Vom Fachbereich für Mathematik und Informatik
der Technischen Universität Braunschweig
genehmigte Dissertation
zur Erlangung eines Doktors der Naturwissenschaften (Dr.rer.nat.)

Michael Bussieck

Optimal Lines in Public Rail Transport

21. Dezember 1998

1. Referent: Prof. Dr. Uwe T. Zimmermann
2. Referent: Prof. Dr. Robert E. Bixby, Prof. Dr. Michael L. Dowling
eingereicht am: 9. September 1998

Meinen Eltern
gewidmet

Acknowledgements

In 1994 the Federal Ministry for Education, Science, Research and Technology (BMBF) in Germany started the program *Application Oriented Joint Projects in Mathematics*. People from industry and research work together in about 70 projects on real life problems. I had the great pleasure of being involved in one of these projects, namely, the project *Optimal Line- and Routeplanning in Traffic Systems (Railroad Traffic)*. The outstanding atmosphere of this research program by means of scientific and commercial experience of the participants and a financial equipment provided an environment that made this kind of applied *and* scientific work possible.

I am grateful to the industrial partners for providing me with real-world data and support. I am especially thankful to SMA und Partner AG, Leo Kroon (Nederlandse Spoorwegen) and Matthias Krista (Adtranz). Matthias is an inexhaustible source for railroad related information which helped me to overcome many obstacles in chapter 2 and 3.

I would like to thank Peter Kreuzer who set the ball "*line optimization*" rolling. His diploma thesis [42] formed the basis of the research proposal submitted to the BMBF as well as the fundamentals of the direct traveler approach (chapter 5).

For the comparison of different MIP solvers (cf. section 5.5) I required access to several commercial software products. I would like to thank Robert Bixby (CPLEX) for solving the huge linear programs arising from the (LOP) formulation in his *garage*. Furthermore I am grateful to the license team of Dash Associates (XPRESS), Franz Nelissen (GAMS), Uwe Suhl (MOPS), Martin Savelsbergh (MINTO), and Knut Haase who provided me with a user account on his machine running OSL.

I had the pleasure of working with my colleague Thomas Lindner who was intensely involved in the cost optimal line planning (chapter 6). He developed a passion for solving the `sp97ic` instance.

The working group of Professor Uwe Zimmermann provided an outstanding environment for my work. I am grateful to him for supporting my work in many aspects. I give my gratitude to my colleagues who never got tired of asking, "Isn't it done yet?". Moreover, I would like to thank our secretary Heidemarie Pförtner who firstly persuaded me to take on the BMBF-position.

I have restricted to mention people from academia and commerce, although there are many others, whose support was at least as important. I am especially very thankful to my parents Margrete and Bruno Bussieck and to my wife Susanne for her patience, encouragement, and love.

Braunschweig, September 1998 *Michael Bussieck*

Contents

Chapter 1

Preface

This thesis deals with the *line planning* problem for public transportation networks based on *periodic schedules.* The models and algorithms represented in this thesis take care of peculiarities of public *rail* transport. However, the ideas mentioned in this monograph can be easily adapted to the line planning problem for other transportation systems with periodic schedules, e.g. *busses.*

A comprehensive discussion of the line planning problem including its modeling and solution applying *mathematical programming* methods, constitutes the core of this thesis. Beyond the practical aspects we concentrate on structural properties of the problems. For instance, we prove that the line planning problem belongs to the class of the hardest optimization problems. For a particular line planning problem we analyze the polyhedral structure of the corresponding integer linear program. These investigations represent the theoretical background of the methods we apply to the different models. Moreover, they prove why certain techniques improve the solution of the models by means of shorter computation time of the corresponding algorithms.

The theoretical characteristics without the peculiarities of the practical problem being under consideration permit a tractable adaptation of models and methods to related problems. This level of problem abstraction which is in fact higher than in most engineering sciences, provides the construction of *optimal* or *provably good solutions.* This type of solution quality can be frequently transfered to the real life problem. Particularly, in *strategic planning* with a planning period of 10 – 20 years this approach becomes most important. The most commonly used way of comparing the new and the current solution is inaccessible at this point of planning.

These advantages of a mathematical approach to practical problems face certain difficulties outside mathematics. Practitioners are sceptical about *non-intuitive* methods and use a complete different language than mathematicians. The Federal Ministry for Education, Science, Research and Technology in Germany started the program *Application Oriented Joint Projects in Mathematics* in 1994 to overcome these obstacles. People from industry and research work together in about 70 projects on real life problems. The project *Optimal Line- and Routeplanning in Traffic Systems (Railroad Traffic)* which forms the fundamentals of this thesis, was carried out together with engineers from *Adtranz Signal GmbH*. Former academic research on real life problems consists of constructing models and algorithms and *proving* their efficiency by testing small random problem instances. In contrast to that, the results elaborated in these projects, must stand the test of (large scale) real life instances. In case of the line planning problem the input data consists of information about the *infrastructure* (including the network topology) and the *customers* (usually given by an *origin-destination matrix*). A proper set of real life data instances is difficult to get hold of. The data, particularly the origin-destination matrix, represents a trade secret. The utilization of data is laid down by certain contracts granting uses which include that publication of data is done by way of example or accumulation only. Consequently, this thesis has to take these regulations into account. We have collected data instances from three federal railroad companies: *Deutsche Bahn AG*, *Nederlandse Spoorwegen*, *Schweizerische Bundesbahnen*. Furthermore, we dispose of instances from two local public transport companies: *Braunschweiger Verkehrs AG* and *Verkehrsbetriebe der Stadt Zürich*.

Traditional mathematical techniques for practical problems, if available, provide an unsatisfactory performance when applied to large scale, real life instances. The improvement of present methods or a new development of models that cope with these instances is a challenge for the mathematical community. Progress in computer technology and in design of efficient algorithms and their implementation together with mathematical advance lead to satisfactory results in some cases. For example, at the beginning of the project we could not solve an *integer linear program* for a particular line planning problem applied to the German InterRegio network. Even after 24 hours of computation time on an HP9000/720-50 with the mixed integer linear programming solver CPLEX 2.0 we obtained a feasible solution with an inacceptable performance guarantee of approximately 85%. Currently, we can solve an improved model on an HP C180 with CPLEX 5.0 in less than one minute (to optimality). The revision of this model permits the use of poorer solvers including software in the public domain, for small and medium sized real world instances.

The models for the line planning problem considered in this monograph are integer linear programs. In most cases, an integer linear program resulting from a pure problem formulation can not be solved in spite of massive computer power. The model improvements, which lead to fast solution times, are based on techniques of *polyhedral optimization*. We derive *valid inequalities* or *cuts* and apply various *preprocessing* techniques to eliminate variables and constraints. In the last decade, these techniques which provide a *tighter* linear description of the polytope associated with the integer linear program, were applied to problems with combinatorial structure. Additionally, the success of these methods is due to improvements of the simplex algorithm and interior point methods for solving *linear programs*.

A lot of work has to be done to put the mathematical solution into practice, such that the project outcome assists the practitioners with their planning decisions. A user-friendly interface and an easy integration of the derived software in the present IT system make an optimization approach more acceptable. Nevertheless, the essential part is the quality of the proposed solution and consequently of model and method. The development of mathematical techniques complying with these requirements is beyond *applied* mathematics and should be circumscribed by *practical* mathematics.

The thesis is organized as follows. In the next two chapters we introduce the line planning problem and the associated planning task in the context of (railroad) traffic planning. Chapter 4 comprises mathematical programming formulations for the *generic line planning problem*. In chapter 5 we discuss a particular approach to the line planning problem. This approach focuses on a line plan with a maximum number of *direct travelers*, that are passengers that need not change lines to travel from their origin to their destination. The objective that takes care about the number of direct travelers represents the service aspect of line planning from the customers point of view. In chapter 6 we present an alternative approach to the line planning problem. We present a cost-optimal line planning problem, introduced by CLAESSENS [22], and suggest a new model. This cost approach, which emphasizes the economical aspects of a transportation system, becomes more and more important, if we think of the privatization process of state-owned railroads. Finally, we draw some conclusions and give a prospect of future research on the line planning problem.

Chapter 2

Public rail transport planning

The process of *transport planning* deals with the determination of *routes* between an origin and a destination and the assignment of necessary *resources* with regard to the future. In the early days of railroad this process followed a stringent order. The stops of fast national and international links with melodious names like *Orient Express*, *Golden Arrow*, *Train Bleu*, and the *Trans-Siberian Railroad* were itself starting points of regional railroads and local trains connecting minor centers and district towns. A comprehensive and network-wide planning, not least because of political reasons did not take place.

