nodes it may be possible to simplify the NDRI model by fixing some problem variables to certain values before the actual solving process starts. In this way, the degrees of freedom of the model would be reduced, which implies a simplification for the solver.

5.2.1 Variables x_{ij}^{used}

If an arc (i, j) is element of each path of a traffic flow τ, i.e.

$$\forall p \in \mathcal{P}^\tau : \delta_{ij}(p) = 1, \tag{5.11}$$

then the trains inevitably have to be routed via this arc. This in turn means that this arc has to be included into the network, so the corresponding *used*-variable x_{ij}^{used} takes the value one. Design decisions regarding the stage of extension still have to be made, but constraints (3.47) and (3.58) ensure that exactly one of the corresponding variables x^{used}_{ijs} takes the value one in each feasible solution.

If an additional condition is satisfied, decision variables w_{ij} of the NDRI-MIP$^{\text{WC}}$ model can be fixed, too.

5.2.2 Variables w_{ij}

Variables w_{ij} indirectly determine the mix ratio of train types using an arc, cf. Section 3.9. So, if that mix ratio is determinable independent of the final routing, such variables can be fixed to corresponding values. Assuming that x^{used}_{ij} is fixed for an arc (i, j). Then variable w_{ij} can be fixed, too if each traffic flow $\tau \in TD$ satisfies one the following formulas:

$$\text{i) } \forall p \in \mathcal{P}^{\tau} : \delta_{ij}(p) \neq 1 \qquad \text{or} \qquad \text{ii) } \forall p \in \mathcal{P}^{\tau} : \delta_{ij}(p) = 1 \tag{5.12}$$

This ensures that each traffic flow either always or never uses arc (i, j). This in turn ensures that the mix ratio of train types on that arc is constant and independent of the routing. So train counts $n_{t_0,\tau}, n_{t_1,\tau}$ of traffic flows τ inevitably using arc (i, j), i.e. fulfilling condition ii), can be summed up and used to determine the value of w_{ij}. Let $\sigma_{t_0,ij}$ and $\sigma_{t_1,ij}$ be the corresponding sums with $\sigma_{t,ij} := \sum\limits_{\tau \text{ satisfying ii)}} n_{t,\tau}$, then the following holds:

$$w_{ij} = \begin{cases} 1 & \text{if } \sigma_{t_0,ij} \leqslant \sigma_{t_1,ij}, \\ 0 & \text{else.} \end{cases} \tag{5.13}$$

5.3 Derivation of Big-M values

As described in Section 3.9, a big-M approach is used to model the decision process induced by decision variables x^{used}_{ijs} and w_{ij}. Depending on the values which are assigned to those variables, such big-M constraints are either trivially satisfied in order to preserve feasibility or they provide the desired restriction of the solution space. It is important to choose big-M values which are, on the one hand, sufficiently large to satisfy the described property and, on the other, as small as possible to preserve tight LP relaxations, see [CRT90].

At first, suitable values for M_{t_0} and M_{t_1} which are used in inequalities (3.56) and (3.57) of the NDRI-MIP$^{\mathrm{WC}}$ model are presented. After that, the derivation of values for M, M', M'', and M''' used in the capacity constraints (3.54) and (3.55) of the same model are shifted into focus.

5.3.1 Mix Ratio Constraints

The constraints (3.56) and (3.57) quantities M_{t_0} and M_{t_1} are used in are:

$$\begin{aligned} F_{t_1,ij} &\leqslant F_{t_0,ij} + M_{t_1} w_{ij} && \forall (i,j) \in A, \\ F_{t_0,ij} &\leqslant F_{t_1,ij} + M_{t_0}(1 - w_{ij}) && \forall (i,j) \in A. \end{aligned}$$

The big-M approach is used to model the ratio of proportion of train types using an arc (i,j). If $w_{ij} = 0$, inequality (3.56) is enabled and flow $F_{t_1,ij}$ is bounded from above by $F_{t_0,ij}$. Flow $F_{t_0,ij}$ in turn is bounded by $F_{t_1,ij} + M_{t_0}$. Since no information about the multi-arc selection is available at this point, $F_{t_0,ij}$ has to be bound from above by max_{t_0} the maximum number of trains of type t_0 which are allowed to operate on the corresponding line. Since, in present case, $F_{t_1,ij}$ may take the value zero, max_{t_0} is assigned to M_{t_0}:

$$M_{t_0} := max_{t_0}. \tag{5.14}$$

Following this reasoning in the case of $w_{ij} = 1$, too, it is easy to see that max_{t_1} is assigned to M_{t_1}:

$$M_{t_1} := max_{t_1}. \tag{5.15}$$

5.3.2 Capacity Constraints

In this section the determination of suitable values M, M', M'', and M''' used in the capacity constraints (3.54) and (3.55) of the NDRI-MIP$^{\mathrm{WC}}$ model is considered. Those capacity constraints, listed again by following equations, provide upper bounds (right-hand side) on the capacity consumption of trains using an arc (left-hand side):

$$\begin{aligned} \alpha^0_{ijs} F_{t_0,ij} + \alpha^1_{ijs} F_{t_1,ij} &\leqslant max_{\mathrm{cc},ijs} x^{\mathrm{used}}_{ijs} + M w_{ij} + M' \sum_{r \neq s} x^{\mathrm{used}}_{ijr} && \forall (i,j,s) \in A^{\mathrm{m}}, \\ \alpha^2_{ijs} F_{t_0,ij} + \alpha^3_{ijs} F_{t_1,ij} &\leqslant max_{\mathrm{cc},ijs} x^{\mathrm{used}}_{ijs} + M''(1 - w_{ij}) + M''' \sum_{r \neq s} x^{\mathrm{used}}_{ijr} && \forall (i,j,s) \in A^{\mathrm{m}}. \end{aligned}$$

case	$\boldsymbol{x_{ijs}^{\mathrm{used}}}$	$\sum_{r \neq s} \boldsymbol{x_{ijr}^{\mathrm{used}}}$	$\boldsymbol{w_{ij}}$	**rhs** (3.54)	**rhs** (3.55)
i)	0	0	0	0	M''
ii)	0	0	1	M	0
iii)	0	1	0	M'	$M'' + M'''$
iv)	0	1	1	$M + M'$	M'''
v)	1	0	0	$max_{cc,ijs}$	$max_{cc,ijs} + M''$
vi)	1	0	1	$max_{cc,ijs} + M$	$max_{cc,ijs}$

Table 5.1: Right-hand side values for capacity constraints.

For an arc $(i, j) \in A$ there is a pair of constraints for each multi-arc $(i, j, s) \in A^{\mathrm{m}}$ in the model. Exactly one constraint or none at all is enabled by decision variables x_{ijs}^{used} and w_{ij} and corresponding big-M values M & M' and M'' & M''', respectively. One pair of constraints is enabled if the corresponding multi-arc is included into the network ($x_{ijs}^{\mathrm{used}} = 1$). One constraint of that pair of constraints is further brought into force by the decision induced by the mix ratio of train types which are routed via (i, j) and (i, j, s), respectively ($w_{ij} = 0 \Leftrightarrow F_{t_1,ij} \leqslant F_{t_0,ij}$). So for the determination of suitable values M, M', M'', and M''' six different cases of value assignments of decision variables have to be distinguished. Table 5.1 shows the values of the right-hand sides (rhs) of that constraints which result from six possible value assignments.

i),ii) If no multi-arc (i, j, s) of the current arc (i, j) is used at all, the capacity consumption is bound to zero. In that regard, it is not necessary to force both right-hand sides to be zero. Since coefficients α_{ijs}^{k} are positive and flows $F_{t_0,ij}$ and $F_{t_1,ij}$ are the same for both constraints, either both left-hand sides take a positive value or both are zero.

v),vi) If multi-arc (i, j, s) is included into the network and $w_{ij} = 0$, capacity constraint (3.54) is enabled and provides $max_{\mathrm{cc},ijs}$ as upper bound on the capacity consumption. Capacity constraint (3.55) provides only the trivial upper bound $max_{cc,ijs} + M''$ (case v)). If $w_{ij} = 1$, while $x_{ijs}^{\mathrm{used}} = 1$ still holds, the corresponding right-hand side values are shown in case vi). Since (i, j, s) is included into the network remaining used-variables of multi-arcs of arc (i, j) take the value zero.

iii),iv) If multi-arc (i, j, s) is not included into the network but some multi-arc (i, j, r)

with $r \neq s$, trivial upper bounds which only depend on big-M values are provided.

To ensure that big-M terms of cases iii)–vi) serve as trivial upper bounds, values for M, M', M'', and M''' have to be chosen sufficiently large. The decisive factors, in this regard, are the maximal values corresponding left-hand sides can take. So, for the determination of suitable big-M values following system of inequalities is derived from cases iii)–vi)[4]:

$$max_{\text{lhs (3.54),iii)}} \leqslant M' \qquad max_{\text{lhs (3.55),iii)}} \leqslant M'' + M''' \tag{5.16}$$

$$max_{\text{lhs (3.54),iv)}} \leqslant M + M' \qquad max_{\text{lhs (3.55),iv)}} \leqslant M''' \tag{5.17}$$

$$max_{\text{lhs (3.54),vi)}} \leqslant max_{\text{cc},ijs} + M \qquad max_{\text{lhs (3.55),v)}} \leqslant max_{\text{cc},ijs} + M'' \tag{5.18}$$

Values $max_{\text{lhs (3.54),k)}}$ and $max_{\text{lhs (3.55),k)}}$ denote the maximal values of left-hand side of the capacity constraints for case k). They are determined using linear optimization problems, shown in Table 5.2. Intuitively, the left-hand side serves as objective function. The second constraint is directly derived from the value assignment of w_{ij}. The idea behind the first constraint is to choose exactly that constraint which is enabled in the current set of multi-arc constraints. This is reasonable since all value assignments of $F_{t_0,ij}$ and $F_{t_1,ij}$ have to fulfill this constraint. For cases v) and vi) this constraint is known, since $x_{ijs}^{\text{used}} = 1$. For the cases iii) and iv) the enabled constraint is not known. Due to the fact that maximal left-hand sides have to be determined, it is assumed that a constraint with minimal coefficients α_{min}^k and maximal right-hand side max_{cc} is enabled. This represents the activation of a double track line with the highest density of overtaking stations available. This assumption guarantees that all constraints are satisfiable even for the largest possible value assignments of $F_{t_0,ij}$ and $F_{t_1,ij}$. *Irrelevant* cases in Table 5.2 correspond to cases with right-hand sides without big-M values, cf. Table 5.1.

5.4 Implementation Environment

This section is a short introduction into the environment used to implement NDRI models and to solve NDRI instances. This encompasses a brief presentation of the used solver software and the computer systems the solvers are executed on.

[4] Cases i) and ii) are dropped, since the corresponding left-hand sides simply have to be zero.

case	$\mathbf{max_{lhs}}$ (3.54)	$\mathbf{max_{lhs}}$ (3.55)
iii)	Maximize $\alpha^0_{ijs}F_{t_0,ij} + \alpha^1_{ijs}F_{t_1,ij}$ s. t. $\alpha^0_{\min}F_{t_0,ij} + \alpha^1_{\min}F_{t_1,ij} \leqslant max_{cc}$ $F_{t_1,ij} \leqslant F_{t_0,ij}$	Maximize $\alpha^2_{ijs}F_{t_0,ij} + \alpha^3_{ijs}F_{t_1,ij}$ s. t. $\alpha^0_{\min}F_{t_0,ij} + \alpha^1_{\min}F_{t_1,ij} \leqslant max_{cc}$ $F_{t_1,ij} \leqslant F_{t_0,ij}$
iv)	Maximize $\alpha^0_{ijs}F_{t_0,ij} + \alpha^1_{ijs}F_{t_1,ij}$ s. t. $\alpha^2_{\min}F_{t_0,ij} + \alpha^3_{\min}F_{t_1,ij} \leqslant max_{cc}$ $F_{t_0,ij} \leqslant F_{t_1,ij}$	Maximize $\alpha^2_{ijs}F_{t_0,ij} + \alpha^3_{ijs}F_{t_1,ij}$ s. t. $\alpha^2_{\min}F_{t_0,ij} + \alpha^3_{\min}F_{t_1,ij} \leqslant max_{cc}$ $F_{t_0,ij} \leqslant F_{t_1,ij}$
v)	irrelevant	Maximize $\alpha^2_{ijs}F_{t_0,ij} + \alpha^3_{ijs}F_{t_1,ij}$ s. t. $\alpha^0_{ijs}F_{t_0,ij} + \alpha^1_{ijs}F_{t_1,ij} \leqslant max_{cc,ijs}$ $F_{t_1,ij} \leqslant F_{t_0,ij}$
vi)	Maximize $\alpha^0_{ijs}F_{t_0,ij} + \alpha^1_{ijs}F_{t_1,ij}$ s. t. $\alpha^2_{ijs}F_{t_0,ij} + \alpha^3_{ijs}F_{t_1,ij} \leqslant max_{cc,ijs}$ $F_{t_0,ij} \leqslant F_{t_1,ij}$	irrelevant

Table 5.2: Determination maximal left-hand side values.

5.4.1 Solvers

Two different optimization suites are used to set up the models and to solve corresponding problem instances. Both suites are described in the following sections.

Gurobi

The Gurobi Optimizer is a linear programming and mixed integer programming solver which exploits modern multi-core processors. Gurobi is currently performance benchmark winner, so it provides the fastest solving times. Although Gurobi offers free academic licenses, the access is limited by an interface. It allows only the usage of restricted sets of functions, parameters and attributes, which can be accessed via library interfaces for C, C++, Java, .NET, MATLAB, R or Python. Despite the restricted interface, it is a powerful solver which additionally supports some modeling systems like MPL and AMPL and is able to read and write LP and MPS files. For further information see Gurobi homepage [Gur12]. Here, Gurobi version 4.5 is used to solve NDRI-MIP$^{\mathrm{WC}}$ instances and NDRI-MIP instances without using column generation, i.e. with the whole set of variables. The implementation of the models is done using Java.

SCIP

SCIP stands for Solving Constraint Integer Programs and was developed at the Zuse Institute for Information Technology in Berlin (ZIB). It is part of the open source ZIB optimization suite. Besides SCIP, it encompasses the modeling language ZIMPL and an LP solver SoPlex. Since SCIP provides the opportunity to embed different LP solvers and Gurobi performs much better than SoPlex[5], Gurobi 4.5 instead of SoPlex 1.6 is used within this work. Since the complete source code of SCIP is available, it allows total control of the solution process and unrestricted access to any information at any stage of the solution process. The user can define write and include own pricers, branching rules, presolvers, heuristics and so on. This is the reason why SCIP is used to implement branch-and-price and to solve NDRI-MIP. The implementation of the model is done using programming languages C and C++. Here SCIP version 2.0.1 is used. Further information about the concept of constraint integer programming and SCIP are provided by Achterberg [Ach09] and the SCIP homepage [SCI12].

5.4.2 Computer Systems

Two different computer systems are used to evaluate NDRI instances. On the one hand, there is a *desktop computer* with Windows 7 64-bit operating system, a dual core processor which provides four parallel threads, and four gigabyte of memory. On the other hand, single nodes of the cluster computer *Bull MPI-L* are used. This cluster computer, which consists of 252 nodes, is part of the *High Performance Computing* service offered by the *Center of Computing and Communication of RWTH Aachen University*. Each node uses Scientific Linux 64-bit as operating system and is equipped with 12 cores providing 24 parallel threads. Up to 96 gigabyte of memory are accessible. Table 5.3 summarizes the performance indicators of both systems. The desktop system provides inferior computational power in comparison to the cluster node. This is not disadvantageous when using SCIP, since SCIP does not support multi-core processors anyway. The advantage of the desktop system lies in its accessibility which does not depend on the load factor of a waiting queue, which is used to control the access to the cluster.

[5]Current SCIP MIP solver benchmarks for different LP solver embeddings can be obtained from the SCIP homepage [SCI12].

indicator	**desktop computer**	**cluster node**
operating system	Windows 7 64-bit	Scientific Linux 64-bit
cpu cores/threads	2/4	12/24
cpu clock rate	2.40 GHz	3.06 GHz
memory	4 GB	96 GB

Table 5.3: Performance indicators of computer systems used to evaluate NDRI models.

6 Evaluation

This chapter deals with the evaluation of NDRI models NDRI-MIP and NDRI-MIP$^{\text{WC}}$. To evaluate NDRI-MIP two approaches are differentiated. NDRI-MIP$^{\text{CG}}$ denotes NDRI-MIP instances which are solved using column generation. Such instances start with a restricted set of configurations and corresponding config-variables. Additional variables are added during the solution process, cf. Section 5.1.1. NDRI-MIP$^{\text{noCG}}$ denotes NDRI-MIP instances which are solved without using column generation. So, the whole set of config-variables is included right from the beginning. Objective values of NDRI-MIP instances stay the same independent of the particular solution approach.

At first the evaluation framework is introduced. This encompasses the setup of a network as well as the derivation of reasonable traffic flows. Furthermore, the occupation time available to trains operationally using a line and the set of paths which is available to traffic flows are specified. Second section deals with the size of NDRI problem instances in terms of number of variables and constraints. Finally, computational results are presented and discussed.