A certain characteristic of this kind of transportation is that hardly more than two trains use the same way through the network, so we may justifiably talk of an *individual* transportation service. Numerous trains on the same route or *line* provide an improved service and furthermore simplify the planning process of the transportation company. If, in addition, the departure times of consecutive trains of the same line always give fixed time intervals, the so called *cycle time*, we talk of a transportation network based on a *train schedule* of departure times at regular intervals, or even simpler on a *periodic schedule*. In urban public transportation (bus and trams), traffic engineers take advantage of line-based periodic service since the beginning of the twentieth century. Due to the development of private transport and cuts in the budget of public transport companies, planners apply the efficient and comfortable concept of periodic schedules to long-distance railroad transportation. For example, the German railroad company (Deutsche Bahn) established a line-based InterCity service in 1971 which initiates further line-based systems in Germany and other densely populated European countries.

	travelers	traveler kilometers	personnel	length of network (km)
Deutsche Bahn AG	1.33 billion	60.51 billion	276957	41718
Nederlandse Spoorwegen	-	14.00 billion	25855	2795
Schweizerische Bundesbahnen	0.25 billion	11.66 billion	33000	3000

Table 2.1: Reference numbers of European railroad companies

The number of passengers in long-distance railroad transport, which has been doubled from 1980 to 1990 in several countries [82], is still increasing due to congested roads and modern high-speed trains like the ICE (Germany), the TGV (France, Belgium, South Korea, Spain), the X2000 (Sweden, Norway, Australia), the TAV (Italy) and the Shinkanses (Japan)[1].

Due to the tremendous size and complexity of such a system (cf. table 2.1)[2] researchers from transportation science constitute a hierarchical planning concept. Figure 2.1 depicts this fundamental scheme of public (rail) transport planning based on periodic schedules. Every demand-oriented transportation service has to be based on the *passenger demand* usually given by an *origin-destination matrix*. The subsequent task of *line planning* determines the lines, i.e. the stops, links, and the cycle time of the regular routes. Afterwards, in the *train schedule planning* all arrival- and departure times will be fixed with respect to the cycle time of the lines. This raw plan will be refined by including operational constraints and temporal variations. Every *trip* of the resulting train schedule requires an engine and several coaches which will be assigned in the planning of *rolling stock*. A similar task, the *crew management* takes over the distribution of personnel in order to equip each trip with the necessary staff, including conductors and engine drivers. Long-term planning, like network (re-)design, and the on-line aspect of the execution of the resulting schedules are excluded from this simplified scheme.

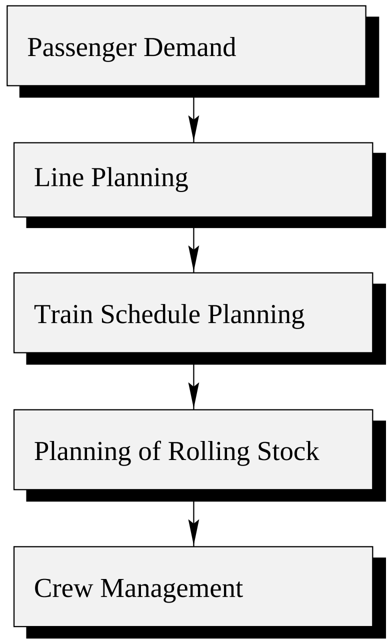


Figure 2.1: Hierarchical Planning Process

The disadvantages of this *top-down* approach are obvious, because the optimal output of a subtask which serves as the input of a subsequent task, will, in general, not result in an overall optimal solution. Nevertheless, this hierarchy decomposes the planning process in manageable segments and reflects the current internal structure of the railroad companies. Furthermore, it provides an integration into the classical temporal division consisting of *strategic*, *tactical*, and *operational* procedures introduced by ANTHONY [3]. Operational decisions reflect the *day-by-day* activities and the disturbances when executing the schedules. Tactical planning addresses *resource allocation* for the period from one to five years ahead. Strategic planning focuses on *resource acquisition* for the period from five to fifteen years ahead. The steps of the described planning process for the *production plans* occur at the tactical level, but certain tasks are also used for strategic planning. In long-term projects planners have to come to a decision on the

[1] Suitable Web resources can be found at URL `http://mercurio.iet.unipi.it/tgv/world.html`.

[2] All numbers collected from the Web sites `www.bahn.de`, `www.ns.nl`, and `www.sbb.ch`.

strength of the analysis of some configurations. In particular, the utilization of planning task at several levels will be perfectly clear for train schedule planning. For example, the foundation or extension of a railroad station requires a valuation concerning the management of future traffic demand. The *capacity*, determined by the topology of the station, must be sufficient for certain prospective train schedules [54] (strategic level). For a production train schedule it is necessary to include certain operational constraints which are neglected in the strategic decisions (tactical level). Due to unpredictable influences (breakdowns, special trains) the operating train schedule must be rearranged in real time in order to limit the changes which trigger irregularities to the schedule of rolling stock and personnel (operational level).

A combination of certain planning tasks, e.g. the integrated planning of rolling stock and personnel, which we encounter at the operational level for long-distance traffic planning only, is more customary for urban public transport. Due to the smaller size of the problem instances, a mutual planning of lines and train schedule [16] or circulation of rolling stock and personnel [30] is possible and provides gain in service as well as in operational cost. Currently, an application of such a combined planning seems to be impossible for larger instances. However, the process of privatization of public transportation companies which enforces the efficient utilization of resources to provide a high quality service, may initiate the application of optimization methods at certain levels.

The remaining part of this chapter represents a brief overview of recent mathematical programming approaches to the particular planning tasks. A more comprehensive survey of discrete optimization techniques in public transport planning can be found in [13, 14].

2.1 Passenger demand

The *volume of traffic* or the *passenger demand* must be given to establish a custom-oriented transportation service. The conventional form of the passenger demand is an *origin-destination matrix*. This matrix whose entries are estimated by sophisticated engineering models, is published e.g. in Germany in the *Bundesverkehrswegeplan*. This matrix is not classified according to various means of transport (car/train/airplane). Models for partitioning this matrix into origin-destination matrices for car/train/airplane transportation which is called the *modal split* (for a mathematical introduction cf. [64]) as well as models for estimating the complete matrix are outside the scope of this thesis.

Another approach which determines the passenger demand for a present transportation service, is based on *traffic census*. A number of cost-intensive interviews may serve as a basis for a statistical analysis which leads to an estimation of the overall demand. Alternatively and even simpler, the travelers on the edges of the transportation network (tracks, streets) can be counted. If we presume certain probable travel routes for *origin-destination pairs*, some statistical [17] and mathematical programming [7, 40, 68] methods for estimating origin-destination matrices from edge counts are available. Note that the resulting matrix reflects the passenger demand of the *current* transportation service. Certainly, the volume of traffic and consequently the estimated demand depends on this service, hence its application for future planning is questionable.

2.2 Line planning

The line planning problem which represents the core of this thesis, will be described in detail in the next chapter. Lines represent the fundamentals of transportation networks based on periodic schedules. A *line* consists of a *route* in the network and a *cycle time*. Suitable cycle times c for a fixed *period* or *basic time interval* $[0\ldots\tau)$ (e.g. $\tau = 60$ minutes) are those which lead to integer *frequencies* $f = \tau/c$. The line planning problem consists of choosing a set of operating lines that complies with the passenger demand and optimizes a given objective. Certain proposals for suitable objective functions will also be given in the next chapter.

2.3 Train schedule planning

The generation of a train schedule consists of fixing the departure- and arrival times for all trains at every station. According to the number of trains that pass a station and their cycle times given by the line plan, the train schedule should be designed to minimize the waiting time for passengers that must change trains. In order to prevent conflict situations when sharing resources (tracks, switches, platforms) the schedule has to take certain operational constraints into account, e.g. different velocities, acceleration, deceleration, and turn-around-times. Several mathematical models have been proposed for the train schedule problem (cf. [16] for a starting point of non-periodic train schedule planning). In a periodic train schedule the departure- and arrival times represent *periodic events*. SERAFINI and UKOVICH [67] introduce the related *periodic event scheduling problem* (PESP) which initiated several papers including applications to train scheduling. The problems of minimizing transfer times [49, 50] as well as satisfying all operational constraints [53, 75] are currently under consideration and may provide an integration of these separate questions in the near future.

The realization of departure- and arrival times according to the *regular* cycle times is only one part of train schedule generation. The second part, the domain of an experienced human planner, consists of adjusting the proposed regular train schedule to meet a bunch of local requirements (rush hours, splitting of lines, etc.) and other peculiarities.

2.4 Circulation of rolling stock and personnel

The trips established by the train schedule must be performed with some vehicles (motor unit, coaches) and a crew (conductors, kitchen staff, engine drivers). The overall cost of a transportation service is primarily based on the dispatch of these resources. Hence *optimal* assignments play a crucial part in *efficient* transportation systems. In the already mentioned federal program *Application Oriented Joint Projects in Mathematics* GRÖTSCHEL et. al. [37, 47] investigate the *vehicle scheduling problem* for local public transport systems. Although they succeed in solving real life problems from Berlin and Hamburg, the application of their models which rest on *integer multi-commodity flows*, is questionable due to larger circulation times of rolling stock in long-distance traffic. SCHRIJVER [66] discusses a model, also based on integer multi-commodity flows, for a small railroad network (four stations) and two different types of vehicles.