6.1 Evaluation Framework

To evaluate the optimization models for NDRI a test scenario/framework has to be created. This framework primarily serves as test bench to analyze quantities like solving time, scalability, and speed-up by column generation, and to figure out which size of a problem instance pushes solvers to their limits. Due to the lack of real data, it is an artificial scenario which does not raise the claim to precisely model network design scenarios with relevance to practice. Following input is required to setup the evaluation framework:

i) a set of nodes N and

ii) a traffic demand TD as a set of traffic flows between elements of N.

Aachen	Aschaffenburg	Augsburg	Baden-Baden	Berlin
Bielefeld	Bonn	Braunschweig	Bremen	Chemnitz
Dresden	Erfurt	Essen	Frankfurt (M)	Frankfurt (O)
Freiburg	Fulda	Göttingen	Hagen	Halle (Saale)
Hamburg	Hamm	Hannover	Heidelberg	Karlsruhe
Kassel	Kiel	Koblenz	Köln	Leipzig
Lübeck	Magdeburg	Mainz	Mannheim	München
Münster	Nürnberg	Oberhausen	Offenburg	Osnabrück
Regensburg	Riesa	Rosenheim	Rostock	Saarbrücken
Solingen	Stuttgart	Trier	Ulm	Weimar
Wiesbaden	Wolfsburg	Wuppertal	Würzburg	

Table 6.1: Set of input nodes.

This is subject of next four subsections. Last two subsections deal with the available occupation time and the set of paths which is available to each traffic flow.

6.1.1 Network

The set of 54 nodes shown in Table 6.1 is chosen out of 63 *large nodes* of the current network of railway infrastructure in Germany. Such large nodes are, for example, characterized by high volumes of traffic or their operational significance in the network. This could be, for example, scheduled stops of trains with high priority. Such nodes were identified within unpublished work done at the Institute of Transport Science RWTH Aachen University [NW03]. The set of nodes is reduced to 54 nodes by dropping those nodes which induce degenerate behavior of the model used to generate traffic flows. This is case if corresponding cities are located close to each other and have many inhabitants like Essen, Duisburg, and Dortmund. Sizes of traffic flows then are overestimated, cf. Section 6.1.3, so, for example, Duisburg and Dortmund are dropped. Another reason to minimize the number of nodes which are located close to each other lies in the fact that long-distance and freight traffic usually operate on long distances.

Using the set of nodes a complete multi-arc network graph $G^{\mathrm{c}} = (N, A^{\mathrm{m}})$ is created, cf. Definition 3.1.

Remark. For the evaluation of smaller problem instances only a subset of N is taken into consideration.

Train type	v_t	d_t	$l_{\mathrm{tr},t}$
IC (Intercity, long distance)	200	0.5	400
RE (Regio Train, short distance)	140	0.5	400
IRC (Cargo Train, freight)	80	0.3	750

Table 6.2: Train type parameters.

6.1.2 Traffic Demand

For the current purpose three types of traffic are distinguished: long distance traffic, short distance traffic, and freight traffic.

The traffic types are represented by train types which are characterized by three parameters introduced in Definition 2.6. The corresponding parameters of each train type considered are listed in Table 6.2.

Traffic demand TD consists of traffic flows $\tau = (st_\tau, de_\tau, tr_\tau)$ for each pair of nodes $s, d \in N$, cf. Section 2.3.2. This results in $|N|(|N| - 1)$ traffic flows. Each traffic flow τ between nodes $s, d \in N$ is just composed of trains of type IC and IRC.

$$\tau = (st_\tau, de_\tau, \{n_{\mathrm{ic}}, n_{\mathrm{irc}}\}) \tag{6.1}$$

The restriction to two train types is motivated by the fact that the NDRI-MIP$^{\mathrm{WC}}$ model is just defined for two different train types. The NDRI-MIP model is able to deal with arbitrary number of train types but in terms of complexity the restriction is, nevertheless, reasonable. Missing short distance traffic is included into the evaluation framework as a *basic traffic load*. This reduces available line capacity, cf. Section 6.1.4.

The reason to choose train types IC/IRC instead of IC/RE or RE/IRC, respectively, lies in the fact that especially large differences between maximum speeds have significant influence on the determination of capacity consumption of a mix of trains. Furthermore, large nodes, which are considered here, are representing those stations which in general serve as start and end station for long distance and freight traffic.

The next section describes how to determine tangible train numbers n_{ic} and n_{irc} for each traffic flow. This section is followed by a treatise on the basic traffic load on arcs $(i, j) \in A$ which is induced by short distance traffic.

Business unit	Volume sold	Load factor	Train type
DB Bahn Long-Distance	$34{,}708 \cdot 10^6$ pkm	45.1 %	IC
DB Schenker Rail	$93{,}948 \cdot 10^6$ tkm	484.0 t	IRC

Table 6.3: Annual Report DB Mobility Logistics 2009[2].

6.1.3 Estimated Traffic Flows

Usually, long-term infrastructure planning processes make use of traffic forecasts to estimate future traffic flows. Such traffic forecasts utilize socio-economic measures like the number of households and their available income or transportation costs to determine origin/destination traffic matrices which geographically link generation of traffic to its destinations. Unfortunately, data like those traffic matrices is not available for this thesis. Similarly, there is no data available to aggregate traffic flows from current timetables or operating programs. So, fairly reasonable estimations of traffic flows or train numbers n_{ic} and n_{irc}, respectively, are made by the use of total traffic volumes of *DB Mobility Logistics AG*[1] which is responsible for the majority of railway traffic load in Germany. Using a *gravity model*, introduced later on, a share of that total traffic volume is assigned to each traffic flow. This share then is *transformed* into train counts and in turn into traffic flows τ.

Since traffic flows consist of train types IC and IRC, see before, total traffic volumes used here are *sold traffic services* of corresponding business units of DB Mobility Logistics AG: long distance traffic (*DB Bahn Long-Distance*) and freight traffic (*DB Schenker Rail*). Table 6.3 shows the traffic volumes derived from the Annual Report 2009 [AG09] and links it to the corresponding train type t. Additionally, load factors of trains of the corresponding business units are listed. They are needed later on, when shares of traffic volumes are transformed into train counts of traffic flows. Traffic volume and load factor assigned to train type t are in the following denoted tv_t and lf_t, respectively.

[1] DB Mobility Logistics AG is the biggest railway undertaking in Germany.

[2] Transport volumes are specified in passenger-kilometers (pkm) per annum and tonne-kilometers (tkm) per annum, respectively. The load factor is expressed as average share of occupied seats per train and average weight of goods transported in tons per train, respectively.

Gravity Model

Gravity models which are used to determine traffic volumes between two locations have their roots in *The law of travel and its application to rail traffic* published by Lill [Lil91] and translated by Hoppner [Hop69]. Lill's law of travel links a traffic volume v_{ij} to the number of inhabitants and the distance between the two considered cities i and j:

$$v_{ij} = k \cdot \frac{P_i \cdot P_j}{D_{ij}^x}, \tag{6.2}$$

where P_i is the population of city i, D_{ij} is the distance between i and j. Parameter k and x may vary from case to case. For $k = 1$ and $x = 2$ Equation (6.2) is similar to Newton's law of gravitation. This is why such models are denoted as gravity models.
The gravity model considered here is used to determine a weighting for traffic flows. The weighting then is used to distribute the total traffic volume by type among the traffic flows. Let $\tau = (st_\tau, de_\tau, tr_\tau)$ be a traffic flow with uninstantiated set tr and train type t. Then traffic volume $v_{st,de,t}$ of train type t between stations st and de is defined as follows:

$$v_{st,de,t} = \frac{P_{st} \cdot P_{de}}{(\frac{l_{1,stde}}{v_t} + 1)^2}, \tag{6.3}$$

The spatial distance of the stations is replaced by the journey time in hours. The offset of one hour is added to model the journey time of the *final mile*, i.e. that amount of time passengers or goods, respectively, need to get to the station and need to reach their final destination. Furthermore, the offset prevents traffic flows with relatively small distances between start and destination station from being overweighted. Despite the offset, cities which are close to each other and, additionally, have many inhabitants were dropped from the list of large nodes, as mentioned in Section 6.1.1. Equation (6.3) clearly shows that pairs of those cities lead to very large (overestimated) traffic volumes.

The share $sv_{\tau,t}$ of the total traffic volume tv_t which is assigned to a traffic flow $\tau = (st_\tau, de_\tau, tr_\tau)$ and the corresponding train type t is calculated as follows:

$$sv_{\tau,t} = \frac{v_{st,de,t}}{\sum\limits_{st \in N} \sum\limits_{de \in N} v_{st,de,t}} \cdot tv_t. \tag{6.4}$$

Gravity models tend to be too optimistic, see Leibbrand [Lei84]. There are two reasons why this is not of consequence for current test purpose. On the one hand, the gravity model is not used to determine the traffic volume, since it is known in advance, see Table 6.3. The model is just used as weighting tool. On the other hand, the traffic demand calculated here is only used for model test purposes and does not raise the claim to model real network design scenarios precisely.

Transformation

Let $\tau = (st_\tau, de_\tau, tr_\tau)$ be a traffic flow and $sv_{\tau,t}$ the share of the total traffic volume of type t which is assigned to τ.

i) Let $s_{\text{ic}} = 440$ be the average number of seats provided by a long distance train and $lf_{\text{ic}} = 45.1\%$ the corresponding load factor of such trains, see Table 6.3. The number of trains of type IC which have to be routed from s to d per day is then defined as:

$$n_{\text{ic}} := \frac{\frac{sv_{\tau,\text{ic}}/365}{l_{1,sd}}}{s_{\text{IC}} \cdot lf_{\text{ic}}/100} = \frac{\frac{sv_{\tau,\text{ic}}/365}{l_{1,sd}}}{440 \cdot 0.451} \tag{6.5}$$

ii) For load factor $lf_{\text{irc}} = 484$ tons, see Table 6.3, the number of trains of type IRC which have to be routed from s to d per day is then defined as:

$$n_{\text{irc}} := \frac{\frac{sv_{\tau,\text{irc}}/365}{l_{1,sd}}}{lf_{\text{irc}}} = \frac{\frac{sv_{\tau,\text{irc}}/365}{l_{1,sd}}}{484} \tag{6.6}$$

6.1.4 Basic Traffic Load

The basic traffic load on lines of the network is expressed as share of the total available line capacity which is consumed by short distance traffic. The total line capacity, which is reduced by this share, constitutes that amount of line capacity which is available for the traffic flows which consist of long distance traffic and freight traffic. To determine that share of line capacity which is used by short distance traffic, two representative (but confidential) lines in Germany with mixed traffic, i.e. a mix of long distance passenger traffic, short distance passenger traffic, and freight traffic, are examined. It is assumed, for the sake of simplicity, that the share of consumed capacity is equal to the share in the number of trains which operate on those lines. Averaged over both lines, the following mix ratio is found, see Table 6.4.

Next subsection introduces the available occupation time of lines which is used for

Type of traffic	Share
Long distance	24.5 %
Short distance	33.6 %
Freight	41.9 %

Table 6.4: Assumed average mix ratio by traffic types on railway lines with mixed traffic.

the evaluation of NDRI instances. For the derivation of that time the basic traffic load, induced by short distance traffic, is taken into account, too.

6.1.5 Available Occupation Time

The available occupation time of a line, i.e. the maximum consumable arc capacity is defined by Equation (2.15):

$$max_{\mathrm{cc},ij} := 0.6 \cdot t_{\mathrm{op}} \cdot n_{\mathrm{tk},ij}.$$

It is defined as 60 % of the total time available and is doubled if the current arc represents a double-tracked line[3]. The value for observation period t_{op} has to be chosen in a reasonable way. For the dimensioning of railway infrastructure in long-term planning processes usually a time slice representing peak hours of operation is taken into consideration. Practical experience in railway operations research has shown that such time slices have to cover at least 240 minutes to be meaningful. Here a time slice of 360 minutes is selected. According to the share of short distance traffic, which is modeled as basic traffic load, nearly one third of it is subtracted. So a time slice of about 240 minutes remains to the routing of long distance and cargo traffic within NDRI.

The occupation time, which is available to trains operationally using a line, is then calculated as follows:

$$max_{\mathrm{cc},ij} = \lceil 0.6 \cdot \underbrace{(360 \cdot (1 - 0.336))}_{t_{\mathrm{op}}} \cdot n_{\mathrm{tk},ij} \rceil = 144 \cdot n_{\mathrm{tk},ij} \text{ [min]}. \tag{6.7}$$

[3]Reminder: here, a double-tracked line consists of two tracks which are exclusively used in the same direction

6.1.6 Paths

For each traffic flow $\tau = (st_\tau, de_\tau, tr_\tau)$ the number of possible st-de-paths can take a maximum value[4] of

$$\lfloor (|N| - 2)!e \rfloor . \tag{6.8}$$

For $|N| = 15$ and even only one single traffic flow this results in a maximum of $1.69 \cdot 10^{10}$ paths, which in turn implies the same count of flow variables. Such magnitude of variables is not manageable for state of the art solvers but, as already mentioned in Section 3.3.2, the set (train) paths for a traffic flow usually is restricted to a meaningful (smaller) subset.

For the evaluation of the NDRI models several different sets of paths $\mathcal{P}_k^\tau$ for each traffic flow τ are used. Each set only contains paths up to a certain length. The restriction to paths of certain lengths is reasonable in relation to practice since journey times are of significant importance to infrastructures managers and have to be minimized to maximize profits. The lengths which are associated to such sets correspond to the lengths of direct links between st_τ and de_τ plus a certain offset value. The offset values are expressed as a percentage of the length of the direct link. So $\mathcal{P}_{10}^\tau$ is the set of paths of traffic flow τ whose lengths are smaller than 1.1 times $l_{1,st_\tau de_\tau}$:

$$\mathcal{P}_k^\tau := \{p \in \mathcal{P}^\tau \mid \sum_{(i,j)\in A} \delta_{ij}(p) l_{1,ij} \leqslant (1 + \frac{k}{100}) \cdot l_{1,st_\tau de_\tau}\} \tag{6.9}$$

Obviously, for each k and l with $k \leqslant l$ following equation is true:

$$\mathcal{P}_k^\tau \subseteq \mathcal{P}_l^\tau . \tag{6.10}$$

For the evaluation only values $k \in \{10, 25, 50, 75, 100\}$ are used and are referred to as *path percentages*.

Further reductions in the number of paths, can be achieved by the use of column generation. New paths can be priced in by solving a *shortest path problem*. If that would be a simple shortest path problem, the pricing problem could be solved efficiently. But since there are restrictions on the paths which are considered here, the pricing problem is a *constrained shortest path problem*. Such problems might be very time-consuming to

[4]Formula (6.8) is taken from van Hoesel et al. [vHKvdLS03].

solve, cf. [ID05]. Because of that, column generation for flow variables is not applied here.

6.2 Problem Size

This section presents instance-independent numbers of variables, constraints, and additional constraints induced by valid inequalities for both models: NDRI-MIP and NDRI-MIP$^{\text{WC}}$.

6.2.1 Variables

Table 6.5 shows the number of variables the MIP models consist of. As mentioned in the beginning of this chapter, NDRI-MIP is solved using two different approaches NDRI-MIP$^{\text{noCG}}$ and NDRI-MIP$^{\text{CG}}$. They differ in their number of binary config variables which are included into the model right from the beginning. This difference is reflected by the corresponding table entries for $y^{\text{used}}_{c_{ijs}}$. The number of variables which are added during column generation cannot be specified, since they vary from case to case. Flow variables $f^{\tau}_{t,p}$ model flow of traffic flow τ and type t on path $p \in \mathcal{P}^{\tau}$. In the problem formulations flow is expressed using flow function f_t. But the implementation of those models requires the use of variables. Flow variables have to be included for each train type, traffic flow and corresponding paths. This results in a large number of variables. This problem alleviated in two ways. On the one hand, the set of paths considered is restricted to paths of a certain length, as described in Section 6.1.6. On the other hand, flow variables are chosen to be continuous, which makes them easier to handle for the solver in comparison to integer variables. As a consequence, fractional trains may occur in solutions of NDRI instances. Because of the underlying macroscopic modeling approach this inaccuracy is of little significance, in comparison to the benefits obtained for the solution process. The value $|A|$ stated as number of decision variables w_{ij} is only true for two train types and two different worst-case timetables.