In the crew management the planner must dispatch railroad crews as well as local staff (cleaning staff, shunting gangs, staff at the ticket office). A bunch of constraints due to union contracts and operational restrictions increases the complexity of the problem and forces a decomposition into *crew rostering* and *crew scheduling*. The latter consists of the generation of *duties* which cover all jobs. Each duty is a sequence of tasks carried out by *one* crew. The crew rostering sequences the duties to final *rosters*. In this step the single tasks are not longer taken into account. Recently, CAPRARA et. al. [14] succeeded in solving crew management problems with exact algorithms as well as approximation methods based on *set covering* formulations, for the Italian railroad company.

Chapter 3

Line planning

The very first paper concerning the line planning problem, we are aware of, is the one by PATZ [61] in 1925. Since then, several papers about this problem especially for urban public transportation have been published. OLTROGGE [55] suggests a frame work for line planning based on a partitioning of this complex problem (cf. figure 3.1).

Public transportation does not form an integrated whole but is split into several *services* to meet the requirements of their customers. In particular, the different *means of transport* (tram/tube/bus) reflect the ordinary splitting of services in urban public transport. The aim of the *supply definition* is the decomposition of the global transportation network into several *supply networks* or *systems*. For urban public transportation networks this decomposition is forced by the physically disjoint networks. But also railroad companies offer different services (*InterCity*, *InterRegio*, local trains) whose trains concurrently use the tracks of the network. The resulting supply networks provide a logical decomposition of the global transportation network. The problem of finding a line plan can be independently performed on the different supply networks.

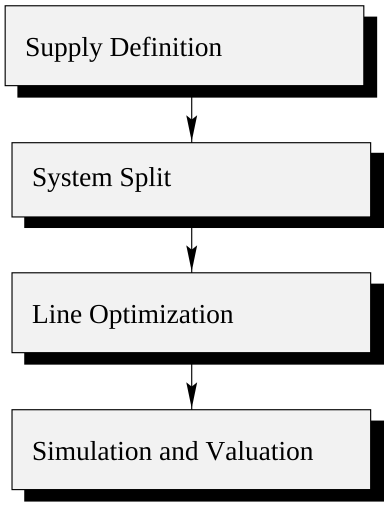


Figure 3.1: Partitioning of Line Planning

The determination of such a line plan should serve the transportation demand with an efficient usage of resources at a high level of quality. The transportation demand is usually given by an origin-destination matrix $T \in \mathbb{Z}_+^{n \times n}$ (n denotes the number of stations in the transportation network) where $T^{a,b}$ represents the number of passenger traveling from station a to station b. The modal split (cf. section 2.1) provides an origin-destination matrix for the global transportation network, only. A procedure introduced by OLTROGGE [55] distributes the passengers among the different supply networks. The main idea of this *system split* is described is section 3.2.

At this time, all necessary data is available and the essential part of the line planning problem, the *line optimization*, can be initiated. The optimization problem consists of finding a set of operating lines, given by routes and frequencies, subject to certain operational constraints that optimizes a given objective. Several different objective functions are proposed. On one hand we may be interested in minimizing the operational cost with respect to a given level of service and quality. On the other hand a maximization of the level of service for fixed operational cost is of valuable interest. A reasonable approach to improve the level of service is to minimize the *total travel time* of all passengers. But remember, at this stage of planning (cf. chapter 2) there is no train schedule, hence the exact waiting time while changing lines is unknown. Changing lines itself is a major inconvenience, hence the line plan which provides a minimum number of *changes*, or somewhat different, a maximum number of travelers on direct connections (*direct travelers*) is of interest.

In a final analysis the line plans of the several systems will be combined. The behavior of the passengers will be simulated and the interaction of the line plans will be valued by calculating different reference numbers.

3.1 Definition of supply networks

Apart from some shutdowns and newly-build routes, the physical track network keeps unchanged over a long period. This is a result of the long-term strategic decision process which precedes every modification in the railroad infrastructure. In contrast to that, the state of the market considerably varies and the transportation authorities must react with different offers in a more or less unchanged infrastructure. For example, several years ago, railroad companies offered *single* connections with fast trains and some local trains to meet the requirements of their customers. Nowadays, the situation and the *products* have changed. The core of a refined service is derived from *line-based* connections for long- and medium distance travelers as well as for local transportation. For the German railroad company (other European railroads have similar subdivisions) the specification of the *InterCity*, *InterRegio* and *Regional Express* or *AggloRegio* services is listed below.

InterCityExpress/InterCity (ICE/IC)

Trains of the ICE/IC system connect principal centers of a country. One of the remarkable features of these trains is the comfortable equipment with dining car, phone, and other *board services*. The average distance of adjacent stations is about 60 kilometers and the tracks must be electrified. The average transit speed is about 150 kilometers per hour and up to 250 kilometers per hour. The average operation time of 16 hours per day together with the 14 coaches per train result in a *line capacity* of about $7000 \cdot \varphi$ passengers per direction and day. $\varphi \in \mathbb{Z}_+$ denotes the frequency of the line with respect to a *basic time interval* of 60 minutes.

InterRegio (IR)

IR trains connect principal centers as well as district towns with an average transit speed of 90 kilometers per hour. The average distance of adjacent stations is about 60 kilometers. IR trains can be served by electric as well as by diesel locomotives, hence there are no further limitations for the tracks. The average number of coaches per train is 8. This leads together with the 16 hours of operation time to a line capacity of $2000 \cdot \varphi$ passengers per direction and day. Again, $\varphi \in \mathbb{Z}_+$ denotes the frequency of the line, but the basic time interval for the IR system is 120 minutes.

Regional Express Train/AggloRegio (AR)

Lines in such a system are designed for local transportation, act as feeder service for long-distance connections, and especially in peak hours serve as *push-in trains*. In Germany the bunch of different train services in local railroad transport is difficult to classify. This is one reason why there is no regional supply network for Germany in our set of test instances (cf. chapter 5).

The different supplies, offered by the railroad company, suggest a logical partition of the physical track network in so called supply networks. Such a network and of course the global railroad network itself, can be modeled using a finite *graph* $G_X = (V_X, E_X)$ where X represents the particular system (e.g. $X \in \{$IC, IR, AR$\}$). The set of *nodes* V_X represents the stations of the supply network and the set of *edges* E_X represents the connecting routes of adjacent stations. An edge $e \in E_X$ in general may consists of a sequence of eligible (e.g. electrified) tracks and stations $v \notin V_X$. G_X may be directed (e.g. networks with one-way tracks) or undirected. For simplicity, throughout the thesis we assume an undirected supply network, but all remaining models and methods can be easily extended to the directed case.

Figure 3.2: Supply networks of the Dutch railroad

The decision, if trains of the IC, IR, or AR system stop at a particular station v is based on the infrastructure of this station as well as on the volume of traffic at v. Usually, for railroad networks we have a hierarchical arrangement of the supply networks, like $V_{\text{IC}} \subset V_{\text{IR}} \subset V_{\text{AR}}$ (cf. figure 3.2 for the Dutch supply networks). Due to the physically disjoint networks in urban public transport the supply networks are more or less disjoint (exception: bus and express bus).

Certain attributes of the edges $e \in E_X$ in a supply network $G_X = (V_X, E_X)$, e.g. the *ride time* in minutes, can be expressed by a mapping $f\colon E_X \to S$, where S is an appropriate set, e.g. $S = \mathbb{Z}_+$ for the ride time mapping f^{RT}. Note that these attributes are sensible within the supply networks only, e.g. the ride time substantially varies for same edges in different supply networks (cf. discussion about average speed in IC, IR, and AR systems).

A line belongs to one system, exactly, hence the determination of a line plan for the global railroad network can be divided into line planning for each supply network in principle. However, some important and required information, namely the volume of traffic, is unavailable for the supply networks.

3.2 System split

The procedure proposed by OLTROGGE [55] *splits* the origin-destination matrix of the complete transportation network into origin-destination matrices for the supply networks. The idea of this method, called *system split*, is very simple. Assume there are a couple of passengers at a small station $a \in V_{\text{AR}}$ which want to travel to another small and far away station $b \in V_{\text{AR}}$. No fast train (ICE/IC or IR) stops at these stations, hence there is a slight hope only for a direct connecting train, and if it exists, it will be very slow. Therefore, we assume that the travelers take an AR train to the next station c, where an ICE/IC or IR train stops, use this fast train to reach a station d near station b and finally get on an AR train to station b. This idea is similar to a strategy used for path finding in road networks with different kinds of roads (highways, trunk roads, small roads). See CAR and FRANK [15] for more information on hierarchical reasoning in the context of path finding. Algorithmic and computational issues are discussed in [31, 72].

In general, a reasonable journey in the transportation network may start with a sequence of *system changes* to superior trains and may terminate with a sequence of changes to inferior trains. For the example mentioned above with systems ICE/IC, IR, and AR we obtain the following combinations.

AR
AR — IR — AR
AR — IR — ICE/IC — IR — AR
AR — IR — ICE/IC — AR
AR — ICE/IC — IR — AR
AR — ICE/IC — AR

The first combination represents travel paths that use AR trains only. The travel paths of the second combination start with some AR trains followed by IR connections and finish with one or more AR trains. With the additional assumption that travelers use the *shortest path* with respect to the ride time inside a system we can calculate the travel route for each combination. Therefore let $D_X \in \mathbb{Z}^{|V_X| \times |V_X|}$ be the *shortest path matrix* of the graph $G_X = (V_X, E_X)$ with edge length f^{RT}

(ride time). $D_X^{a,b}$ with $a,b \in V_X$ represents the length of a shortest path connecting a and b in G_X. The matrix D_X and the corresponding paths can be computed with the FLOYD-WARSHALL algorithm [33, 73]. Hence we can compute the travel route for each combination. For example

$$\min\{D_{\text{AR}}^{a,v_1} + D_{\text{IR}}^{v_1,v_2} + D_{\text{ICE/IC}}^{v_2,v_3} + D_{\text{AR}}^{v_3,b} \mid v_1 \in V_{\text{AR}} \cap V_{\text{IR}},\, v_2 \in V_{\text{IR}} \cap V_{\text{ICE/IC}},\, v_3 \in V_{\text{ICE/IC}} \cap V_{\text{AR}}\}$$

where a,b,v_1,v_2,v_3 are pairwise different, provides the length and the path itself of the travel route R for the combination AR — IR — ICE/IC — AR of the station pair a,b. One might as well apply the algorithms for hierarchical shortest path to each combination. From the passengers point of view the different reasonable combinations and the resulting travel path are more or less attractive concerning several attributes. The sophisticated valuation of the travel path proposed by OLTROGGE [55] is based on the ride time, price, level of comfort, and the number of system changes. Note that a system change always forces a change of lines. The passengers commuting between a and b do not form an integrated whole but can be classified by their *trip purpose*, e.g. business trips, private or vacation trips[1]. The valuation produces different results for different trip purposes and provides an assignment of the volume of traffic to the different travel routes. Let us assume that $t \leqslant T^{a,b}$ passengers of the origin-destination pair a, b use route r with

$$r = a \overset{\text{AR}}{\leftrightsquigarrow} v_1 \overset{\text{IR}}{\leftrightsquigarrow} v_2 \overset{\text{ICE/IC}}{\leftrightsquigarrow} v_3 \overset{\text{AR}}{\leftrightsquigarrow} b.$$

The t passengers contribute to the origin-destination pairs a, v_1 and v_3, b in the AR origin-destination matrix T_{AR}. Similarly, the t passengers increase the volume of traffic of origin-destination pair v_1, v_2 in T_{IR} and of pair v_2, v_3 in $T_{\text{ICE/IC}}$.

An aggregation over all possible routes and all origin-destination pairs leads to an origin-destination matrix for each supply network. Additionally, the distribution of passengers along the transportation network provides for each edge $e \in E_X$ the *traffic load* $ld(e)$, i.e. the number of passengers using a particular edge e.

This framework for line planning is widely accepted by researches as well as by practitioners. However, some parts, especially the system split, provides some weak spots, but even in the case where the line plan and the train schedule are known the computation of reasonable travel routes is not obvious [48].

3.3 Line optimization

The decomposition of the complete transportation network into supply networks described above permits a separate line optimization. In this section we briefly summarize some approaches to the line optimization problem that can be found in the transportation science literature. The papers under consideration give an overview of mathematical representations of line plans as well as appropriate objective functions. The line plans computed with the associated algorithms provide an *approximate* solution without any performance guarantee. The purpose of this thesis is to

[1] Origin-destination matrices of the German railway network are classified by **seven** trip purposes.

represent some models and algorithms that overcome these inadequacies, particularly, for real world instances of the line optimization problem.

In 1925 PATZ [61] represented a model for the line optimization problem that determines a line plan with small *penalty*. The penalty of line l is calculated with respect to the number of empty seats and the number of passengers in l changing to another line to reach their destination. The algorithm starts with a line plan containing a line for each origin-destination pair. Lines will be successively eliminated from the line plan in a greedy fashion with respect to the penalty. The capacity for passengers of the eliminated line will be assigned to other lines. The size and the structure of the network (a tree with 10 nodes) permits a detailed analysis, based on linear programming (!). For the instance of 10 nodes PATZ could prove optimality of the generated solution. The favorable analysis mainly depends on the size and structure of the particular instance and cannot be extended to more general networks.

WEGEL [79] introduced the widespread notion of *line frequency requirements*. For every edge e of the transportation network the line frequency requirement $lfr(e)$ represents the required number of trains in a line plan to serve the traffic load $ld(e)$ (number of passengers) on edge e. A fixed line/vehicle capacity C permits the computation of the required number of lines for edge e by $lfr(e) = \lceil ld(e)/C \rceil$. The method of WEGEL computes line plans that maximize the number of direct travelers subject to the line frequency requirement for each edge e. The algorithm starts with a basic line plan that covers each edge with one line ($lfr \equiv 1$), exactly. For a slight generalized problem DIENST [27] introduced a *branch-and-bound* procedure (cf. section 5.3 for a detailed description) that computes a basic line plan with a maximal number of direct travelers. Afterwards, some lines are added to the basic line plan with respect to the remaining line frequency requirement in order to reduce the number of changes between lines.

In the doctoral thesis of SONNTAG [70, 71] a procedure for computing line plans with a small sum of average travel times is represented. In contrast to former models, the *set of possible lines* is restricted. A line must begin and end in a so called *classification yard*. Similar to the method of PATZ the algorithm starts with a line plan containing a line for each origin-destination pair. This line plan is reduced by eliminating lines and diverting the passengers to short but not necessarily shortest travel paths with respect to the ride time. Combining this elimination with the connection of lines, after some iterations a line plan of appropriate size, with small average travel times, and a large number of direct travelers is computed.

SIMONIS [69] constructs a line plan iteratively starting with an empty line plan. The algorithm successively chooses lines on shortest paths with a maximum number of direct travelers. The procedure terminates if all passengers find an appropriate travel path or the line plan exhausts a preset length.

PAPE et. al. [58] suggest a decomposition of the set of possible lines. Lines containing a large number of travelers, the so called *core lines*, are combined in a complete enumeration scheme. The *best* partial line plan with respect to the number of direct travelers is extended by lines with low traffic to include uncovered edges in the line plan.

This collection of papers concerning the line optimization problem does not claim to be comprehensive, however, it summarizes the constituents of models, objectives, and algorithms. Furthermore, a general problem with these heuristics becomes evident. The only way of valuating a generated solution is to compare it with solutions derived from alternative algorithms for the same model. This local information cannot be extended to a performance guarantee referring to an *optimal* solution. A convenient approach for providing provable good solutions is the analysis of the heuristic (e.g. cf. [63] for the analysis of the tree heuristic for the Δ–TSP) or the computations of lower and upper bounds on the optimal solution by solving some relaxed problems.

3.4 Simulation and valuation

In the final step of the framework proposed by OLTROGGE the line plans of the different supply networks individually generated by a line optimization procedure will be composed. This composition together with the initial origin-destination matrix is analyzed by simulating the passengers' behavior when traveling from their origin to their destination. The simulation is based on a more realistic model of passengers' behavior than the optimization models. The simulation terminates with a bunch of reference numbers like

- number of direct travelers,
- number of changes,
- capacity utilization,
- total travel time.

An experienced human planner may take advantage of these numbers. The adjustment of some parameters of the system split or the line optimization procedure can be used to model several operational and political constraints which cannot be included in a mathematical model. In order to provide an interactive and flexible *decision support system* for supply planning in public transportation, each step of the framework described above (cf. figure 3.1) must be efficiently performed. From the computational point of view the line optimization represents the bottleneck. Hence a fast algorithm which produces provable good solutions is of valuable interest.

In chapter 5 and 6 we focus on particular objectives for the line optimization problem. The resulting models and methods are based on a mathematical programming approach. Therefore, in the next chapter we briefly introduce the mathematical background for these kind of models.

Chapter 4

Models for line planning

In this chapter we give the fundamentals for a proper mathematical modeling of line plans. A mathematical programming approach requires a precise definition of all objects, i.e. the set of possible lines $\mathcal{L}$, the *line plan* L itself, and the resulting set $\mathbb{L}$ of all possible line plans. The definition of the fundamental term *line* requires some mathematical basics. Although, we already used some notations from graph theory in an informal way in the previous chapters, we give a mathematical definition which follows the notation of Nemhauser and Wolsey [52]. Moreover, we briefly summarize the fundamentals of computational complexity. In section 4.5 we prove that the generic line planning problem belongs to the class of intractable recognition problems which justifies an integer linear programming approach. The solution methods for integer linear programming problems, described in section 4.7 have a close connection to *rational polyhedra* presenting the set of feasible solutions of the integer linear program to some extend. In section 4.3 we investigate the basic notation and well known results from polyhedral theory.