6.2.2 Constraints

In considering the number of constraints, two types of constraints are differentiated. Constraints necessarily needed to model the problem (Table 6.7) are analyzed separate from constraints induced by valid inequalities (Table 6.8), which are used to refine the

variable	**type**	**NDRI-MIP$^{\text{noCG}}$**	**NDRI-MIP$^{\text{CG}}$**	**NDRI-MIP$^{\text{WC}}$**
x_{ij}^{used}	binary	$\lvert A\rvert$	$\lvert A\rvert$	$\lvert A\rvert$
x_{ijs}^{used}	binary	$\lvert A^{\text{m}}\rvert$	$\lvert A^{\text{m}}\rvert$	$\lvert A^{\text{m}}\rvert$
$f_{t,p}^{\tau}$	continuous	$\lvert T\rvert \cdot \lvert\mathcal{P}\rvert$	$\lvert T\rvert \cdot \lvert\mathcal{P}\rvert$	$\lvert T\rvert \cdot \lvert\mathcal{P}\rvert$
$y_{c_{ijs}}^{\text{used}}$	binary	$\sum_{(i,j,s)}\lvert\mathcal{C}_{ijs}\rvert$	$\lvert T\rvert \cdot \lvert A^{\text{m}}\rvert$	-
w_{ij}	binary	-	-	$\lvert A\rvert$

Table 6.5: Number of variables used in problem formulations NDRI-MIP and NDRI-MIP$^{\text{WC}}$.

constraint type	**NDRI-MIP**	**NDRI-MIP$^{\text{WC}}$**
capacity	(3.46)	(3.54) and (3.55)
demand	(3.49)	(3.59) and (3.60)
arc used	(3.47)	(3.58)
config used	(3.48)	-
mix ratio	-	(3.56) and (3.57)

Table 6.6: Constraint types for models NDRI-MIP and NDRI-MIP$^{\text{WC}}$.

model towards a more efficient solution process. NDRI-MIP$^{\text{noCG}}$ and NDRI-MIP$^{\text{CG}}$ do not have to be differentiated since both approaches use the same set of model constraints and additional valid inequalities. Table 6.6 shows the terms which are used to address the different types of constraints. They are derived from roles corresponding constraints take within model. Analogous the number of variables w_{ij}, the number of mix ratio and capacity constraints of NDRI-MIP$^{\text{WC}}$ is only true for two train types and two different worst-case timetables. The labeling of valid inequalities is generated in the same way. Terms *MIS & MOS*, *MIM & MOM*, and γ *bound* used for single node cuts and the γ

type of constraint	**NDRI-MIP**	**NDRI-MIP$^{\text{WC}}$**
capacity	$\lvert T\rvert \cdot \lvert A\rvert$	$2 \cdot \lvert A^{\text{m}}\rvert$
demand	$\lvert T\rvert \cdot \lvert TD\rvert$	$\lvert T\rvert \cdot \lvert TD\rvert$
arc used	$\lvert A\rvert$	$\lvert A\rvert$
config used	$\lvert A^{\text{m}}\rvert$	-
mix ratio	-	$2 \cdot \lvert A\rvert$

Table 6.7: Number of constraints problem formulations NDRI-MIP and NDRI-MIP$^{\text{WC}}$ consist of.

valid inequality	**NDRI-MIP**	**NDRI-MIP$^{\mathrm{WC}}$**
MIS & MOS	$2 \cdot \lvert TD \rvert$	$2 \cdot \lvert TD \rvert$
MIM & MOM	$2 \cdot \lvert N \rvert$	$2 \cdot \lvert N \rvert$
connected	1	1
simple ub 1	-	$\lvert T \rvert \cdot \lvert A \rvert$
simple ub 2	-	$\lvert A \rvert$
γ bound	-	$\lvert T \rvert \cdot \lvert A^{\mathrm{m}} \rvert$

Table 6.8: Number of additional constraints which have to be added to the problem formulation of NDRI-MIP and NDRI-MIP$^{\mathrm{WC}}$, respectively, if the corresponding type of valid inequality is included.

bound are already introduced in Sections 4.1 and 4.3.2, respectively. The label *connected* denotes the connected-cut introduced in Section 4.2. Simple upper bounds for the traffic flow routed via an arc, which are described in Section 4.3.1, are labeled with *simple ub 1* (see (4.9)) and *simple ub 2* (see (4.10)).

6.3 Results

In this section computational result of some selected NDRI instances are presented. Problem instances are evaluated within 24 hours of solving time. Different NDRI instances are distinguished by

i) the model which is used: NDRI-MIP or NDRI-MIP$^{\mathrm{WC}}$ (abbreviated as WC), NDRI-MIP instances are further distinguished regarding the chosen solution approach: NDRI-MIP$^{\mathrm{noCG}}$ (NoCG) or NDRI-MIP$^{\mathrm{CG}}$ (CG),

ii) the path percentage which limits the maximum length of paths, cf. Section 6.1.6, five different values are used: $\{10, 25, 50, 75, 100\}$, and

iii) the number of nodes, smallest instances consist of 10 nodes; this number increase in steps of 5 nodes up to 40 nodes which results in this set of nodes: $\{10, 15, 20, 25, 30, 35, 40\}$.

At first benefits of column generation are evaluated.

6.3.1 Column Generation

To evaluate the benefits of column generation several NDRI-MIP instances are solved with and without column generation. In both cases, SCIP with Gurobi as LP solver is used. Since SCIP does not support multi-core processors only one single thread is used for the solving process. Results are compared using two benchmarks: solvability and speed-up. Further analysis regarding the solution process is done using bound evolution charts.

Figures 6.1 and 6.2 show which instances could be solved to optimality within the given time frame of 24 hours. Additionally, instances which reached a relative optimality gap[5] of at most five percent are listed. The speed-up in terms of the solving time is shown in Figure 6.3. Values displayed in the chart are speed-up factors. A factor of 2, for example, indicates that column generation solves the corresponding problem instance twice as fast as the solution approach without column generation. It turns out that the bigger the problem instance the higher the benefits gained from column generation. Speed-up factors up to 25 are achieved. Whereas Figure 6.3 presents a relative measurement of solving speeds, Figure 6.4 shows the absolute solving time values of problems instances solved with column generation. Longest solving time corresponds to the largest problem instance in terms of variables and constraints. The problem instance with 20 nodes and a path percentage of 10 consists of 3478 binary variables, 328 continuous variables and 2303 constraints. During column generation 5188 variables are added.

Detailed information about the solving process is visualized using bound evolution charts. They show how primal (incumbent) and dual (best bound) bounds evolve during the solution process. The primal bound improves whenever a new feasible solution with smaller objective value is found. Each of such events is separately marked in the chart. Dual bound improvements are mainly obtained by solved problem relaxations. In comparison to primal bound improvements such events do occur relatively numerous. Because of that, only start and end dual bound are marked.

Figures 6.5 and 6.6 show bound evolutions which are characteristic for NDRI instances solved using column generation. In this regard, the major part of the solving time is needed find the optimal solution and once it is found the optimality proof is obtained

[5]The relative optimality gap is used to determine the quality of the current solution and how far the solution process is progressed. It is an upper bound on the relative error made so far, cf. 3.1.2

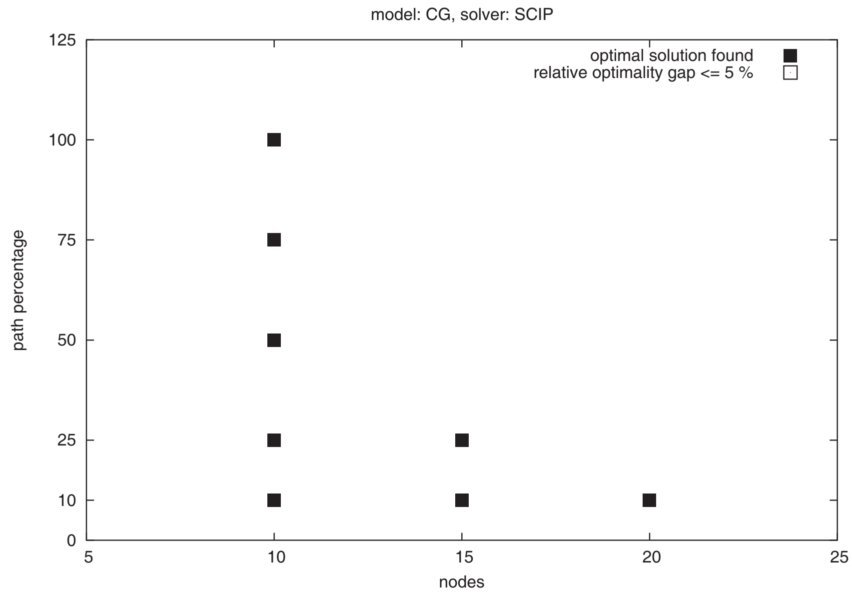

Figure 6.1: Solvability of NDRI-MIP$^{\mathrm{CG}}$ instances within 24 hours using SCIP.

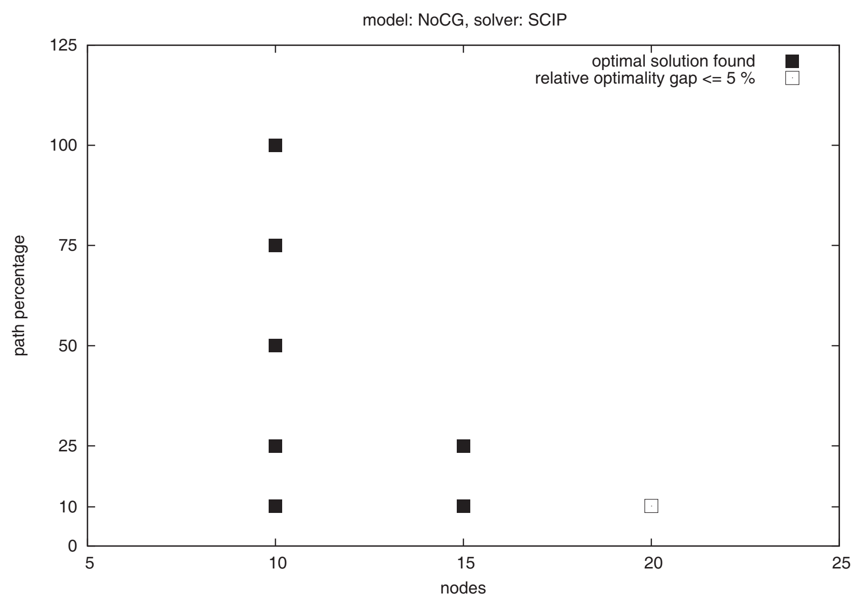

Figure 6.2: Solvability of NDRI-MIP$^{\mathrm{noCG}}$ instances within 24 hours using SCIP.

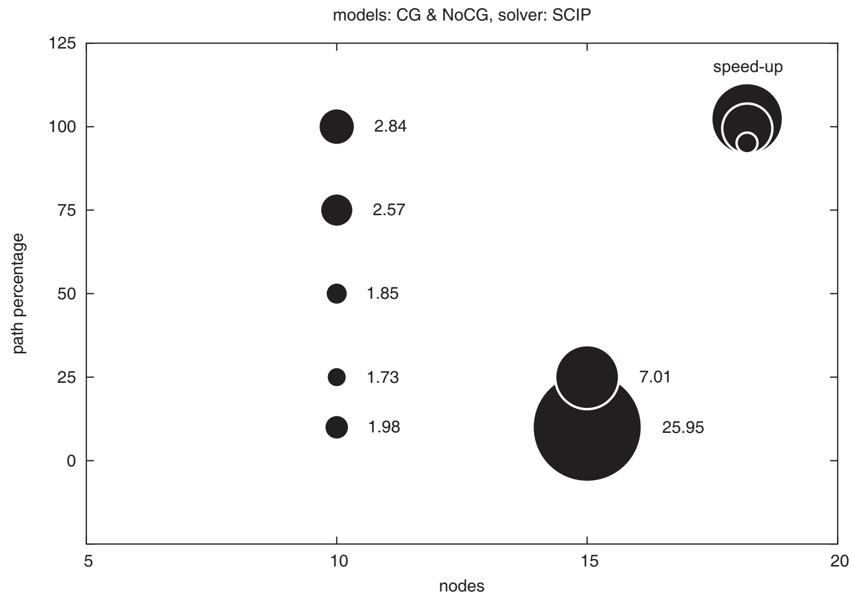

Figure 6.3: Speed-up factors achieved by column generation using SCIP.

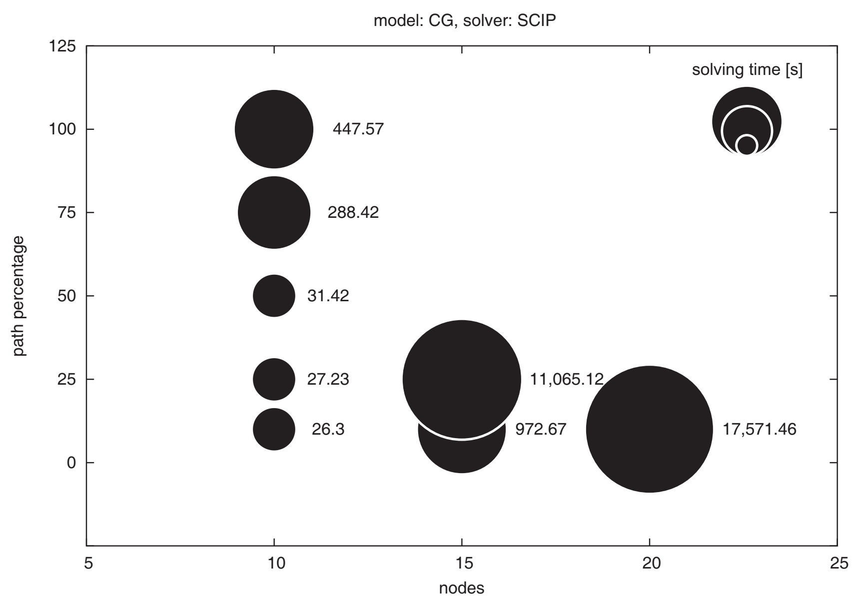

Figure 6.4: Solving times of NDRI-MIP^{CG} instances using SCIP.

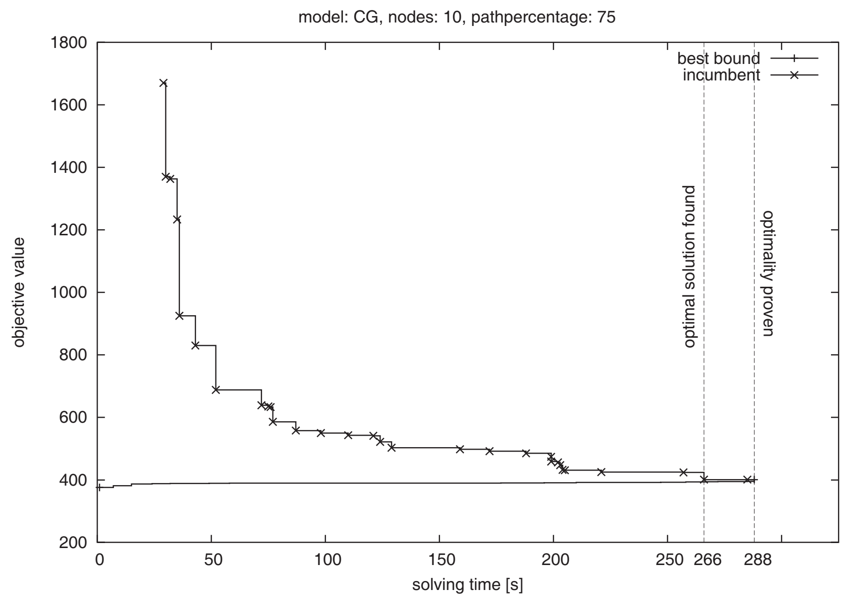

Figure 6.5: Bound evolutions. Characteristic for NDRI-MIP^{CG} instances: once the optimal solution is found, optimality is proven comparatively quick.

comparatively quick. It turns out that NDRI-MIP^{CG} instances need an average of close to 90% of the total solving time to find the optimal solution.

6.3.2 Without Column Generation

Whereas custom designed column generation requires the use of SCIP, solving without column generation permits the use of Gurobi as MIP solver. This has the benefit that multi-core (multi-thread) architectures can be exploited. Therefore, instances are evaluated on a cluster computer using 24 threads, cf. Section 5.4.2. Unfortunately, it turns out that solvability and solving times do not scale with the number of threads in a well-defined or predictable way. Section 6.3.4 briefly covers this topic. Nevertheless, solvability and solving times significantly improve even by the use of Gurobi as MIP solver. This is shown in Figures 6.7 and 6.8 which present, on the one hand, the considerably faster solving times and, on the other, the widened range of solvable problem instances. Largest problem instance solved to optimality is that one with 25 nodes and a path percentage of 10. It comes up with 65,480 binary variables, 868 continuous variables and 3,427 constraints. Largest problem instance which is solved up to an optimality gap

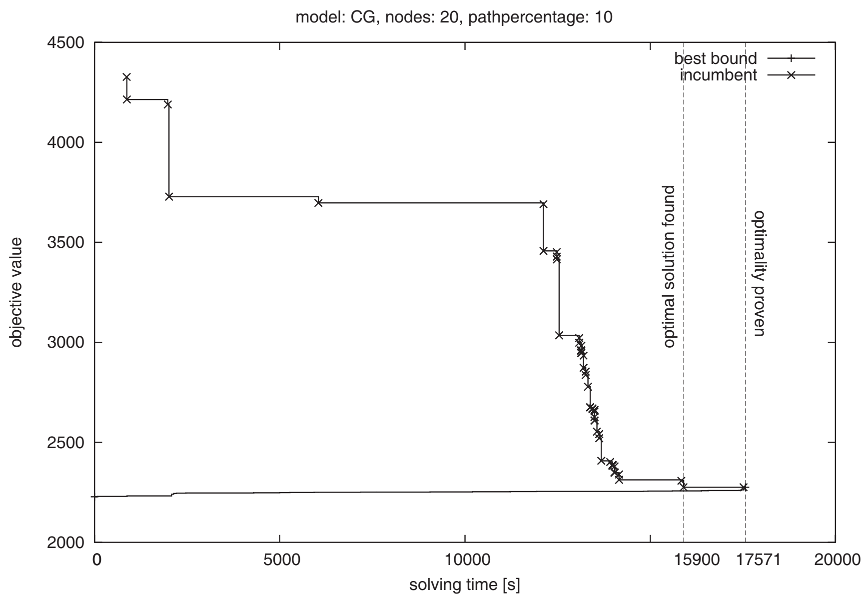

Figure 6.6: Bound evolutions. Characteristic for NDRI-MIP^{CG} instances: once the optimal solution is found, optimality is proven comparatively quick.

of 5% consists of 40 nodes and a path percentage of 10. It is composed of 182,130 binary variables, 4,976 continuous variables and 9,071 constraints.

To analyze how primal and dual bound evolve, instances which are solved to optimality are examined separate from instances which are solved up to an optimality gap of at most 5%. For the latter class of instances it turns out that the five percent threshold is reached very early in the solving process. So a solution, which could be useful in relation to practice, is obtained comparatively quick. The corresponding share of total solving time ranges from 0.01% to 8%, i.e. from 11 to 6,652 seconds. In this regard, another interesting quantity is the solving time spent until the best incumbent (within the time frame) is found. If, for example, the best incumbent is found very early and the current gap is very small, the current incumbent potentially is the optimal solution and the solver just fails to effectively proof optimality. This may indicate that the solver suffers from symmetries induced by *config*-variables, cf. Section 3.8, and cause of this seems to enumerate the whole B&B tree. An example is given in Figure 6.9. The five percent threshold is passed after 11 seconds and the best incumbent is found after 2,323 seconds, at this point the gap is 0.34%. During remaining 84,077 seconds (= 23.35 hours) only

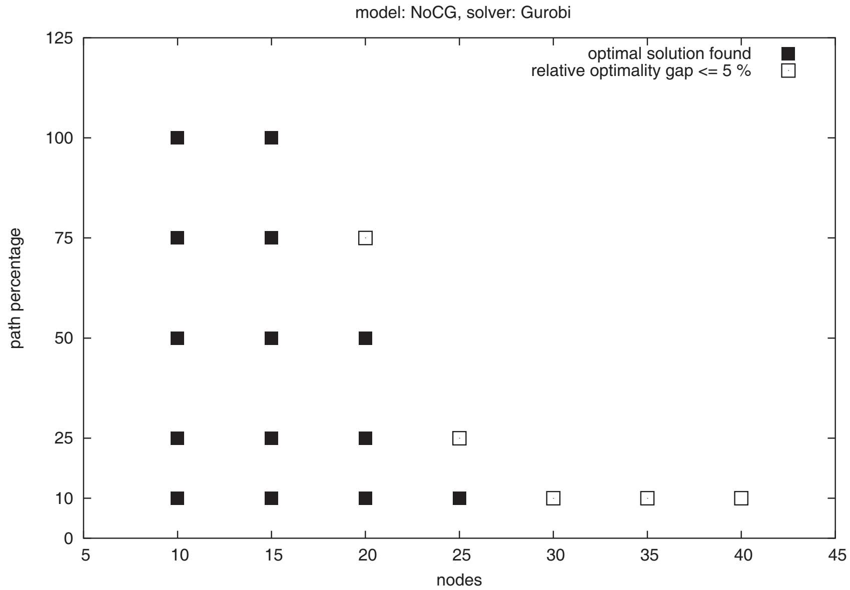

Figure 6.7: Solvability of NDRI-MIP$^{\mathrm{noCG}}$ instances within 24 hours using Gurobi.

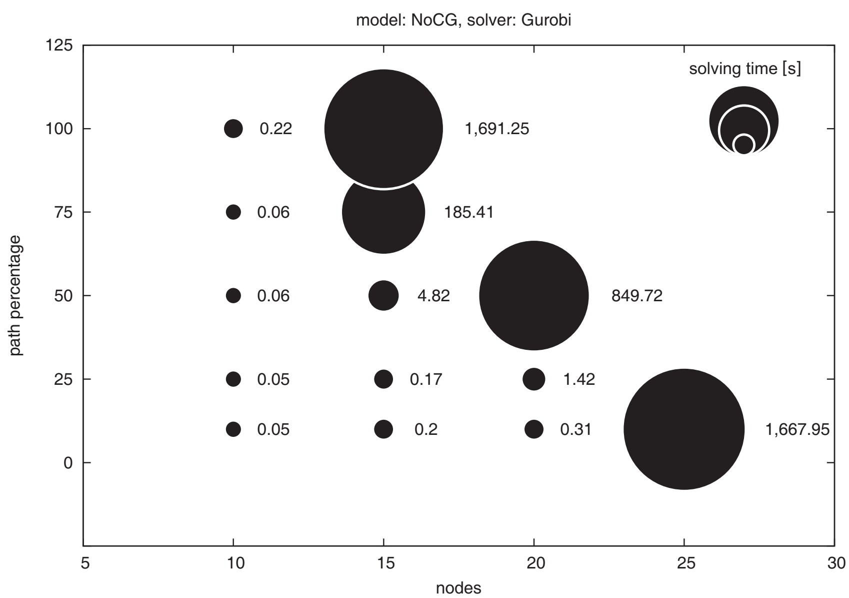

Figure 6.8: Solving times of NDRI-MIP$^{\mathrm{noCG}}$ instances using Gurobi.

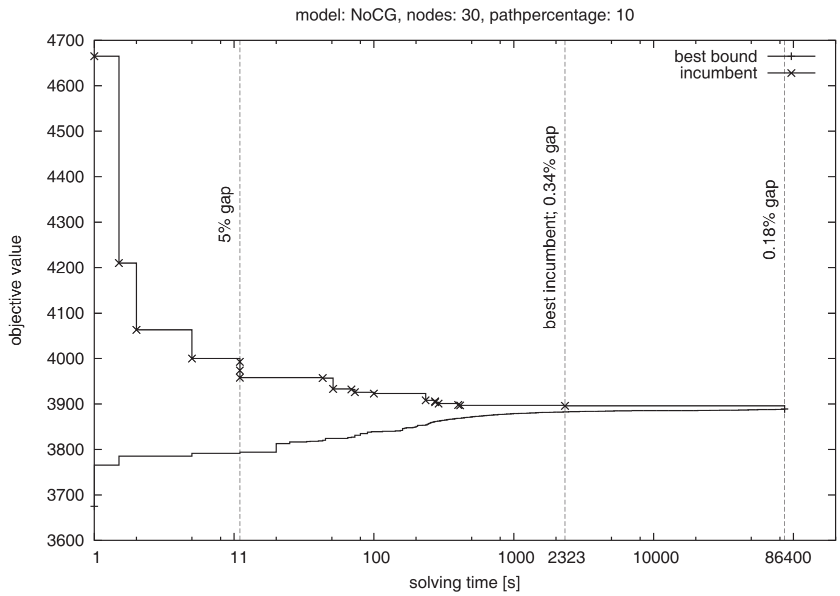

Figure 6.9: Bound evolutions. 5% gap is reached very early. Best incumbent within 24 hours of solving is found very early, too. No significant improvement during remaining 23 hours.

very small improvements of the dual bound can be observed and the gap reduces only to 0.18%. But such behavior cannot be generalized. The time share needed to find the best incumbent within the time frame varies from 2.7%, as mentioned above, up to 98.8% (Figure 6.10).

Bound evolutions of instances which are solved to optimality mainly reveal same characteristics described for NDRI-MIP^{CG} instances. Over 90% of the total solving time is spent to find the optimal solution. Once it is found, optimality is proven comparatively quick. Figure 6.11 shows an example of that behavior.

The solution of the largest problem instance which is solved to optimality is visualized in Figure 6.13, the corresponding traffic demand in Figure 6.12. Each undirected arc in the demand graph represents the number of trains which have to be routed from one incident node to the other and vice versa. Between Bonn and Frankfurt am Main, e.g., 24 trains each have to be routed from Bonn to Frankfurt and from Frankfurt to Bonn. This symmetric demand structure is created by the gravity model used for the traffic flow generation. To simplify the visualization, trains of both types are summed up to one

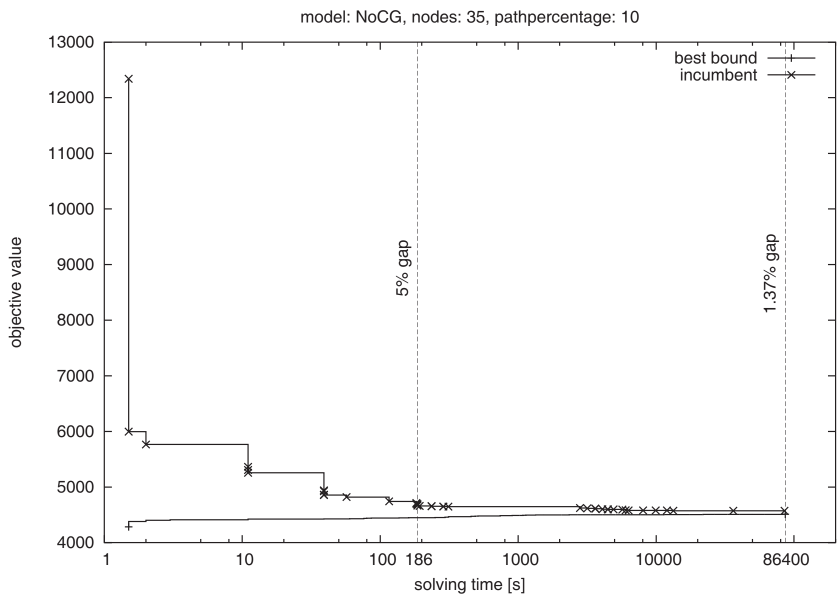

Figure 6.10: Bound evolutions. 5% gap is reached very early. In contrast to Figure 6.9, the best incumbent is found shortly before the end of the 24 hour time frame.

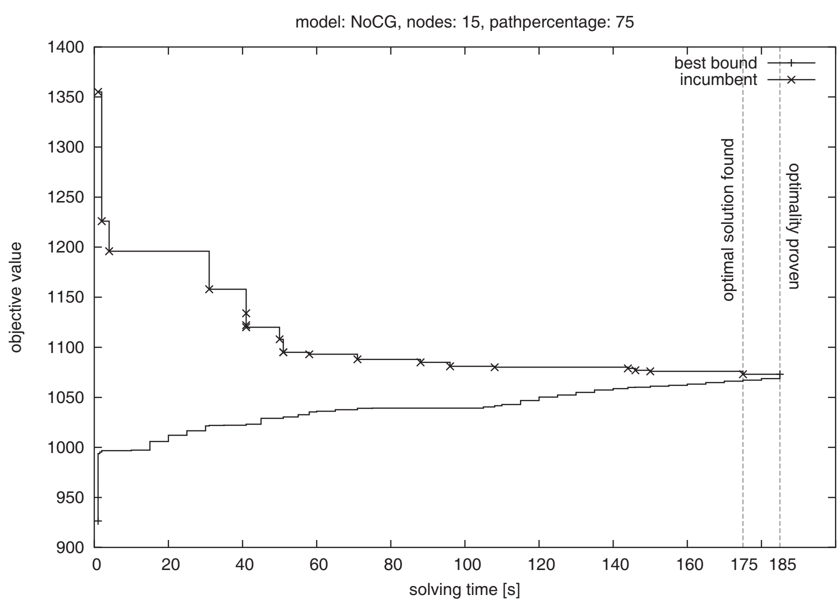

Figure 6.11: Bound evolutions. Characteristic for NDRI-MIP$^{\text{noCG}}$ instances: once the optimal solution is found, optimality is proven comparatively quick.

train count. The optimal network design, calculated by Gurobi, is shown in Figure 6.13. Each undirected arc represents the line in its specific stage of extension which has to be built for both directions.

For this problem instance it has to pointed out that the length of a path is restricted to 1.1 times the length of a direct connection, i.e. a path percentage of 10. This may lead to design decisions which seem to be unusual. The following example illustrates that. With regard to the design costs, it is better to build a line between Berlin and Frankfurt a. d. O. and to route the traffic flow from Dresden to Frankfurt a. d. O. via Berlin. But due to the restriction in the length of a path, such routing is not available. To avoid these solutions (without a link between Berlin and Frankfurt a. d. O.), the number of available paths or the path percentage, respectively, has to be increased. If the alteration is done globally, like for the evaluation in this work, the solvability of the problem instance is affected, because the total number of path variables may increase very fast. In relation to practice, this problem does not occur to this extent. Instead of global fixings of path lengths, the network designer usually creates individual sets of paths for each traffic flow. So reasonable[6] sets of train paths for each traffic flow are derivable without the negative effects of the global path percentage alteration. Nevertheless, global path percentages are used here for the evaluation, because the information how to derive reasonable sets of trains paths for given traffic flows was not available for this work.

6.3.3 Worst-case Capacity consumption

To overcome problems arising from *config*-variables (number of binary variables and symmetry) at model level, NDRI-MIP$^{\text{WC}}$ is introduced in Section 3.9. Instead of using *config*-variables, the linearization of the capacity constraint is based on a timetable which models the worst-case capacity consumption of a line. This reduces the number of binary variables and avoids symmetry problems.

Design costs[7] of NDRI-MIP$^{\text{WC}}$ optimized networks are, in comparison to NDRI-MIP networks, on average only 5% higher. This keeps NDRI-MIP$^{\text{WC}}$ attractive in relation to practice. Design costs for both cases are visualized in Figures 6.14 and 6.15. In this regard, NDRI-MIP$^{\text{CG}}$ and NDRI-MIP$^{\text{noCG}}$ clearly do not have to be differentiated

[6] In particular cases, longer paths or longer journey times, respectively, may be accepted if significantly savings regarding the design costs are possible.

[7] Design costs are optimal objective values of NDRI instances, i.e. the life-cycle costs of corresponding optimal networks of railway infrastructure.

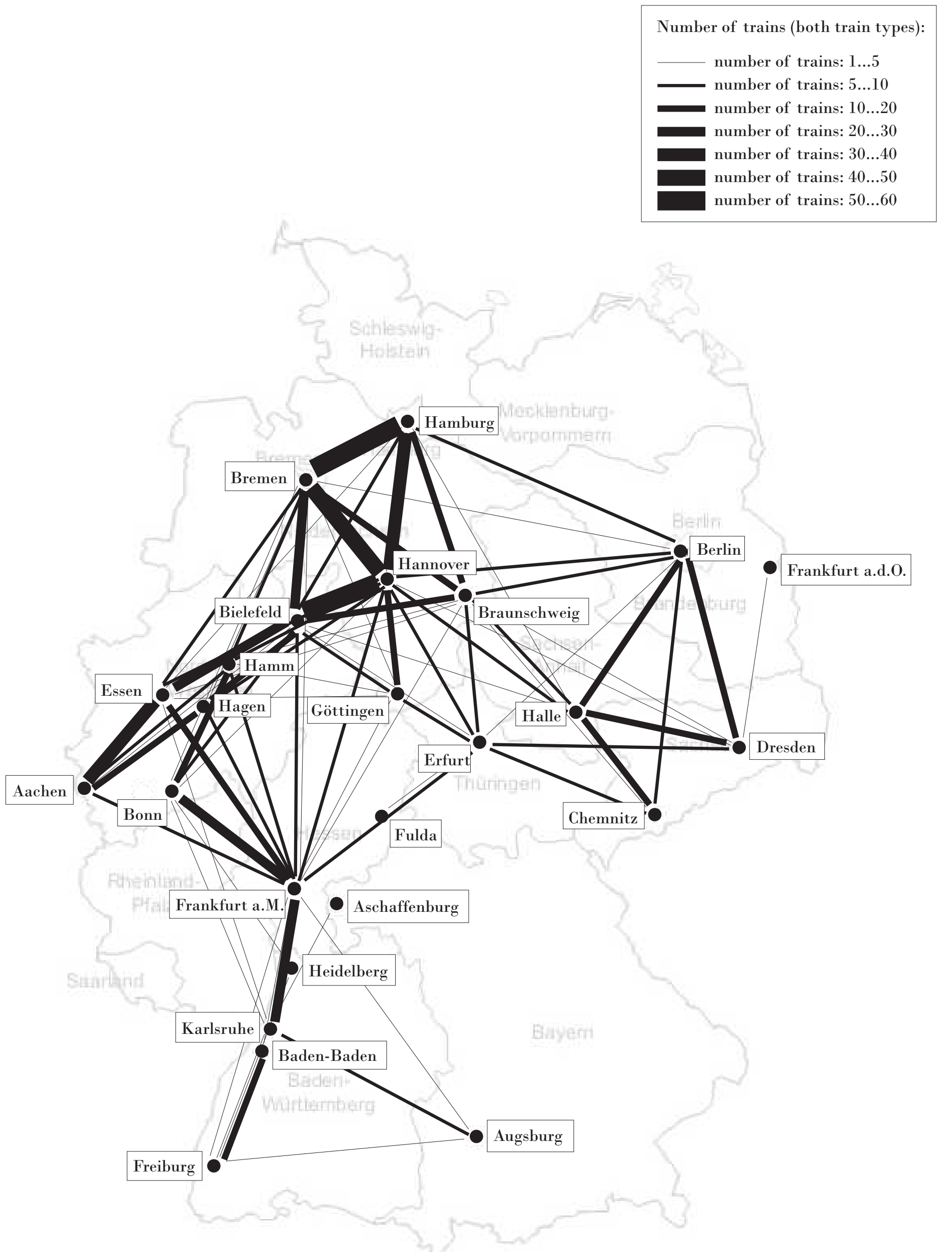

Figure 6.12: The Traffic demand for a problem instance with 25 nodes and a path percentage of 10. The train counts for both train types are summed up. The traffic flows for each relation are the same for both directions.