4.1 Graphs

An (undirected) graph $G = (V,E)$ consists of a finite, nonempty set $V = \{v_1, v_2, \dots, v_n\}$ and a set $E = \{e_1, e_2, \dots, e_m\}$ whose elements are subsets of V of size 2, that is, $e_k = \{v_i, v_j\}$ (or $v_i v_j$ for short) with v_i, $v_j \in V$. The elements of V are called *nodes*, and the elements of E are called *edges*. We say $e \in E$ is *incident* to $v \in V$ or that v is an *endpoint* or *terminal* of e if $v \in e$. One way to represent a graph is by its $n \times m$ *node-edge incidence matrix* $A = (a_{ij})$ where

$$a_{ij} = \begin{cases} 1 & \text{if } e_j \text{ is incident to node } v_i \\ 0 & \text{otherwise} \end{cases}$$

Note that each column of A contains exactly two 1's. The number of 1's in row i equals the number of edges incident to node v_i and is called the *degree* of node v_i. The set of edges incident to node v_i is denoted by $\delta(v_i)$. This can be generalized to node sets of size greater than 1. Let $T \subset V$ then $\delta(T) = \{e = v_i v_j \in E \mid v_i \in T, v_j \notin T\}$.

A node sequence $u_0, u_1, \dots, u_k$, $k \geqslant 1$ is called a u_0–u_k *walk* if $u_{i-1}u_i \in E$ for $i = 1, \dots, k$. Node u_0 is called the *origin*, node u_k is called the *destination*, and nodes $u_1, \dots, u_{k-1}$ are intermediate nodes. We can also represent a walk by its edge sequence $e_1, e_2, \ldots, e_k$ where $e_i = u_{i-1}u_i$ for $i = 1, \ldots, k$. The *length* of a walk $e_1, e_2, \ldots, e_k$ with respect to an edge weight $w : E \to \mathbb{R}$ is $\sum_{i=1}^{k} w(e_i)$. A u_0–u_k walk p is said to be the *shortest* u_0–u_k *walk* with respect to weight w, if there is no other u_0–u_k walk p' with length $w(p') < w(p)$. A walk is called a *path* if there are no node repetitions. A u_0–u_k walk is said to be *closed* if $u_0 = u_k$. A closed walk is called a *cycle* if $k \geqslant 3$ and $u_0, u_1, \ldots, u_{k-1}$ is a path. A graph is said to be *acyclic* if it does not contain any cycles. G is said to be *connected* if for all pairs $u, v \in V$ there is a path with origin u and destination v. An acyclic and connected graph is called a *tree*.

For various applications is it useful to assign a direction to the edges of a graph. The resulting *directed graph* or *digraph* $D = (V, A)$ consists also of a finite nonempty set V of nodes and a set $A = \{e_1, e_2, \ldots, e_m\}$ whose elements are *ordered* subsets of V of size 2 called *arcs*. In a digraph, $v_i v_j$ and $v_j v_i$ are different elements and we may have neither, one, or both of these elements in A. By removing the direction from the arcs of a digraph D, that is, replacing the arcs by edges and removing edge duplications, we obtain a graph G that is said to *underlie* D. Conversely, the replacement of edges $\{v_i, v_j\}$ by $v_i v_j$ and $v_j v_i$ we obtain a directed version $\vec{G} = (V, \vec{E})$ of $G = (V, E)$. Directed walks, paths, cycles and other elements can be defined similarly to graphs.

4.2 Computational complexity

The theory of *computational complexity* attempts to categorize the computational requirement of algorithms and important classes of problems. Although, this theory and the corresponding notation can be found in a wide range of textbooks (e.g. cf. [1, 35, 52, 57]) we include this section to make the thesis more self-contained.

Before we enter the theory of $\mathcal{NP}$*-completeness* we derive the fundamental *time complexity function* for an algorithm. This function measures the *running time* of an algorithm by means of the number of *basic operations*, like assignment steps (assigning some value to a variable), arithmetic steps (addition, subtraction, multiplication, and division), and logical steps (e.g. comparison of two numbers). The running time of an algorithms depends on both the nature and the size of the input. The *size of the input* is the number of bits needed to store all the data that defines a particular problem instance. For example, the size of an integer i is $1 + \lceil \log |i| \rceil$ and hence the size of a knapsack problem instance with n items, profit values $p_1, \ldots, p_n \in \mathbb{Z}_+$, size $s_1, \ldots, s_n \in \mathbb{Z}_+$, and the knapsack size $b \in \mathbb{Z}_+$ is $1 + \lceil \log b \rceil + \sum_{i=1}^{n}(1 + \lceil \log p_i \rceil + 1 + \lceil \log s_i \rceil)$.

The time complexity function for an algorithm is a function of the problem size and specifies the largest amount of time (number of basic steps) needed by the algorithm to solve *any* problem instance of given size n. In order to classify complexity classes of algorithms and problems it is sufficient to measure the complexity function by means of an asymptotic growth rate. Therefore, we introduce the “big O” notation. An algorithm is said to run in $O(f(n))$ time if for some numbers c and n_0, the time complexity function is at most $c \cdot f(n)$ for all $n \geqslant n_0$. An algorithm is said to be a *polynomial-time algorithm* of *efficient algorithm* if the algorithms runs in $O(f(n))$, where f is bounded by a polynomial, e.g. $O(n^2)$ and $O(n \log n)$. An algorithm is said to be an *exponential-time algorithm* if its complexity function cannot be polynomially bounded by the input size n, e.g. $O(2^n)$ and $O(n!)$.

Obviously, polynomial-time algorithms are “good” algorithms. Nevertheless, we might not succeed in developing a polynomial-time algorithm for a particular problem. The theory of NP-*completeness* provides us a way to prove that the problem is inherently hard in the sense that if we can develop an efficient algorithm for this problem, we would be able to develop an efficient algorithm for a huge class of intractable problems, including famous problems like the *traveling salesman problem* (TSP) and *graph coloring*.

The theory of NP-completeness helps us to classify a given problem into broad classes:

1. easy problems that can be solved by polynomial-time algorithms, and
2. hard problems that are not likely to be solved in polynomial-time and for which all known algorithms require exponential time.

Most of the problems discussed in this thesis are optimization problems. The theory of NP-completeness requires that problems are stated so that we can answer them with a *yes* or *no*. It is easy to see that the optimization and the *recognition* version of a problem are equivalent in terms of whether or not they can be solved in polynomial time. We refer to an instance of the recognition problem as a *yes* instance if the answer to this problem instance is yes, and a *no* instance otherwise. We say that a problem P_1 *polynomially transforms* to another problem P_2 if for every instance I_1 of P_1 we can construct in polynomial-time in terms of the size of I_1 an instance I_2 of P_2 so that I_1 is a yes instance if and only if I_2 is a yes instance of P_2. If problem P_1 polynomially transforms to problem P_2, P_2 is at least as hard as P_1: Given an algorithm for problem P_2 we can always use it to solve problem P_1 with comparable (i.e. polynomial or not) running times.

Class P

We say that a recognition problem P belongs to class P if some polynomial-time algorithm solves problem P.

Class NP

For a recognition problem P to be in NP we require that if I is a yes instance of P, then there exists a concise (that is, of length bounded by a polynomial in size of I) *certificate* for I, which can be checked by a *certificate-checking algorithm* in polynomial-time for validity.

Class NP-complete

A recognition problem P is said to be (in) NP-complete if

1. $P \in NP$, and
2. all other problems in the class NP polynomially transform to P.

We establish the completeness part (2) of an NP-completeness proof for a problem P_1 by showing that a known NP-complete problem, say P_2, polynomially transforms to P_1.

If we do succeed in showing that a problem is $\mathcal{NP}$-complete, we have sufficient reasons to believe that the problem is hard and no efficient algorithm can ever be developed to solve it. We should concentrate our efforts on developing efficient heuristics and at developing various types of enumeration algorithms.

4.3 Polyhedral theory

In this section we briefly summarize well known results and notions from *polyhedral theory*. A comprehensive discussion, including basics of linear algebra, can be found in [52].