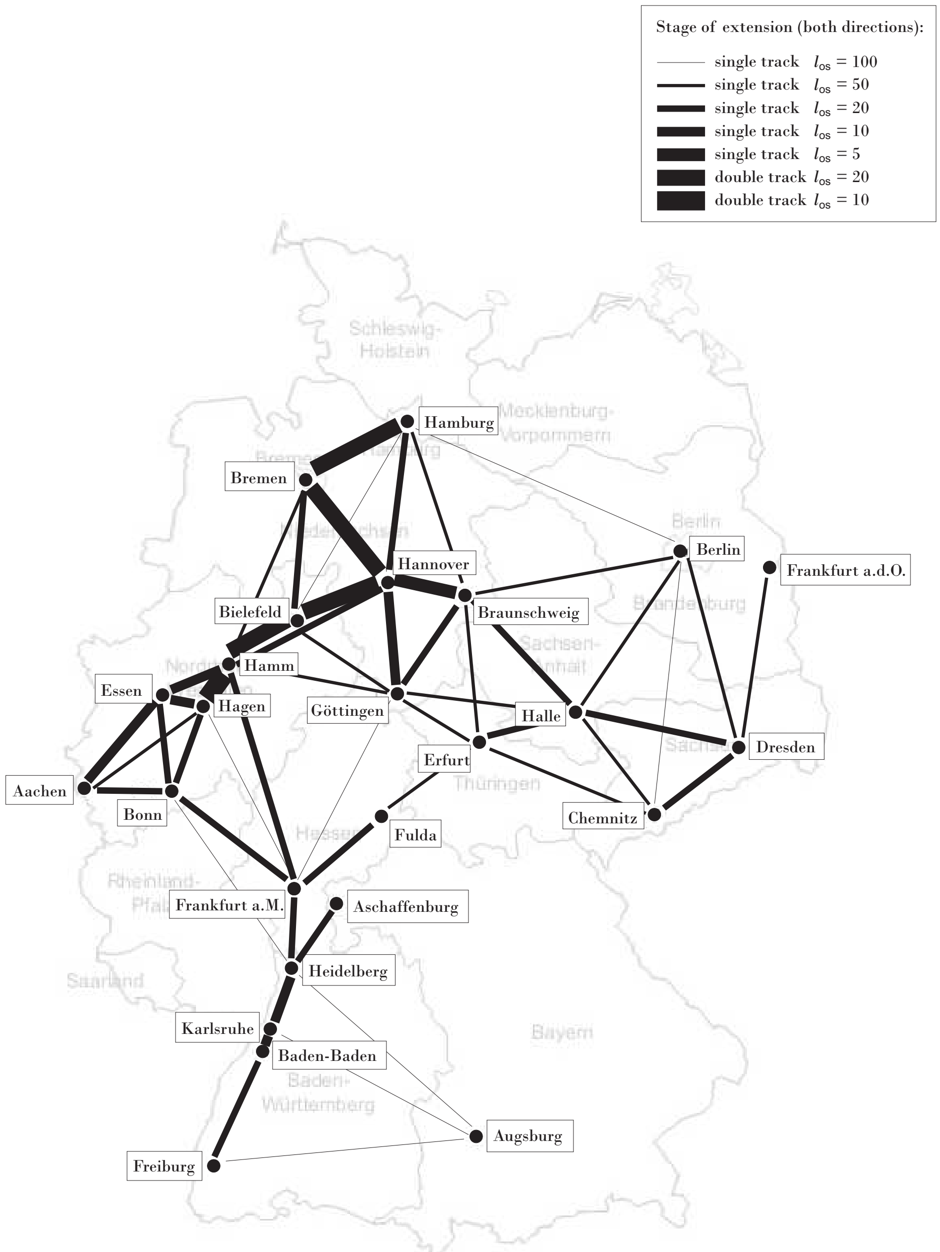

Figure 6.13: The optimal network graph for a problem instance of NDRI-MIP$^{\text{noCG}}$ with 25 nodes and a path percentage of 10. The corresponding traffic demand is shown in Figure 6.12. The stage of extension of each connection is the same for both directions.

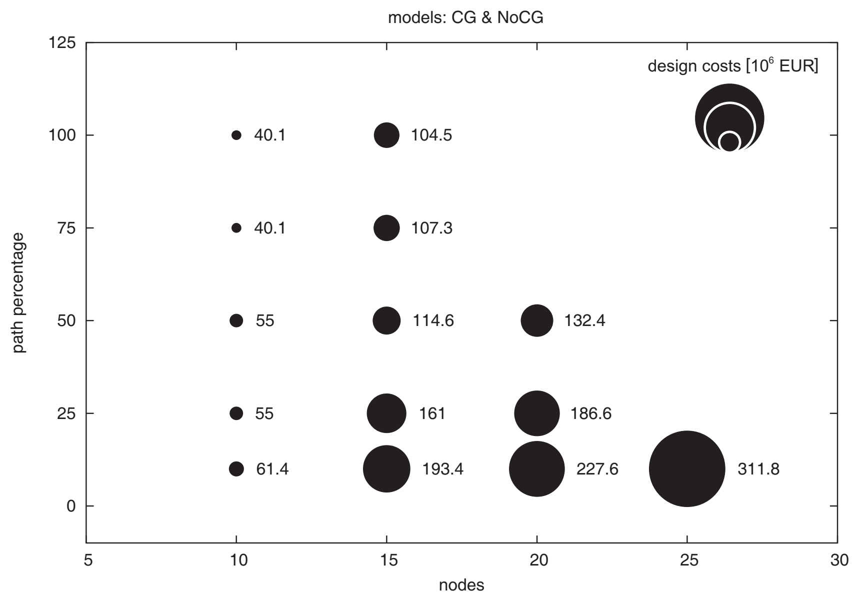

Figure 6.14: Design costs of NDRI-MIP instances.

since design costs do not depend on the solution approach. As one would expect, in both cases design costs increase with the number of nodes. Furthermore, design costs decrease with rising path percentages. This is caused by bundling effects: increasing sets of available paths lead to an increasing number of arcs which are common element of paths of different traffic flows. So flows can be bundled on that arcs which in turn permits higher degrees of utilization of such arcs. This results in fewer arcs needed to route all traffic flows and as a consequence in lower design costs.

Despite mentioned advantages of NDRI-MIP$^{\text{WC}}$, the solver, which is again Gurobi on a cluster computer, is not able to perform better on NDRI-MIP$^{\text{WC}}$ instances in comparison to NDRI-MIP$^{\text{noCG}}$. Figures 6.16 and 6.17 ground this observation. In this regard, largest problem instance solved to optimality consist of 25 nodes and a path percentage of 10. It is composed of 6,352 binary variables, 868 continuous variables and 10,445 constraints.

Largest problem instance which is solved up to an optimality gap of 5% is that on with 30 nodes and a path percentage of 10. It consists of 9,868 binary variables, 1,684 continuous variables and 15,303 constraints.

Analysis of the primal and dual bound evolution again is two-part. Both, 5% gap instances and instances which are solved to optimality, do not reveal a characteristic or

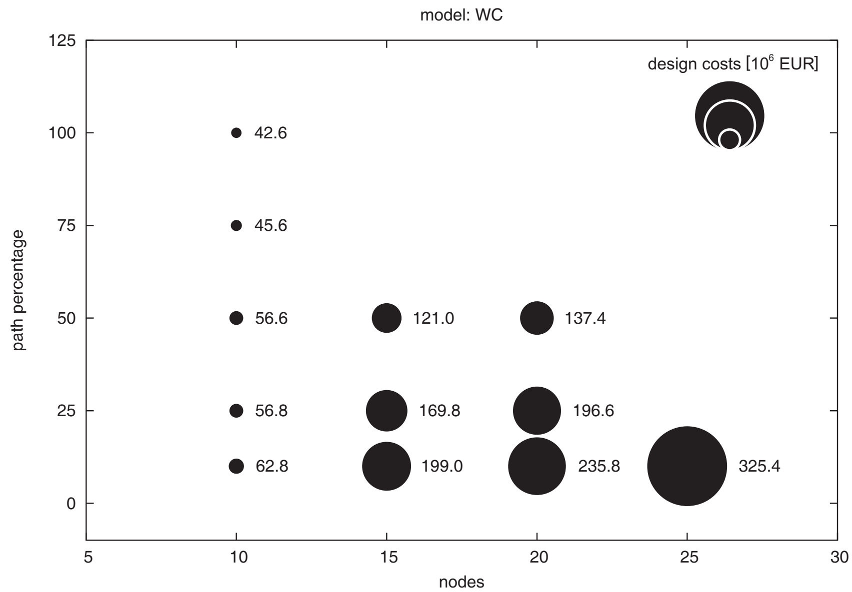

Figure 6.15: Design costs of NDRI-MIP$^{\mathrm{WC}}$ instances.

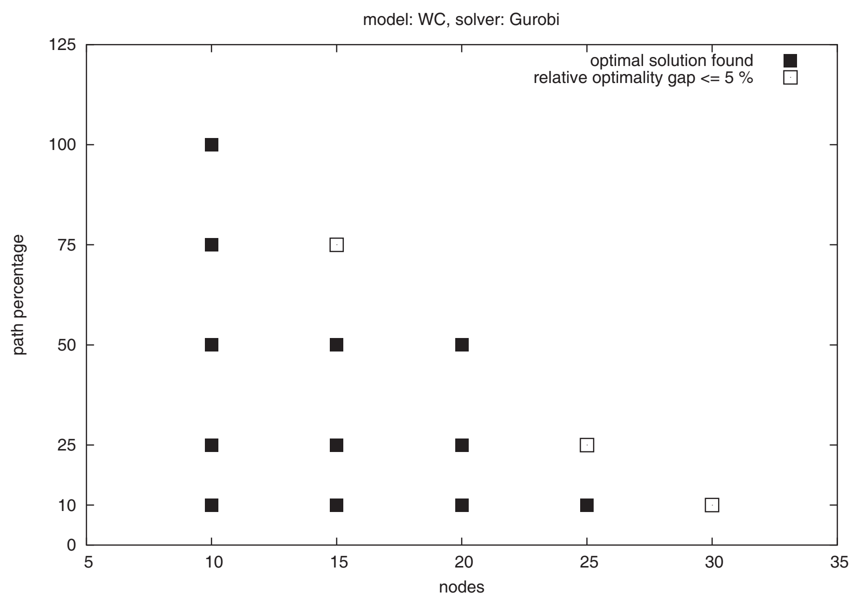

Figure 6.16: Solvability of NDRI-MIP$^{\mathrm{WC}}$ instances within 24 hours using Gurobi.

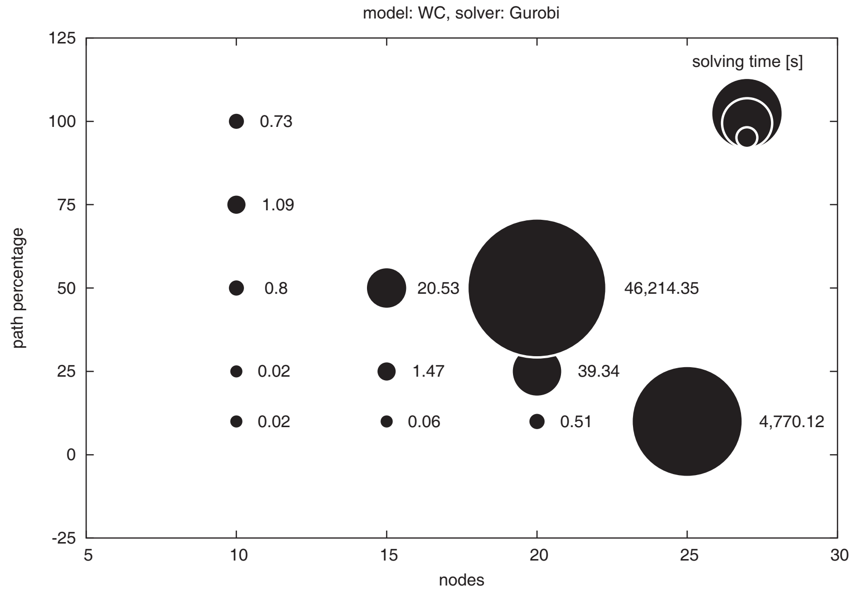

Figure 6.17: Solving times of NDRI-MIP$^{\mathrm{WC}}$ instances using Gurobi.

predictable course of the solving process. The optimal solution, for example, is found in the beginning (Figure 6.18), at the end (Figure 6.20), or somewhere in between (Figure 6.19).

6.3.4 Scalability

As already mentioned, solvability and solving times do not scale with the number of threads in a well-defined or predictable way. This may be caused by some different factors. Besides the problem formulations itself, there are many other technical and hardware dependent reasons why efficiency drops as the number of threads is increased. A very good treatise on this topic is the tech report [KRS11] written by Koch, Ralphs, and Shinano. Since this topic goes beyond the scope this work, it is not discussed in detail. Nevertheless, two benchmarks are defined following [KRS11]. They are evaluated for two problem instances to give examples of the scaling behavior.

The *efficiency* E_n of a program exploiting n threads is defined as

$$E_n := \frac{(T_1/T_n)}{n}, \tag{6.11}$$

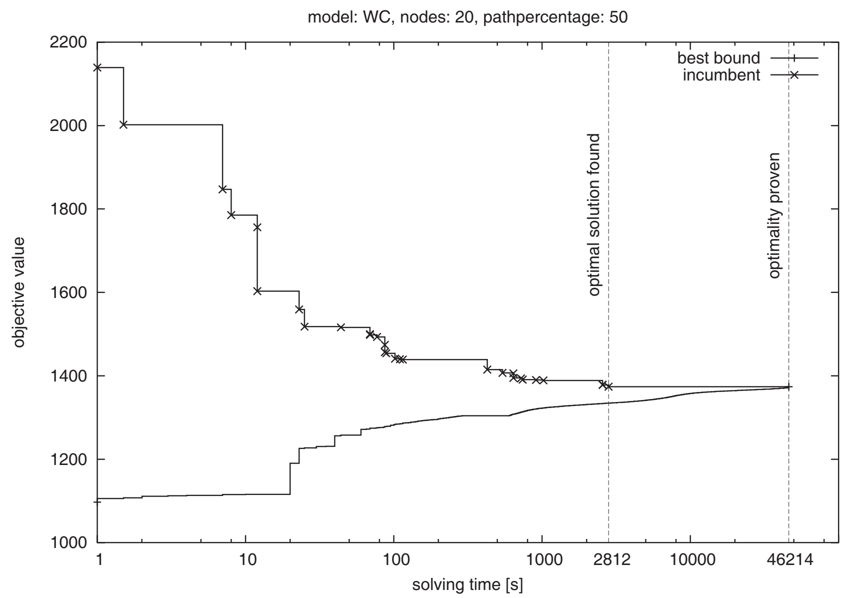

Figure 6.18: Bound evolutions. No characteristic behavior, cf. instances shown in Figures 6.19 and 6.20.

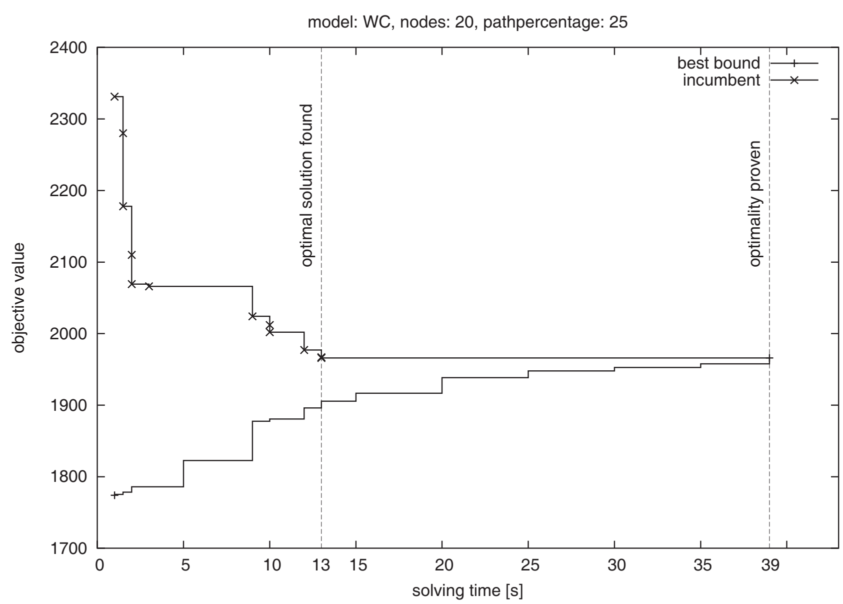

Figure 6.19: Bound evolutions. No characteristic behavior, cf. instances shown in Figures 6.18 and 6.20.