A *polyhedron* $P \subset \mathbb{R}^n$ is a set of points that satisfy a finite number of linear inequalities, that is, $P = \{x \in \mathbb{R}^n \mid Ax \leqslant b\}$ where (A,b) is an $m \times (n+1)$ matrix. If P is bounded, i.e. $P \subset \{x \mid \Leftrightarrow\omega \leqslant x \leqslant \omega\}$ for an $\omega \in \mathbb{R}$, the polyhedron is called a *polytope*. A polytope is of *dimension* k, denoted by $\dim(P) = k$, if the number of affinely independent points in P is $k+1$. Let $M^{=} := \{i \in \{1,\dots,m\} \mid a^i x = b^i \text{ for all } x \in P\}$ and let $(A^{=}, b^{=})$ be the corresponding rows of (A,b). If $P \subset \mathbb{R}^n$, then $\dim(P) + \mathrm{rg}(A^{=}, b^{=}) = n$, where $\mathrm{rg}(A^{=}, b^{=})$ denotes the rank of matrix $(A^{=}, b^{=})$. An inequality $\pi x \leqslant \pi_0$ is called a *valid inequality* for polytope P if it is satisfied by all points in P. If $\pi x \leqslant \pi_0$ is a valid inequality for P, and $F = \{x \in P \mid \pi x = \pi_0\}$, F is called a *face* of P, and we say that $\pi x \leqslant \pi_0$ represents F or $\pi x \leqslant \pi_0$ *defines* the face F. A face of P is a *facet* of P if $\dim(F) = \dim(P) \Leftrightarrow 1$. The single point of a zero-dimensional face $F = \{x_0\}$ of a polytope P is said to be an *extreme point* of P. A point $x \in P$ is an extreme point of polytope P if and only if there do not exist $x^1, x^2 \in P$, $x^1 \neq x^2$ such that $x = \frac{1}{2}x^1 + \frac{1}{2}x^2$. A non-empty polytope can be characterized by *convex combinations* of its extreme points (MINKOWSKI's theorem), i.e.

$$P = \{x \in \mathbb{R}^n \mid x = \sum_{k \in K} \lambda_k x^k,\ 1^T\lambda = 1,\ \lambda \geqslant 0\} =: \mathrm{conv}\ \{x^k \mid k \in K\}$$

where $\{x^k \mid k \in K\}$ is the set of extreme points of P. Conversely, the convex combination of any finite set of points can be identified as a polytope (WEYL'S theorem). Hence we can represent a polytope either by its linear description $Ax \leqslant b$ or by using the convex combinations of its extreme points.

In linear programming the set of feasible points P can be described by a set of linear inequalities $P = \{x \mid Ax \leqslant b, x \in \mathbb{R}^n_+\}$. Integer linear programming is different. Typically, we have a set $S \subset \mathbb{Z}^n_+$ of feasible solutions and implicitly describe this set using linear inequalities and the integrality add-on $S = \{x \mid Ax \leqslant b, x \in \mathbb{Z}^n_+\}$. By WEYL'S theorem we know that there is a linear description of $\mathrm{conv}\ (S) = \{x \mid A'x \leqslant b', x \in \mathbb{R}^n_+\}$. Unfortunately, for integer linear programs representing a model for an $\mathcal{NP}$ complete problem is it most improbable (unless $\mathcal{NP} = \mathrm{co}\mathcal{NP}$) that the corresponding linear description has a "good characterization" [56].

From the computational point of view, we are not looking for the complete linear description of $\mathrm{conv}\ (S)$ but are interested in a representation of an integer linear program by a linear program that has the same optimal solution. Even if we cannot establish a linear representation or formulation with this property we should concentrate on a linear representation that provides an improved linear programming relaxation. The linear programming relaxation plays a major role in relaxation algorithms like *branch-and-bound* for solving integer linear programs.

In general, there are several linear formulations $P_i = \{x | A^i x \leqslant b^i, x \in \mathbb{R}^n_+\}$ representing S, i.e. $P_i \cap \mathbb{Z}^n_+ = S$ (cf. figure 4.1). We say P_i is *tighter* than P_j if and only if $P_i \subset P_j$. For two linear formulations P_i, P_j representing S we have

$$\max\{c^T x \mid x \in \text{conv}\,(S)\} \leqslant \max\{c^T x \mid x \in P_i\} \leqslant \max\{c^T x \mid x \in P_j\}$$

if P_i is tighter than P_j. A linear formulation representing S can be tightened by adjuncting addi-

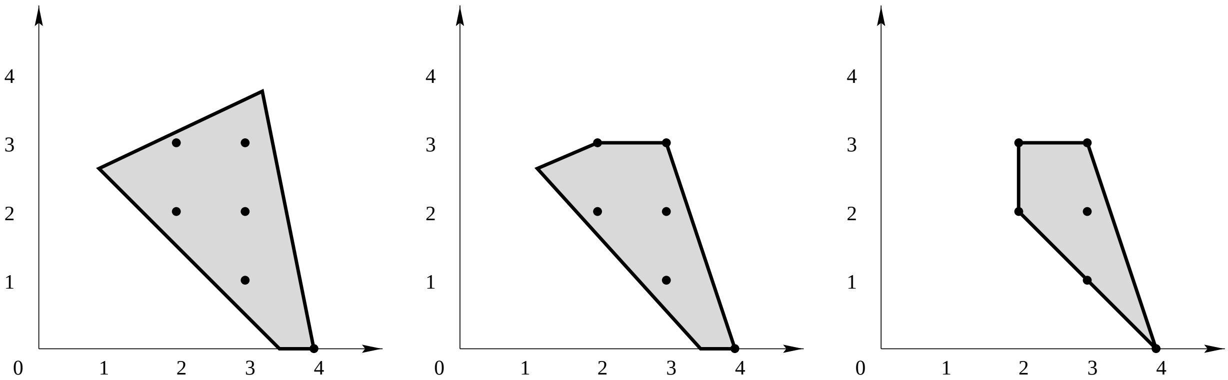


Figure 4.1: Feasible regions representing $S = \{(2,3),(2,2),(3,3),(3,2),(3,1),(4,0)\}$

tional linear inequalities.

We already mentioned that the linear inequality $\pi^T x \leqslant \pi_0$ with $(\pi, \pi_0) \in \mathbb{Q}^{n+1}$ is said to be a valid inequality or a *cut* of conv (S) if and only if $\pi^T x \leqslant \pi_0$ for all $x \in$ conv (S). A valid inequality $\pi^T x \leqslant \pi_0$ of conv (S) is not necessarily a valid inequality for a linear formulation representing S. Hence the adjunction of $\pi^T x \leqslant \pi_0$ to the linear representation of S may yield a tighter formulation. For each *facet* of the polytope conv (S), a valid inequality that represents this facet is necessary in the linear description of S (cf. [52] I.4.3). Hence cuts representing faces of high dimension and in particular facet-defining inequalities are of valuable interest.

Given a point $x^* \in \mathbb{R}^n_+$, the problem of showing that $x^* \in$ conv (S) or finding a violated valid inequality (π, π_0) of conv (S), i.e. $\pi^T x^* > \pi_0$ is known as the *separation problem*. Separation is most important in the context of *cutting plane methods* (cf. sections 4.9.2). Although the general separation problem can be polynomially transformed to the original optimization problem and vice versa [38], separation of particular classes of valid inequalities for some combinatorial problems significantly improves the linear programming relaxation.

4.4 A linear edge formulation

We have already noticed that graphs are a convenient way to represent supply networks. A line $l = (r, \varphi) \in \mathcal{L} \subset \mathcal{R} \times \mathbb{Z}_+$ consists of a *route* $r \in \mathcal{R}$ and a *frequency* $\varphi \in \mathbb{Z}_+$. The route $r \in \mathcal{R}$ is a path or a cycle[1] in the supply network $G_X = (V_X, E_X)$ (henceforth we omit the index

[1]Lines on a cyclic track sometimes occur in practice, e.g. the famous *Circle Line* of the London tube.

$X \in \{$ICE/IC, IR, AR$\}$). Below we will restrict $\mathcal{L}$ to lines with particular routes in order to cover certain operational constraints. The frequency $\varphi \in \mathbb{Z}_+$ denotes the number of trains that serve the line within the basic time interval $[0,\ldots,\tau)$ of system X. The concept of the *line frequency requirement* introduced by WEGEL [79] reflects capacity and operational constraints. For each edge $e \in E$ we introduce lower and upper bounds $\underline{lfr}(e) \leqslant \overline{lfr}(e)$ on the number of trains in the basic time interval. This extends WEGEL'S approach that comes with $\underline{lfr} = \overline{lfr}$. Here are some of the constraints which can be modeled using $\underline{lfr}$, $\overline{lfr}$.

- For a frequently used edge e the safety regulations, e.g. the minimum headway h, i.e. the temporal distance of consecutive trains, provide $\overline{lfr}(e) \leqslant \lfloor \tau/h \rfloor$.
- If the line plan shall be designed to transport *all* passengers, the load $ld(e)$ of an edge together with a fixed (train) capacity C gives a lower bound on the number of required trains: $\underline{lfr}(e) \geqslant \lceil ld(e)/C \rceil$.
- For economical reasons the number of empty seats should be bounded from above by e.g. 20%. This leads to $\overline{lfr}(e) \leqslant \lceil 1.2 \cdot ld(e)/C \rceil$.