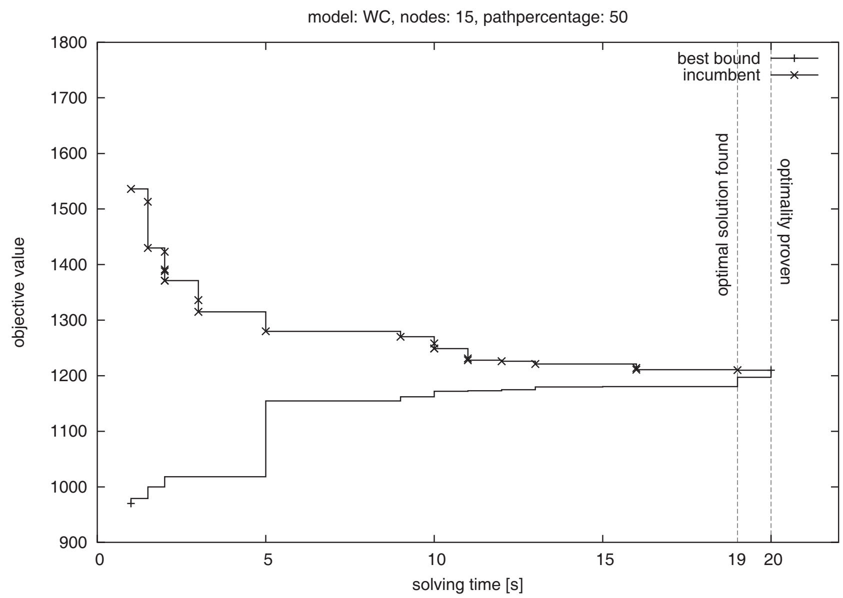

Figure 6.20: Bound evolutions. No characteristic behavior, cf. instances shown in Figures 6.18 and 6.19.

where T_1 is the running time using 1 thread and T_n the running time using n threads. Efficiency represents the rate of utilization of the threads. Perfect scaling behavior is achieved if $E_n = 1$ for all n. The *speed-up* S_n then is defined as follows:

$$S_n := nE_n = \frac{T_1}{T_n}. \tag{6.12}$$

Speed-up and efficiency are both determined for one NDRI-MIP$^{\text{noCG}}$ and one NDRI-MIP$^{\text{WC}}$ instance with 25 nodes and a path percentage value of 10. Results shown in Figures 6.21 and 6.22 do not reveal a well-defined scalability of NDRI instances on multi-core architectures regardless of the modeling approach used. This is not a NDRI specific phenomenon or problem, since Koch et al. obtained similar results for some of the problems analyzed in [KRS11], e.g., the *Cyclic Railway Timetabling Problem.*

6.3.5 Conclusions

In the following main results are summarized, possible implications examined, and performance improvements discussed.

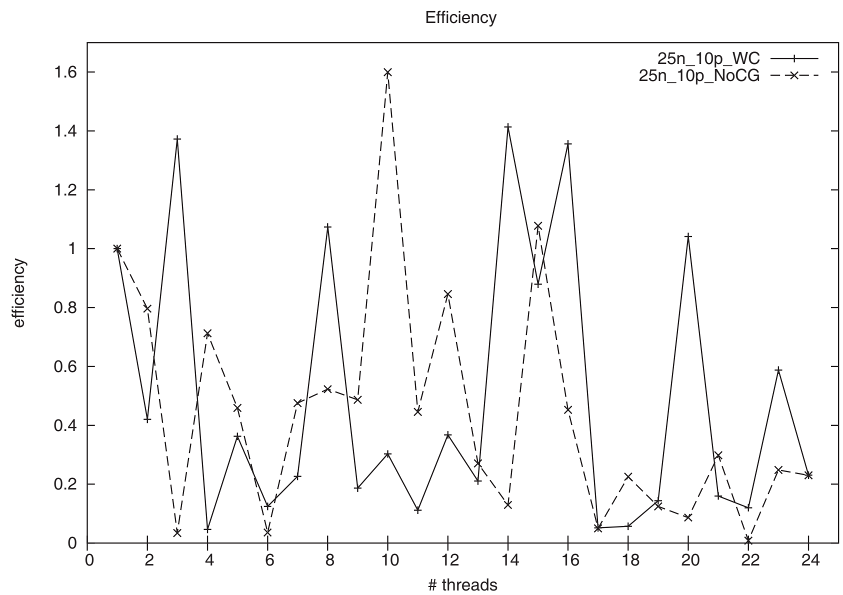

Figure 6.21: Two examples for the evaluation of efficiency with increasing number of threads.

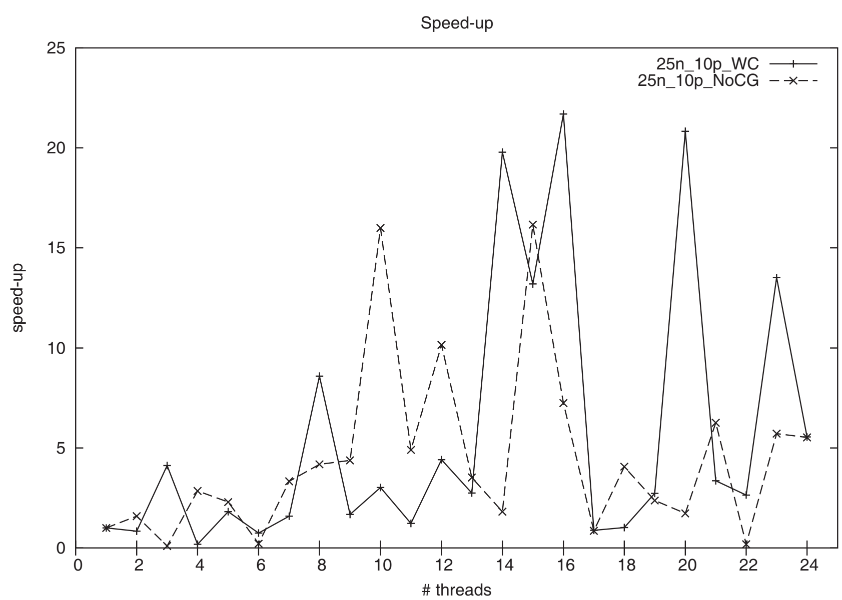

Figure 6.22: Two examples for the evaluation of speed-up with increasing number of threads.

i) In terms of solvability and solving times NDRI-MIP using *config*-variables to model capacity consumption performs best. NDRI-MIP$^{\text{noCG}}$ instances with up to 25 nodes can be solved to optimality and instances up to 40 nodes reach and optimality gap less than 5% within the time frame of 24 hours. These instances consist of 65,480 (182,130) binary variables, 868 (4,976) continuous variables and 3,427 (9,071) constraints. Column generation in comparison to no column generation relatively performs even better (speed-up factors up to 25 are achieved) but the SCIP framework impedes a better absolute performance. A user defined column generation having the opportunity to exploit Gurobi would probably solve significantly larger NDRI-MIP instances, but up to now the Gurobi interface does not provide this functionality.

ii) Analyzes of bound evolutions for both models do not reveal a characteristic behavior of the solving processes, but hints regarding solving problems are provided. Some evaluations of NDRI-MIP$^{\text{noCG}}$ instances show that after a relatively small amount of time the solver has serious difficulties to find better primal solutions or significant improvements of the dual bound. In such cases, an extension of the time frame from 24 hours to 48 or 96 hours is not very promising, since the branch-and-bound process in such case seems to degenerate to a full enumeration of the branch-and-bound tree which may consume an arbitrarily large amount of time. The degeneration suggests that the solver suffers from symmetries induced by config-variables and so, smart designed symmetry breaking inequalities may produce relief, cf. Section 3.8.

iii) For both models, solvability and solving times do not scale with the number of threads in a well-defined or predictable way. So increasing the number of threads (CPUs) beyond 24 threads (12 CPUs) to achieve a better performance seems to be not very promising.

7 Summary, Conclusions & Outlook

This chapter at first summarizes key facts of the research which is presented in this work. After that, concluding remarks are drawn and an outlook shows improvement potential which can be explored by future research.

7.1 Summary

The network design problem for railway infrastructure aims to find a network of railway infrastructure which meets given traffic demands at lowest possible design costs. This problem, for example, comes up in long-term infrastructure planning processes. Objective of this planning process is the adaptation of railway infrastructure to future requirements and infrastructure manager's objectives, respectively. In this context, the development of network variants is mainly based upon infrastructure manager's experience. Since infrastructure investments are highly capital-intensive, it is desirable to determine provable optimal design decisions. This work presents an approach to optimally solve the network design problem for railway infrastructure using linear programming.

Railway infrastructure is modeled macroscopically, which is a reasonable level of precision for long-term planning processes, using nodes (stations) and arcs (lines). Stations are assumed to be equipped with unbounded capacity to reduce complexity, so design issues are the network topology and the capacity of lines. Capacity of a line depends on the stage of extension which in turn mainly depends on the number of tracks and the average length of overtaking sections. Different stages of extension of the same line are modeled as parallel arcs between the corresponding nodes. In this way a complete multi-arc network is created.

Traffic demand consists of traffic flows which in turn consist of sink node, source node and a number of trains of different train types. These multicommodity flows have to be routed through the complete network graph to meet their demands.

Resulting multicommodity flow problem on a complete multi-graph is stated as a non-linear optimization problem in a way such that for an optimal solution those arcs which are necessary to route the different flows determine the network with minimum design cost. This clearly determines the topology of the network. Line capacities are determined by forcing the routing to use at most one of the parallel arcs of one node-to-node relation. To decrease the number of possible routes for each traffic flow, sets of paths are introduced. They limit the length of a route to the length of a direct connection between corresponding source and sink nodes plus a surcharge given as percentage of the length of the direct connection. During evaluation different (path) percentages are used. This restriction to paths of certain lengths is reasonable in relation to practice since journey times are of significant importance to infrastructure managers and have to be minimized to maximize profits.

Mentioned non-linearity is brought into the model by capacity constraints which have to be satisfied by each feasible routing. Capacity constraints limit the usability of each arc by restricting the capacity consumption of a mix of trains to 60 percent of the observation time, which models the satisfying quality of service of mixed traffic lines regarding UIC Code 406. The capacity consumption itself is determined using timetable independent expected service times. Timetable independence is motivated by the fact that future timetables are not known during design time. The model then is transformed into a linear one, which is highly desirable since very powerful algorithms and solvers become applicable. The transformation is done using configurations similar to cutting patterns used in the well-known cutting stock problem. Each configuration consists of a tuple of train counts of different train types which represent one way how to fully exploit the capacity of the corresponding arc. The resulting (linear) mixed integer programming problem NDRI-MIP is then further refined by valid inequalities. They tighten the feasible region of the LP relaxation to achieve better approximations of the MIP feasible region which improves the solving process.

But configuration variables raise two problems. On the one hand, they induce symmetry in the problem, in sense that for feasible solutions it is often possible to vary in the assignment of values to config-variables without changing feasibility or cost. Primal and dual bound evolution seems to give the indication that the solver suffers from such symmetries. A suggestion how this problem may be overcome is given in the outlook. On the other hand, NDRI-MIP instances consist of a huge number of binary config-variables, which is crucial for MIP solvers, too. This problem is handled by column generation.

Thereby the solution process starts with a small subset of config-variables. Additional variables are added to the problem only if they improve the solution, which is currently the best during the solution process. The implementation of column generation requires a solver interface which provides deep control of the solving process. SCIP 2.0.1, which is used with Gurobi 4.5 as embedded LP solver, offers such an interface. Within this framework, significant speed-ups of the solving time up to a factor of 25 are achieved by column generation.

Nevertheless, best results in terms of the size of solvable problem instances are obtained without column generation but the use of Gurobi as stand-alone MIP solver. Gurobi does not offer the opportunity to embed customized column generation but it is a more powerful solver than SCIP and, furthermore, is able to exploit multi-core processors.

Largest instances, which can be solved to optimality within a time frame of 24 hours consist of 25 nodes and a path percentage of 10, composed of 65,480 binary variables, 868 continuous variables, and 3,427 constraints. In relation practice instances which can be solved within the time frame of 24 hours up to a relative error, i.e. the so-called optimality gap, of 5% are of interest, too. In this regard, a problem instance with 40 nodes and a path percentage of 10 can be solved. It is composed of 182,130 binary variables, 4,976 continuous variables, and 9,071 constraints.

Besides the use of config-variables another approach to handle the non-linearity, which is induced by timetable independent derivation of line capacity consumption, is introduced in this work. The corresponding model is called NDRI-MIP$^{\mathrm{WC}}$. Instead of a timetable independent derivation, the capacity consumption is determined using worst-case timetables. A network designed by this means provides an upper bound on the infrastructure needed to satisfy the given traffic demand, which creates timetable independence too. NDRI-MIP$^{\mathrm{WC}}$ is defined for two different train types, which produces two worst-case timetables. Depending on which train type outnumbers the other in the traffic flow of the current line one of the two timetables is used to determine the capacity consumption. The corresponding decision process is modeled using a big-M approach. To preserve tight LP relaxations big-M values, which are used for capacity constraints, too, are chosen as small as possible. This is achieved by the solution of small mixed integer programs.

As one would expect, design costs increase using NDRI-MIP$^{\mathrm{WC}}$ in comparison to

NDRI-MIP but they are on average only 5% higher, which keeps this approach attractive in relation to practice.

Despite the advantages of a model without config-variables, evaluation reveals a better performance of NDRI-MIP in comparison to NDRI-MIP$^{\text{WC}}$. So, NDRI-MIP is the model of choice to solve the network design problem for railway infrastructure and because of that conclusions and outlook are focussing on it.

7.2 Conclusions & Outlook

NDRI-MIP proves to be a good and promising model for the network design problem for railway infrastructure. In the following, at first concluding remarks on strengths of the model are made then improvement potential is discussed.

7.2.1 Conclusions

The big strength of NDRI-MIP lies in its flexibility, which is mainly created by the use of configuration variables. The special characteristic of capacity of railway lines is its dependency on infrastructural and operational properties. Configuration variables not only combine these properties in one quantity, but also provide a kind of interface between the mathematical model and the models for infrastructure, operation, and capacity. This enables changes of these models without changing the mathematical model. So, on the one hand, instead of expected service times, every other measure may be used, e.g., scheduled and unscheduled waiting times which are state of the art of analytic models of railway operations research in Germany. On the other hand, infrastructural measures for capacity can be modeled more fine-grained or microscopic, respectively. In addition, the number of different train types can be increased without fundamentally changing the model. However, the number of config-variables will increase significantly.

Another strength of the model lies in the use of paths to restrict the routing of traffic flows. A set of paths can be assigned separately to each traffic flow. This is not only reasonable in relation to practice, e.g., if some station necessarily have to be on each available route of a traffic flow, but also very beneficial to the performance of solvers which only have to cope with a restricted set of routings.

Besides the design of network graphs from scratch, NDRI-MIP can be used to support design decisions concerning large existing networks of railway infrastructure. Infrastructure managers may be faced with problems like the linking of new stations to existing networks or the removal of bottlenecks by infrastructure upgrades. In such cases, parts of the network and the current routing can be fixed by fixing the corresponding variables to the desired values. For new lines or parts of the network which have to be revised, every possible embedding into the existing network has to be included into the problem formulation. The solving of such instances then results in the optimal embedding of these new or revised parts of the network. In this manner, very large networks are manageable as long as the size of the subnetwork, whose embedding is subject to design decisions, has a manageable size.

7.2.2 Outlook

Evaluation reveals available opportunities to improve the performance of the solutions process in terms of solvability and solving times of large problem instances.

Only limited improvements can be expected by an increasement in the number of CPUs or threads, respectively, because solvability and solving times do not scale with the number of threads in a well-defined or predictable way. But further research in this field may uncover whether there exists an optimal number of threads. Generally extending the time frame from 24 hours to larger time frames to solve large problem instances is not very promising, too. On the one hand, long solving times are not desirable in relation to practice and, on the other, some evaluations suggests that the time frame is not the limiting factor. So, limiting factors are to be sought in the structure of the model and the solution process.

In this regard, the determination of symmetry breaking inequalities seems to be a very promising option, since isomorphic solutions then will be prevented, which supports pruning in the branch-and-price tree and in turn will provide faster convergence of the branch-and-price.

Another option is an extension of the branch-and-price algorithm to a branch-and-price-and-cut algorithm. Additional cutting planes which then will added during the solution process will tighten the LP feasible region, which leads to improved bounds, better pruning, and again faster convergence of the solution process.

Finally, column generation, which is already applied here, bears great potential for further performance improvements. Gurobi currently performs better without column generation than SCIP with column generation, so if future versions of Gurobi will al-low customized column generation, significant improvements in terms of solvability and solving times can be expected.

Bibliography

[Abe72] G. Aberle, *Verkehrsinfrastrukturinvestitionen im Wachstumsprozeßentwickelter Volkswirtschaften*, Buchreihe des Instituts für Verkehrswissenschaft an der Universität zu Köln, pp. 13–14, R. Willeke, 1972.

[Ach09] T. Achterberg, *SCIP: Solving constraint integer programs*, Mathematical Programming Computation **1** (2009), no. 1, 1–41, `http://mpc.zib.de/index.php/MPC/article/view/4`.

[AG09] DB Mobility Logistics AG, *2009 Annual Report*, 2009, www.deutschebahn.com/dbml.

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows : Theory, algorithms, and applications*, Prentice Hall, Upper Saddle River, New Jersey, 1993.

[Ben05] J.-P. Bendfeldt, *Möglichkeiten der Standartisierung in der Infrastrukturplanung von Eisenbahnknoten*, Wissenschaftliche Arbeiten des Instituts für Verkehrswesen, Eisenbahnbau und -betrieb, Universität Hannover, no. 62, Eurailpress, Hamburg, 2005, Ph.D. thesis.

[BJS10] M. S. Bazaraa, John J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, 4 ed., Wiley, 2010.

[Chv83] V. Chvátal, *Linear Programming*, W. H. Freeman and Company, New York, 1983.

[CRT90] J. D. Camm, A. S. Raturi, and S. Tsubakitani, *Cutting Big M Down to Size*, INTERFACES **20** (1990), 61–66.

[Dan63] G. Dantzig, *Linear Programming and Extensions*, Princeton University Press, 1963.

[DDS05] G. Desaulniers, J. Desrosiers, and M. M. Solomon (eds.), *Column Generation*, Springer, New York, 2005.

[DL05] J. Desrosiers and M. E. Lübbecke, *A Primer in Column Generation*, Column Generation (G. Desaulniers, J. Desrosiers, and M. M. Solomon, eds.), Springer, New York, 2005, pp. 1–32.

[FFV01] M. Fellendorf, M. Friedrich, and P. Vortisch, *Kopplung makroskopischer und mikroskopischer Verkehrsmodelle - ein Verfahren für die Integration von großräumiger Planung und Detailplanung*, Tagungsband der 18. Verkehrwissenschaftlichen Tage, TU Dresden, 2001, pp. 632–657.

[FH90] K. Fischer and G. Hertel, *Bedienungsprozesse im Transportwesen*, transpress - Verlagsgesellschaft mbH, Berlin, 1990.

[Gas87] I. Gast, *Wartezeiten infolge stehender Überholungen auf Eisenbahnstrecken*, Veröffentlichungen des Verkehrswissenschaftlichen Institutes der Rheinisch-Westfälischen Technischen Hochschule Aachen, no. 39, Verkehrswissenschaftliches Institut der RWTH Aachen, 1987, Ph.D. thesis.

[GCF98] B. Gendron, T. G. Crainic, and A. Frangioni, *Multicommodity Capacitated Network Design*, Telecommunications Network Planning (B. Sansó and P. Soriano, eds.), Kluwer Academic Publishers, Norwell, 1998, pp. 1–19.

[Gur12] Gurobi, *Gurobi Optimization*, February 2012, `http://www.gurobi.com/`.

[Hap59] O. Happel, *Sperrzeiten als Grundlage für die Fahrplankonstruktion*, ETR **8** (1959), no. 2, 79–90.

[Hop69] L. Hoppner, *The Law of Travel and its Application to Rail Traffic*, Harvard Papers in Theoretical Geography: Geography and Properties of Surfaces, no. 25, Laboratory for Computer Graphics and Spatial Analysis, Harvard Center for Environmental Design Studies, the Graduate School of Design, Harvard University, 1969, Translated from E. Lill: *Das Reisegesetz und seine Anwendung auf den Eisenbahnverkehr*, 1891.

[HP08] I. A. Hansen and J. Pachl (eds.), *Railway Timetable & Traffic*, Eurailpress, 2008.

[Hro01] J. Hromkovič, *Algorithmics for Hard Problems*, Springer, 2001.

[ID05] S. Irnich and G. Desaulniers, *Shortest Path Problems with Resource Constraints*, Column Generation (G. Desaulniers, J. Desrosiers, and M. M. Solomon, eds.), Springer, New York, 2005, pp. 33–65.

[Kar84] N. Karmarkar, *A New Polynomial-Time Algorithm for Linear Programming*, Combinatorica **4** (1984), 302–311.

[Kha79] L.G. Khachiyan, *A Polynomial Algorithm in Linear Programming*, Doklady Akademii Nauk **244** (1979).

[KM10a] A. M. C. A. Koster and X. Muñoz (eds.), *Graphs and Algorithms in Communication Networks*, Springer, Berlin Heidelberg, 2010.

[KM10b] A. M. C. A. Koster and X. Muñoz, *Graphs and Algorithms in Communication Networks on Seven League Boots*, Graphs and Algorithms in Communication Networks (A. M. C. A. Koster and X. Muñoz, eds.), Springer, Berlin Heidelberg, 2010, pp. 1–56.

[KRS11] T. Koch, T. Ralphs, and Y. Shinano, *What could a million CPUs do to solve Integer Programs?*, Tech. Report 11-40, ZIB, 2011.

[KSH04] M. Kettner, B. Sewcyk, and J. Hunold, *Verknüpfung von mikroskopischen und makroskopischen Simulationsmodellen des Schienenverkehrs*, Der Eisenbahningenieur (2004), no. 2.

[LD60] A. H. Land and A. G Doig, *An Automatic Method of Solving Discrete Programming Problems*, Econometrica **28** (1960), no. 3, 497–520.

[Lei84] K. Leibbrand, *Die Weiterentwicklung der Verkehrsmodelle*, Verkehrsannalen - Mitteilungen der Österreichischen Verkehrswissenschaftlichen Gesellschaft **1** (1984), 14–20.

[Lil91] E. Lill, *Das Reisegesetz und seine Anwendung auf den Eisenbahnverkehr*, Spielhagen & Schurich, 1891.

[Lüb10] M. E. Lübbecke, *Column Generation*, Wiley Encyclopedia of Operations Research and Management Science, Wiley, 2010.

[Mar10] F. Margot, *Symmetry in Integer Linear Programming*, 50 Years of Integer Programming 1958–2008 (M. Jünger, T. Liebling, D. Naddef, G. Nemhauser, W. Pulleyblank, G. Reinelt, G. Rinaldi, and L. Wolsey, eds.), Springer, 2010, pp. 647–681.

[Min89] M. Minoux, *Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications*, Networks **19** (1989), no. 3, 313–360.

[MW84] T. L. Magnanti and R. T. Wong, *Network Design and Transportation Planning: Models and Algorithms*, Transportation Science **18** (1984), no. 1.

[Nie08] N. Nießen, *Leistungskenngrößen für Gesamtfahrstraßenknoten*, Ph.D. thesis, RWTH Aachen University, Aachen, 2008.

[NW03] N. Nießen and E. Wendler, *Weiterentwicklung der belastungsabhängigen Wartezeitfunktion der Strecken in WIZUG*, Verkehrswissenschaftliches Institut der RWTH Aachen, October 2003.

[NW05] N. Nießen and E. Wendler, *Infrastrukturmodellierung für die strategische Netzplanung*, Tagungsband der 20. Verkehrwissenschaftlichen Tage, TU Dresden, 2005.

[NW07] N. Nießen and E. Wendler, *Grenzleistung von Gesamtfahrstraßenknoten*, Tagungsband der 21. Verkehrwissenschaftlichen Tage, TU Dresden, 2007.

[ORK+10] S. Orlowski, C. Raack, A. M. C. A. Koster, G. Baier, T. Engel, and P. Belotti, *Branch-and-Cut Techniques for Solving Realistic Two-Layer Network Design Problems*, Graphs and Algorithms in Communication Networks (A. M. C. A. Koster and X. Muñoz, eds.), Springer, Berlin Heidelberg, 2010, pp. 95–116.

[Pac08] J. Pachl, *Timetable design principles (Chapter 2)*, Railway Timetable & Traffic (I. A. Hansen and J. Pachl, eds.), Eurailpress, Hamburg, 2008, pp. 9–42.

[PM04] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, Morgan Kaufmann, 2004.

[Pot70] G. Potthoff, *Verkehrsströmungslehre*, vol. 1, Transpress - VEB Verlag für Verkehrswesen, Berlin, 1970.

[Rad08] A. Radtke, *Infrastructure Modelling (Chapter 3)*, Railway Timetable & Traffic (I. A. Hansen and J. Pachl, eds.), Eurailpress, Hamburg, 2008, pp. 43–57.

[Ros01] S. Ross, *Strategische Infrastrukturplanung im Schienenverkehr*, Deutscher Universitäts-Verlag, Wiesbaden, 2001.

[Sau10] D. Sauss, *Die Abbildung von Infrastrukturkosten im Modellansatz zur Ermittlung der wirtschaftlich optimalen Zugzahl von Eisenbahnstrecken am Beispiel der Strecke Aachen - Köln*, Diploma thesis, RWTH Aachen University, 2010.

[SCI12] SCIP, *Solving constraint integer programs*, February 2012, `http://scip.zib.de/`.

[Sew04] B. Sewcyk, *Makroskopische Abbildung des Eisenbahnbetriebs in Modellen zur langfristigen Infrastrukturplanung*, Wissenschaftliche Arbeiten des Instituts für Verkehrswesen, Eisenbahnbau und -betrieb, Universität Hannover, no. 61, Eurailpress, Hamburg, 2004, Ph.D. thesis.

[SRW07] B. Sewcyk, A. Radtke, and G. Wilfinger, *Combining Microscopic and Macroscopic Infrastructure Planning Models*, 2nd International Seminar on Railway Operations Research (ISROR), Leibniz Universität Hannover, 2007.

[Sta08] H. Stange, *Optimierte Anordnung der Infrastrukturelemente von Eisenbahnstrecken*, Wissenschaftliche Arbeiten des Instituts für Verkehrswesen, Eisenbahnbau und -betrieb, Universität Hannover, no. 70, Eurailpress, Hamburg, 2008, Ph.D. thesis.

[Tar71] R. Tarjan, *Depth-first search and linear graph algorithms*, Switching and Automata Theory, 1971., 12th Annual Symposium on, 1971, pp. 114–121.

[TV09] G. Theeg and S. Vlasenko (eds.), *Railway Signalling & Interlocking*, Eurailpress, 2009.

[UIC04] UIC Code 406 - Capacity, *International Union of Railways*, 2004, `http://www.uic.org/`.

[vHKvdLS03] S.P.M van Hoesel, A.M.C.A Koster, R.L.M.J van de Leensel, and M.W.P Savelsbergh, *Bidirected and unidirected capacity installation in telecommunication networks*, Discrete Applied Mathematics **133** (2003), no. 1-3, 103–121.

[Wak85] H. Wakob, *Ableitung eines generellen Wartemodells zu Ermittlung der planmäßigen Wartezeiten im Eisenbahnbetrieb unter besonderer Berücksichtigung der Aspekte der Leistungsfähigkeit und der Anlagenbelastung*, Veröffentlichungen des Verkehrswissenschaftlichen Institutes der Rheinisch-Westfälischen Technischen Hochschule Aachen, no. 36, Verkehrswissenschaftliches Institut der RWTH Aachen, 1985, Ph.D. thesis.

[Wen99] E. Wendler, *Analytische Berechnung der planmässigen Wartezeiten bei asynchroner Fahrplankonstruktion*, Veröffentlichungen des Verkehrswissenschaftlichen Institutes der Rheinisch-Westfälischen Technischen Hochschule Aachen, no. 55, Verkehrswissenschaftliches Institut der RWTH Aachen, 1999, Ph.D. thesis.

[Wen02] E. Wendler, *Bemessungsmethoden für große Eisenbahnknoten*, Eisenbahntechnische Rundschau (2002), no. 7-8, 418–424.

[Wen07] E. Wendler, *The scheduled waiting time on railway lines*, Transportation Research Part B **41** (2007), no. 2, 148–158.

[Wen08] E. Wendler, *Queuing (Chapter 6)*, Railway Timetable & Traffic (I. A. Hansen and J. Pachl, eds.), Eurailpress, Hamburg, 2008, pp. 106–117.

[Wie06] T. Wieczorek, *Inverse Kapazitätsbestimmung für die Eisenbahn-Infrastruktur*, Wissenschaftliche Arbeiten im Eisenbahnwesen Universität Karlsruhe (TH), no. 1, Eurailpress, Hamburg, 2006, Ph.D. thesis.

[Wol98] L. A. Wolsey, *Integer programming*, Wiley-Interscience, New York, 1998.

A Blocking Time Theory

A.1 Blocking Time

Using the example of a block section, the blocking time denotes that amount of time during which a train which uses the block section for operational purposes exclusively occupies the block section. Due to the railway safety system and operational purposes, the blocking time consists of several elements besides the pure running time in the block section. This is shown in Figure A.1. A list of variables used in Figure A.1 is given below. Detailed explanations concerning each element are given by Pachl [Pac08].

$$\begin{array}{rl} l_{\mathrm{ps}} & \text{presignalling distance [m]} \\ l_{\mathrm{bk}} & \text{length of a block section [m]} \\ l_{\mathrm{so}} & \text{length of safety overlap [m]} \\ l_{\mathrm{tr},t} & \text{length of a train of type } t \text{ [m]} \\ l_{\mathrm{cl}} & \text{length of clearing section [m]} \end{array} \tag{A.1}$$

A.2 Minimum Headway Time

The minimum headway time of two trains—competing for the same infrastructure—is that amount of time it takes before the succeeding train is allowed to start occupying the infrastructure. If the corresponding blocking time stairways of such trains, are shifted in time such that there is, on the hand, no overlapping of blocking times and, on the other, at least one block with touching blocking times; the minimum headway time can be read out as distance between the start of each blocking time in the block section. The block section with touching blocking times is referred to as *relevant* block section. Whereas

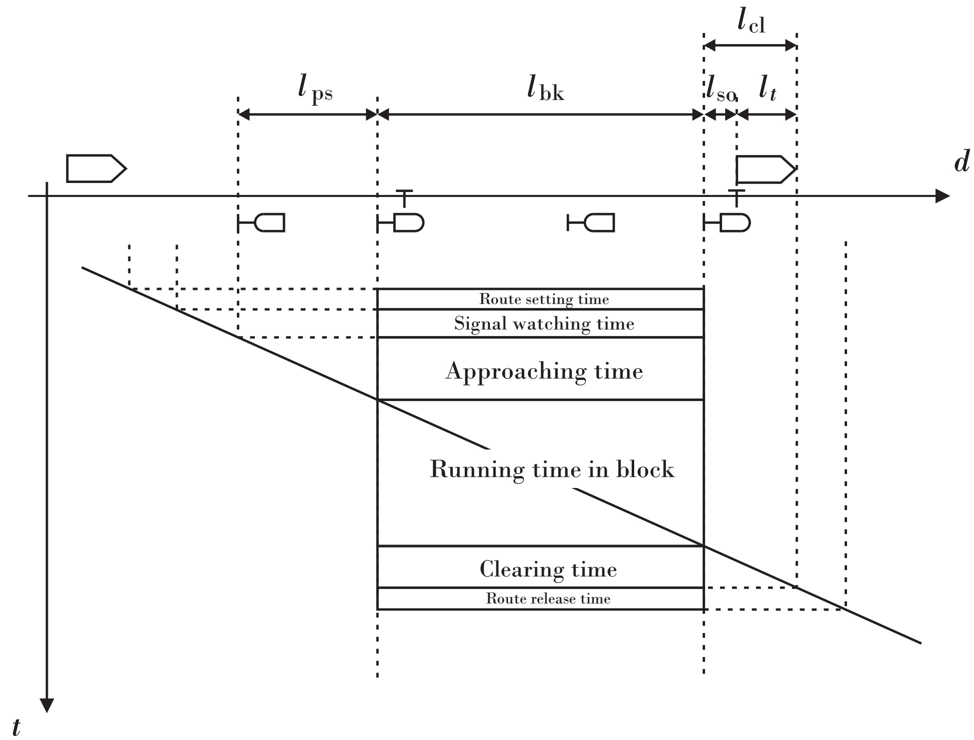

Figure A.1: Blocking time elements.

calculation of headway times requires consideration of pairs of trains, two cases must be differentiated. The distinction (and later on the estimation of minimum headway times) is made according to Ross [Ros01]. He distinguishes three cases. For the current purpose it suffices to consider two cases which differ from each other by the position of the relevant block section. Let t, u two different train types and w. l. o. g. $v_t > v_u$. For train j following train i this yields to following two cases:

i) **Train i is of type t and train j is of type u or type t.** In this case the minimum headway time simply is determined by the blocking time of train i in the first section, which is the relevant block section. This is shown in Figure A.2. The corresponding minimum headway time is calculated as follows:

$$z_{tu} = t_{\mathrm{rs}} + t_{\mathrm{s}} + \underbrace{0.06\frac{l_{\mathrm{ps}}}{v_t}}_{t_{\mathrm{ap},i}} + \underbrace{0.06\frac{l_{\mathrm{bk}}}{v_t}}_{t_{\mathrm{r},i}} + \underbrace{0.06\frac{l_{\mathrm{so}} + l_{\mathrm{tr},t}}{v_t}}_{t_{\mathrm{cl}}} + t_{\mathrm{rr}} \qquad \text{[min]} \tag{A.2}$$

t_{rs}	route setting time	$t_{\mathrm{r},i}$	runtime of train i in the first block section
t_{s}	signal watching time	t_{cl}	clearing time
$t_{\mathrm{ap},i}$	approaching time of train i	t_{rr}	route release time

ii) **Train i is of type u and train j is of type t.** Here, the relevant block section is the last block section, see Figure A.3. The last block section is not inevitable the relevant one but it holds for this work since block sections' length and trains' speed are assumed to be constant.