The examples above show that the traffic load ld plays an important role in computing $\underline{lfr}$ and $\overline{lfr}$. One way to compute ld is represented in the system split procedure (cf. section 3.2). A different, widely used method is based on KIRCHHOFF'S laws for electrical circuits. For an origin-destination pair a, $b \in V$ compute the $k_{a,b}$ shortest paths connecting a and b with respect to the ride time f^{RT} (cf. [18] for suitable algorithms and additional references). A suitable value for $k_{a,b}$ mainly depends on the type of the transportation network. For example, in an urban transportation system a path that is twice as long as the shortest path is still of valuable interest. For long-distance railroad networks such paths will never be accepted and hence $k_{a,b}$ will be of substantially smaller size. Let $P^{a,b}$ be the set of the $k_{a,b}$ shortest paths between a and b. The length of paths corresponds to the resistance in a parallel electrical circuit. According to KICHHOFF's law, the ld value can be computed as follows.

$$ld(e) = \sum_{a,b \in V} \sum_{\substack{p \in P^{a,b} \\ e \in p}} \frac{T^{a,b}}{1 + \sum_{p' \in P^{a,b} \setminus p} \frac{f^{RT}(p)}{f^{RT}(p')}}$$

With the notation of line frequency requirement we can now give a precise definition of a *feasible* line plan, i.e. a line plan that fulfills the constraints modeled by $\underline{lfr}$ and $\overline{lfr}$.

Given a graph $G = (V,E)$, bounds $\underline{lfr}(e)$ and $\overline{lfr}(e)$ (without loss of generality suppose $\underline{lfr} \leqslant \overline{lfr}$ and $\overline{lfr} \geqslant 1$) for each edge $e \in E$, and a set of possible lines $\mathcal{L}$. $L \subset \mathcal{L}$ is a *feasible* line plan if and only if

$$\underline{lfr}(e) \leqslant \sum_{\substack{(r,\varphi) \in L \\ e \in r}} \varphi \leqslant \overline{lfr}(e) \tag{4.1}$$

holds for each edge $e \in E$.

Due to a close relationship of paths, cycles and *flows* or *circulations* [1] we obtain an elegant description of $\mathbb{L}$, which represents the set of all feasible line plans. Consider the directed version $\vec{G} = (V, \vec{E})$ of $G = (V, E)$. Add a *supernode* v_s to V and enlarge $\vec{E}$ by arcs $v_s v$, $v v_s$ for all $v \in V$. Consider the following integer linear system.

$$\text{(EDGE)} \quad \begin{aligned} \sum_{uv \in \vec{E}} x_{uv} - \sum_{vu \in \vec{E}} x_{vu} &= 0 && \forall\, u \in V && (4.2)\\ x_{uv} + x_{vu} &\geqslant \underline{lfr}(\{u,v\}) && \forall\, \{u,v\} \in E && (4.3)\\ x_{uv} + x_{vu} &\leqslant \overline{lfr}(\{u,v\}) && \forall\, \{u,v\} \in E && (4.4)\\ x_{uv} &\in \mathbb{Z}_+ && \forall\, uv \in \vec{E} && (4.5) \end{aligned}$$

A vector $x \in \mathbb{R}_+^{|\vec{E}|}$ fulfilling the *linear equalities, inequalities* (4.2)–(4.4) and the integrality requirement (4.5) of system (EDGE) is said to be a *feasible circulation* in $\vec{G}$.

For $L = \{(r, \varphi) \mid r$ is a path or a cycle in G and $\varphi \in \{1, \ldots, \min_{e \in r} \overline{lfr}(e)\}\}$ the *flow decomposition theorem* ([1] section 3.5, p. 80) provides the following relation between feasible line plans and feasible circulations. Note that with this particular choice of L there is always a feasible line plan, e.g. $L = \{(r, \varphi) \mid r = e,\ \varphi = \overline{lfr}(e)\}$.

PROPOSITION 4.1
Every feasible line plan $L \in \mathbb{L}$ has a unique representation as a feasible circulation. Conversely, every feasible circulation describes a line plan (though not necessarily uniquely).

PROOF The algorithmic proof in [1] directly applies to our situation and hence we omit it. □

The representation of line plans by circulations has certain limits. Due to an ambiguous representation of line plans by one circulation, line plans with equal edge frequencies ($\sum_{(r,\varphi) \in L, e \in r} \varphi$) do not differ in the edge formulation. This is an intolerable property if we take the potential valuations of line plans (cost, direct travelers) into account. Furthermore, the edge formulation is inapplicable to handle operational constraints, e.g. a restriction of R to routes with a minimum and maximum length.

4.5 Complexity results

The simple choice of R and hence of L stated in the previous section does not fit the requirements for practical line planning. The determination of the set of possible routes R is subject to various rules depending on the particular network and the operating authority. The following enumeration represents a couple of convenient constraints for possible routes $r \in R$.

1. r is a (simple) path in G.
2. The length of a route according to the ride time is bounded from below and above.
3. The origin and destination of a route belongs to a particular subset $V' \subset V$.
4. Some node/edge sequences are excluded in r.

Constraints 3 and 4 reflect a particular situation in rail networks. A station in which a line may start/end must have a special equipment (e.g. sidings to compose trains or reversing loops for trams). Let $V' \subset V$ describe these *classification yards*. Figure 4.2 illustrates a *switch point* at b. All routes with node sequence a, b, d must be excluded.

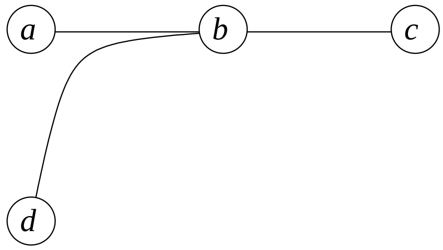


Figure 4.2: A switching point

The operational constraints dramatically reduce the number of possible routes and hence the size of $\mathcal{L}$, e.g. a real-world instance of NS Reizigers, a business unit of Nederlandse Spoorwegen, contains 832 possible routes; the corresponding network with 38 nodes and 52 edges provides 568107 different paths.

With a general choice of routes it is no longer obvious that there exists a feasible line plan even if we keep the set of frequencies untouched, i.e. $\varphi \in \{1, \ldots, \min_{e \in r} \overline{lfr}(e)\}$. This *feasibility problem* can be formulated in the notation of GAREY and JOHNSON [35].

FEASIBLE LINE PLANS (FLP)
INSTANCE: Set $\mathcal{R}$ of paths of a graph $G = (V, E)$, lower $\underline{lfr}(e)$ and upper $\overline{lfr}(e)$ bounds for the line frequency requirement for each edge, and a set $\mathcal{L} = \{(r, \varphi) \mid r \in \mathcal{R},\ \varphi \in \{1, \ldots, \min_{e \in r} \overline{lfr}(e)\}\}$ of lines.
QUESTION: Is there a subset $L \subset \mathcal{L}$ that satisfies (4.1), i.e. L is a feasible line plan?

THEOREM 4.2
FEASIBLE LINE PLAN *is* $\mathcal{NP}$*-complete.*

PROOF The polynomially transformation of the $\mathcal{NP}$-complete problem EXACT COVER BY 3-SETS (X3C) [35] together with the obvious result that FLP belongs to the class $\mathcal{NP}$, proves the statement. An instance of X3C consists of a set X with $|X| = 3q$ and a collection C of 3-element subsets of X. Does C contain an exact cover for X, i.e. a subcollection $C' \subset C$ such that every element of X occurs in exactly one member of C'?

With X and C we construct an instance of FLP in the following way. The graph $G = (V, E)$ consists of nodes $\underline{x}$, $\overline{x}$ for each $x \in X$ and edges $\overline{x_i}\underline{x_j}$, $\overline{x_j}\underline{x_k}$ for each $c = (x_i, x_j, x_k) \in C$ and $\underline{x}\overline{x}$ for each $x \in X$ (eliminate duplicated edges). The set of paths is defined as follows.

$$\mathcal{R} := \{r \mid r = \underline{x_i}, \overline{x_i}, \underline{x_j}, \overline{x_j}, \underline{x_k}, \overline{x_k} \text{ for each } c = (x_i, x_j, x_k) \in C\}$$

The definition of $\overline{lfr} \equiv 1$ and

$$\underline{lfr}(e) = \begin{cases} 1 & \text{if } e = \underline{x}\overline{x} \\ 0 & \text{otherwise} \end{cases}$$

results in this particular set of possible lines $\mathcal{L} := \{(r, 1) \mid r \in \mathcal{R}\}$ (cf. figure 4.3). This transformation provides the necessary equivalence. Suppose C contains an exact cover C' of X then

$$L := \{(r, 1) \mid r = \underline{x_i}, \overline{x_i}, \underline{x_j}, \overline{x_j}, \underline{x_k}, \overline{x_k} \text{ for each } c = (x_i, x_j, x_k) \in C'\} \subset \mathcal{L}$$

represents a feasible line plan. Conversely, let $L \subset \mathcal{L}$ be a feasible line plan then

$$C' := \{(x_i, x_j, x_k) \mid r = \underline{x_i}, \overline{x_i}, \underline{x_j}, \overline{x_j}, \underline{x_k}, \overline{x_k} \text{ with } (r,1) \in L\} \subset C$$

is an exact cover of X. The observation that the transformation is polynomial completes the proof □