$$z_{ut} = t_{\mathrm{rs}} + \underbrace{0.06\frac{l_{\mathrm{ps}}}{v_u}}_{t_{\mathrm{ap},i}} + t_{\mathrm{s}} + \underbrace{0.06\frac{l_{\mathrm{os}} - l_{\mathrm{st}}}{v_u}}_{t_{\mathrm{r},i}} + t_{\mathrm{cl}} - \underbrace{0.06\frac{l_{\mathrm{os}} - l_{\mathrm{st}} - l_{\mathrm{bk}}}{v_t}}_{t_{\mathrm{r},j}} + t_{\mathrm{rr}} \qquad \text{(A.3)}$$

$t_{\mathrm{r},i}$	runtime of train i in the common track sections
$t_{\mathrm{r},j}$	runtime of train j in common track sections, except of the last block section

The clearing time t_{cl} reflects the fact that the slower train, which is passed by the faster train, starts breaking in the last part of their common track sections to stop at the stopping place of the passing track. The last (and relevant) common element of infrastructure in this context is the first switch of the (overtaking) station. It has to be cleared before the succeeding train is allowed to occupy the last block section before the station block. Calculation of t_{cl}, which is shown in the following equation, requires a case distinction to distinguish between different speeds and deceleration rates of the preceding train, which influence the position of the brake application point. This is expressed by the value x which is the length of that part of the station block which is used with constant/linear speed. A value smaller than zero indicates that the train has to start braking before it reaches the entry signal. Besides that, it may happen that the train has to reduce its speed not only to stop at the stopping place of passing track but also because of the speed limit of the switch which has to be used to enter the passing track. For the sake of simplicity this is not considered here.

$$t_{\mathrm{cl}} := \begin{cases} \frac{v_u}{216 d_u} - \sqrt{\frac{v_u^2}{216^2 d_u^2} - \frac{2(l_{\mathrm{so,st}} + l_{\mathrm{tr},u})}{60^2 d_u}}, & x < 0 \\ \underbrace{0.06 \dfrac{x}{v_u}}_{\text{constant speed}} + \underbrace{\dfrac{v_u}{216 d_u} - \sqrt{\dfrac{v_u^2}{216^2 d_u^2} - \dfrac{2(l_{\mathrm{so,st}} + l_{\mathrm{tr},u} - x)}{60^2 d_u}}}_{\text{braking}}, & 0 \leqslant x \leqslant l_{\mathrm{so,st}} + l_{\mathrm{tr},u} \\ 0.06 \frac{l_{\mathrm{so,st}} + l_{\mathrm{tr},u}}{v_u}, & x > l_{\mathrm{so,st}} + l_{\mathrm{tr},u} \end{cases}$$

$$x := l_{\mathrm{st}} - \frac{v_u^2}{3.6^2 \cdot 2 \cdot d_u}$$

Variables v_t, d_t, and $l_{\mathrm{tr},t}$ model speed, deceleration rate and length of a train of type t, as defined in Definition 2.6. Conversion factor 0.06 is needed to convert from unit $\left[\frac{\mathrm{m}}{\mathrm{km/h}}\right]$ to unit [min], $\frac{1}{216}$ converts $\left[\frac{\mathrm{km/h}}{\mathrm{m/s^2}}\right]$ to [min].

The following values are assumed to be constant. This sufficient for a reasonable estimation of the minimum headway times.

length of a station block	$l_{\mathrm{st}} = 1{,}500$ m (cf. (2.4))
length of block section	$l_{\mathrm{bk}} = 2{,}000$ m (cf. (2.5))
length of safety overlap	$l_{\mathrm{so}} = 200$ m
length of safety overlap in stations	$l_{\mathrm{so,st}} = 250$ m
presignalling distance	$l_{\mathrm{ps}} = 1{,}000$ m

Furthermore, following assumptions are made:

- Presence of a train control system[1] with intermittent data transmission, e.g., the German system *PZB90* which includes a braking supervision. In this regard, the signal watching time is set to the following value which is reasonable in relation to practice.

  signal watching time $t_{\mathrm{s}} = 0.2$ min

- Presence of electronic interlockings. This has an influence on route setting and

[1]For information about train control systems and railway signalling and interlocking see [TV09].

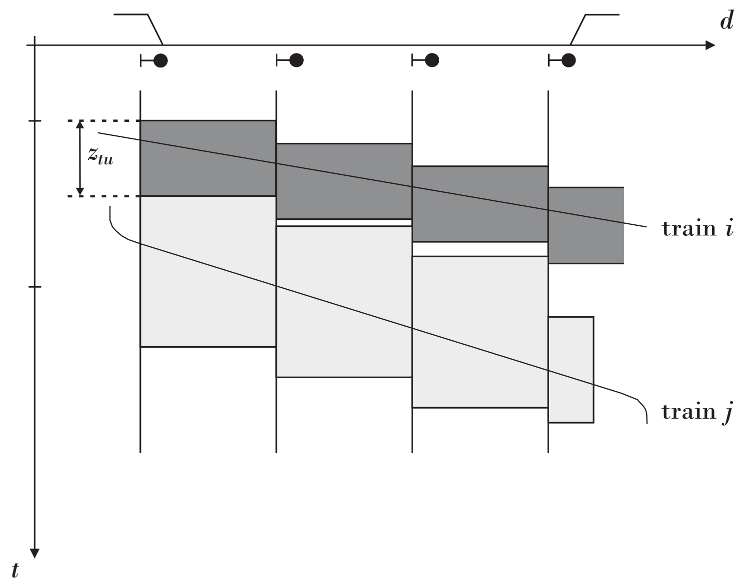

Figure A.2: Minimum headway time z_{tu}. Train i is of type t, train j of type u, with $v_t > v_u$. First block section is the relevant block section.

route release times. Mechanical interlockings would lead to a prolongation of such times. The values given below are reasonable in relation to practice.

route setting time	$t_{\text{rs}} = 0.15$ min
route release time	$t_{\text{rr}} = 0.05$ min

Example

The example deals with case ii) pictured in Figure A.3. Let t and u be two train types with $Param_t := \{200, 0.5, 400\}$ and $Param_u := \{80, 0.3, 750\}$, $l_{\text{os}} := 10{,}000$ m, train i of type u, and train j of type t. Then the minimum headway time z_{ij} is calculated as follows:

$$l_{\text{so,st}} + l_{\text{tr},u} = 250 + 750 = 1{,}000 \text{ m}$$

$$x = l_{\text{st}} - \frac{v_u^2}{3.6^2 \cdot 2 \cdot d_u} = 1{,}500 - \frac{80^2}{3.6^2 \cdot 2 \cdot 0.3} = 676.955 \text{ m}$$

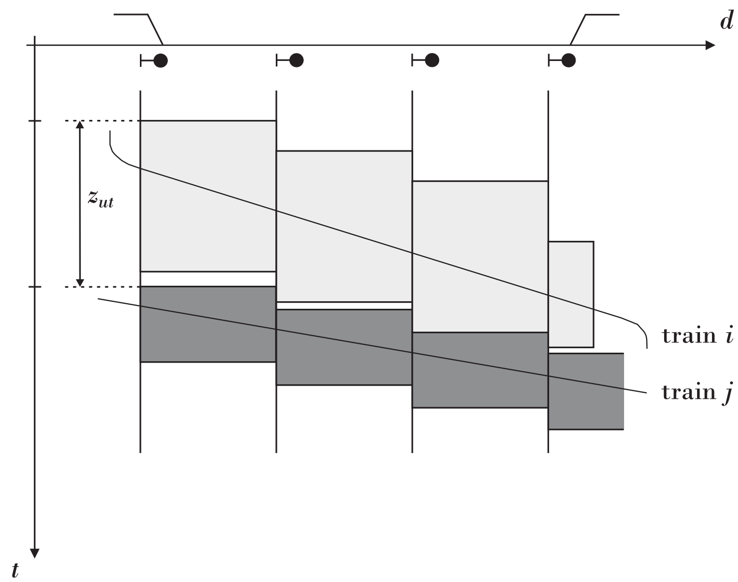

Figure A.3: Minimum headway time z_{ut}. Train i is of type u, train j of type t, with $v_t > v_u$. Last block section is the relevant block section.

$$\begin{aligned}
\overset{0\leqslant x\leqslant 1{,}000}{\Rightarrow} t_{\text{cl}} &= 0.06\frac{x}{v_u} + \frac{v_u}{216 d_u} - \sqrt{\frac{v_u^2}{216^2 d_u^2} - \frac{2(l_{\text{so,st}} + l_{\text{tr},u} - x)}{60^2 d_u}} \\
&= 0.06\frac{676.955}{80} + \frac{80}{216\cdot 0.3} - \sqrt{\frac{80^2}{216^2\cdot 0.3^2} - \frac{2(250+750-676.955)}{60^2\cdot 0.3}} \\
&= 0.780 \text{ min}
\end{aligned}$$

$$\begin{aligned}
z_{ij} &= t_{\text{rs}} + 0.06\frac{l_{\text{ps}}}{v_u} + t_{\text{s}} + 0.06\frac{l_{\text{os}} - l_{\text{st}}}{v_u} + t_{\text{cl}} - 0.06\frac{l_{\text{os}} - l_{\text{st}} - l_{\text{bk}}}{v_t} + t_{\text{rr}} \\
&= 0.15 + 0.06\frac{1{,}000}{80} + 0.2 + 0.06\frac{10{,}000 - 1{,}500}{80} + 0.78 - 0.06\frac{10{,}000 - 1{,}500 - 2{,}000}{200} + 0.05 \\
&= 0.15 + 0.75 + 0.2 + 6.375 + 0.78 - 1.95 + 0.05 \\
&= 6.355 \text{ min}
\end{aligned}$$

Fast train j has to wait 6.355 min to enter the first block section of the track sections slow train i occupied before.

B Life-Cycle Cost

As described in Section 2.2.2 costs $cost_{ij}$ of an arc $(i,j) \in A$ (of a network of railway infrastructure $G = (N, A)$) are defined by life-cycle costs of the corresponding railway line, cf. (2.14):

$$cost_{ij}(l_{\mathrm{l},ij}, l_{\mathrm{bk}}, n_{\mathrm{tk},ij}, n_{\mathrm{os},ij}) := n_{\mathrm{tk},ij} \left(LCC_{\mathrm{def},ij}(l_{\mathrm{l},ij}, l_{\mathrm{bk}}) + n_{\mathrm{os},ij} \cdot LCC_{\mathrm{pt}}\right).$$

Whereas Section 2.2.2 is dealing with general information and points out which parameters of the infrastructure model are used to reasonably estimate line life-cycle cost, the following sections are focussing on the derivation of function $LCC_{\mathrm{def},ij}$ and constant LCC_{pt} as well as on a detailed sample calculation.

B.1 Line and Passing Track

Let $G = (N, A)$ be a network of railway infrastructure and $(i,j) \in A$ an arc. Minimum life-cycle costs of the railway line without overtaking stations, i.e. $l_{l,ij} = l_{os,ij}$, corresponding to (i,j) consist of two components: the life-cycle costs of the track, depending on its length, and the signalling needed to obtain block sections, see Figure B.1.

$$LCC_{\mathrm{def},ij}(l_{\mathrm{l},ij}, l_{\mathrm{bk}}) := l_{\mathrm{l},ij} \cdot LCC_{\mathrm{track}} + \frac{l_{\mathrm{l},ij}}{l_{\mathrm{bk}}} \cdot LCC_{\mathrm{signal}}, \tag{B.1}$$

where LCC_{track} denotes the life-cycle costs per meter track and LCC_{signal} the life-cycle costs of a signalling unit a block section has to be equipped with. Both constants are defined later on. The number of required signalling units is, like l_{bk}, an average value which may be a fractional number.

Each new overtaking station adds 1,000 meters of track for the passing track. This length of the passing track is selected in such a way that cargo trains with length of 750

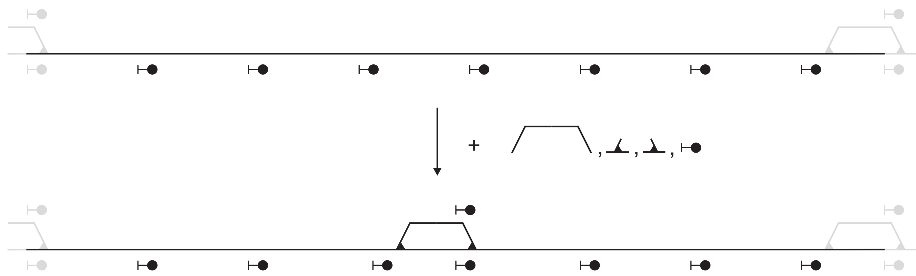

Figure B.1: Line with and without overtaking station. Colored black: elements which are relevant to life-cycle cost calculation.

meters[1] fit in the passing track and so can be passed by other trains. In addition to the track, two switches connecting passing track and main track and a depart signal at the end of the passing track are added. Entry and depart signal of the main track in the overtaking station are assumed to be re-used block signals. Additional infrastructure elements of one newly added overtaking station are displayed in Figure B.1. The corresponding constant life-cycle costs can determined according to (B.2).

$$LCC_{\text{pt}} := 1{,}000 \cdot LCC_{\text{track}} + 2 \cdot LCC_{\text{switch}} + 1 \cdot LCC_{\text{signal}}, \tag{B.2}$$

B.2 Track, Signalling & Switches

A track consists of the elements *rail*, *concrete sleeper* and *ballast*. The only type of switch chosen here is a widely built in one with a radius of 760 meters: *UIC 60-760* on concrete sleepers. For the sake of simplicity, signalling is simply modeled by *main signals* where block signals, depart signals or entry signals are not differentiated. Distant signals and other interlocking equipment are not taken into account. Equations (B.3)–(B.5) show the calculation of elements LCC_{track}, LCC_{signal}, and LCC_{switch} of line life-cycle cost. Required input data, displayed in Table B.1, is taken from Sauss [Sau10].

[1] 750 meters are the upper bound on the length of cargo trains operating in Germany. However, the use of trains with lengths up to 1,000 meters have been tested and may become a standard on selected routes.

element	**unit**	**acquisition** [€]	**lifespan** [a]	**maintenance** [€/a]
UIC60 rail	m	190.71	25	2.24
concrete sleeper	m	190.81	50	1.19
ballast	m	108.22	50	0.49
main signal	piece	105,732.8	37.5	528.66
single UIC60-760	piece	246,166.03	35	5,527.56

Table B.1: LCC properties of infrastructure elements.

$$
\begin{aligned}
LCC_{\text{track}} &:= \underbrace{\frac{AC_{\text{rail}}}{L_{\text{rail}}} + MC_{\text{rail}}}_{LCC_{\text{rail}}} + \underbrace{\frac{AC_{\text{sleeper}}}{L_{\text{sleeper}}} + MC_{\text{sleeper}}}_{LCC_{\text{sleeper}}} + \underbrace{\frac{AC_{\text{ballast}}}{L_{\text{ballast}}} + MC_{\text{ballast}}}_{LCC_{\text{ballast}}} && \text{(B.3)}\\
&= \frac{190.71}{25} + 2.2 + \frac{190.81}{50} + 1.19 + \frac{108.22}{50} + 0.49\\
&= 9.8684 + 5.0062 + 2.6544\\
&= 17.529\ \frac{€}{\text{m} \cdot \text{a}},\\
LCC_{\text{signal}} &:= \frac{AC_{\text{signal}}}{L_{\text{signal}}} + MC_{\text{signal}} && \text{(B.4)}\\
&= \frac{105{,}732.8}{37.5} + 528.66\\
&= 3{,}348.201\ \frac{€}{\text{a}},\\
LCC_{\text{switch}} &:= \frac{AC_{\text{switch}}}{L_{\text{switch}}} + MC_{\text{switch}} && \text{(B.5)}\\
&= \frac{246{,}166.03}{35} + 5{,}527.56\\
&= 12{,}560.875\ \frac{€}{\text{a}}.
\end{aligned}
$$

B.3 Sample Calculation

Let $G = (N, A)$ a network of railway infrastructure and $(i, j) \in A$ an arc with $l_{\mathrm{l},ij} =$ 20,000 m, $l_{\mathrm{os},ij} = 5{,}000$ m $\Rightarrow n_{\mathrm{os},ij} = \frac{20{,}000}{5{,}000} - 1 = 3$, $n_{\mathrm{tk},ij} = 1$ and $l_{\mathrm{bk}} = 2{,}000$ m.

$$\begin{aligned}
cost_{ij}(20{,}000, 2{,}000, 1, 3) &= 1\,(LCC_{\mathrm{def},ij}(20{,}000, 2{,}000) + 3 \cdot LCC_{\mathrm{pt}}) \\
&= 20{,}000 \cdot LCC_{\mathrm{track}} + 10 \cdot LCC_{\mathrm{signal}} \\
&\quad + 3 \cdot (1{,}000 \cdot LCC_{\mathrm{track}} + 2 \cdot LCC_{\mathrm{switch}} + 1 \cdot LCC_{\mathrm{signal}}) \\
&= 522{,}058.863 \text{ €}/a
\end{aligned}$$