$X = \{1,2,3,4,5,6\}$
$C = \{(1,3,6),\ (1,4,6)$
$\quad (2,3,5),\ (4,5,6)\}$

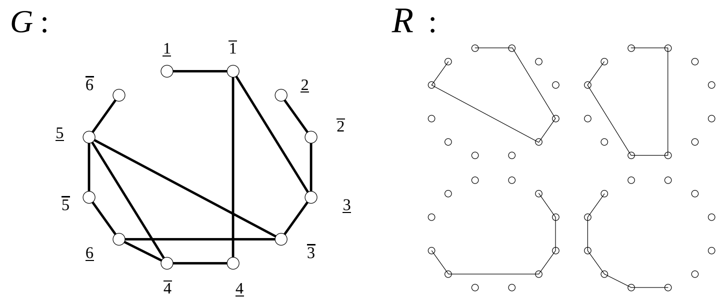


Figure 4.3: An example of the transformation X3C to FLP

COROLLARY 4.3
FLP with $\underline{lfr} \equiv \overline{lfr} \equiv 1$ *remains* $\mathcal{NP}$*-complete.*

PROOF The proof of theorem 4.2 directly applies if we enhance the set of paths $\mathcal{R}$ by paths $\overline{x_i}, \underline{x_j}$ and $\overline{x_j}, \underline{x_k}$ for each $c = (x_i, x_j, x_k) \in C$. Note that the transformation remains polynomial. We fulfill the requirement of edges not covered by lines of length 5 (length is measured according to the number of edges) by the one-edge lines. Conversely, C' is constructed by lines of L of length 5 only. □

If the answer to the recognition problem FLP is negative the data (graph, set of lines, $\underline{lfr}$, $\overline{lfr}$) must be modified in order to find a feasible line plan. In some cases the reduction of $\underline{lfr}$ and the augmentation of $\overline{lfr}$ will overcome the feasibility problem. But the problem of determining an *optimal* adjustment of the line frequency requirement is as hard as the line planning problem itself, as we can see in the next proposition.

MINIMUM ADJUSTMENT OF $\underline{lfr}$, $\overline{lfr}$
INSTANCE: Set $\mathcal{R}$ of paths of a graph $G = (V,E)$, lower $\underline{lfr}(e)$ and upper $\overline{lfr}(e)$ bounds on the line frequency requirement for each edge, integers k, K, and a set $\mathcal{L} = \{(r,\varphi) \mid r \in \mathcal{R},\ \varphi \in \{1,\dots,\min_{e\in r} \overline{lfr}(e)\}\}$ of lines.
QUESTION: Are there adjustments $\underline{lfr}' := \underline{lfr} - \underline{x}$ and $\overline{lfr}' := \overline{lfr} + \overline{x}$ of at most k and K units, i.e. $\underline{x}$, $\overline{x} \in \mathbb{Z}_+^{|E|}$ and $1^T \underline{x} \leqslant k$, $1^T \overline{x} \leqslant K$, such that $(G, \mathcal{L}, \underline{lfr}', \overline{lfr}')$ provides a feasible line plan?

COROLLARY 4.4
MINIMUM ADJUSTMENT OF $\underline{lfr}$, $\overline{lfr}$ *is* $\mathcal{NP}$*-complete.*

PROOF FLP is a subproblem of MINIMUM ADJUSTMENT OF $\underline{lfr}$, $\overline{lfr}$ ($k = K = 0$). The obvious membership of this problem in the class $\mathcal{NP}$ completes the proof. □

4.5.1 Polynomially solvable cases

Besides the disappointing hardness results there are particular cases of the FEASIBLE LINE PLAN problem that can be solved in polynomial-time. Trivially, we have the case where $\mathcal{R}$ contains the one-edge routes $r = e$ for each edge $e \in E$. Until further notice we will restrict to instances with frequencies $\varphi \in \{1, \dots, \min_{e \in r} \overline{lfr}\}$ for routes $r \in \mathcal{R}$. Hence an instance of the line planning problem is given by G, $\mathcal{R}$, $\underline{lfr}$, and $\overline{lfr}$.

Another *good-natured* class of FLP instances is related to *star graphs* (cf. figure 4.4). A star graph $G = (V, E)$ consists of one center node v and some circumjacent nodes $v_1, \dots, v_k$, and edge set $E = \{vv_i \mid i = 1, \dots, k\}$. First of all let $\underline{lfr} \equiv \overline{lfr} \equiv 1$ and without loss of generality suppose that $\mathcal{R}$ consists of several two-edge routes v_i, v, v_j. It is quite obvious that such an instance of FLP contains a feasible line plan if and only if there is a *perfect matching* in the graph $G' = (\{v_1, \dots, v_k\}, E')$ where $v_i v_j \in E'$ for each route $r = v_i, v, v_j \in \mathcal{R}$. Generalized matchings, so called *b-matchings* provide the solution of FLP instances with arbitrary $b_{vv_i} := \underline{lfr}(e) = \overline{lfr}(e) \in \mathbb{Z}_+$. Let $x \in \mathbb{Z}_+^{|E'|}$ be a perfect b-matching, i.e.

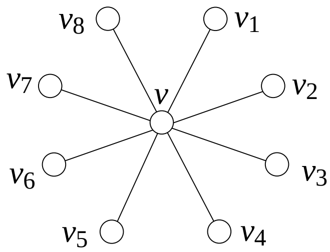


Figure 4.4: A star graph

$$\sum_{v_i v_j \in \delta(v_i)} x_{v_i v_j} = b_{vv_i} \qquad \text{for all } i = 1, \dots, k$$

where $\delta(v_i)$ denotes the set of edges incident to v_i in G'. The b-matching x in G' corresponds to a feasible line plan

$$L := \{(r, \varphi) \mid r = v_i, v, v_j;\ \varphi = x_{v_i v_j} \text{ for each } x_{v_i v_j} > 0,\ v_i v_j \in E'\} \subset \mathcal{L}$$

with respect to $\overline{lfr} \equiv \underline{lfr}$. Perfect ($b$-)matchings can be computed in strongly polynomial-time [24]. Hence FLP is polynomially solvable for star graphs. Beyond the feasibility problem the efficient algorithms for maximum weighted perfect b-matchings, i.e.

$$\max\{\sum_{v_i v_j \in E'} w_{v_i v_j} \cdot x_{v_i v_j} \mid \sum_{v_i v_j \in \delta(v_i)} x_{v_i v_j} = b_{vv_i} \text{ for all } i = 1, \dots, k,\ x_{v_i v_j} \in \mathbb{Z}_+\}$$

provides a solution of some optimization variants of the line planning problem, e.g. where $w_{v_i v_j}$ represents the profit respectively $\Leftrightarrow w_{v_i v_j}$ the cost of the line $((v_i, v, v_j), 1)$. Variants of the b-matching problem lead to particular variants of the line planning problem in star graphs, e.g. binary perfect b-matchings ($x_{v_i v_j} \in \{0, 1\}$) restrict the frequency f of lines to 1. An overview of polynomial cases of the b-matching problem and related algorithms can be found in the recent book of COOK, CUNNINGHAM, PULLEYBLANK, and SCHRIJVER [24]. Furthermore ARÓS et. al. [4] give (polyhedral) reductions of b-matching problems to the 1-matching case, including *matching-coverings*, i.e.

$$\underline{b}_{v_i v_j} \leqslant \sum_{v_i v_j \in \delta(v_i)} x_{v_i v_j} \leqslant \overline{b}_{v_i v_j}$$

which permit the transformation of line planning problems in star graphs with $\underline{lfr} \leqslant \overline{lfr}$.

Instances based on star graphs seem to be of theoretical interest only, but if we focus on night bus networks we often are faced with one certain node in the center of the city where all buses meet at particular times and star-shaped connections to the suburbs.

4.6 A linear path formulation

Due to the strict constraints for routes in a rail network the set of possible routes $\mathcal{R}$ is rather small. Hence a model that includes the routes explicitly (the edge formulation handles routes implicitly) is of valuable interest. Therefore, we introduce an integer vector $x \in \mathbb{Z}_+^{|\mathcal{R}|}$ where x_r represents the frequency of the line using route $r \in \mathcal{R}$. According to this particular representation of lines, the set of feasible line plans $\mathbb{L}$ can be described as follows.

$$\mathbb{L} := \{x \in \mathbb{Z}_+^{|\mathcal{R}|} \mid \underline{lfr}(e) \leqslant \sum_{\substack{r \in \mathcal{R} \\ e \in r}} x_r \leqslant \overline{lfr}(e) \text{ for all } e \in E\}. \tag{4.6}$$

Equations and inequalities containing the term $\sum_{r \in \mathcal{R},\, e \in r} x_e$ will occur frequently, therefore we introduce the $|E| \times |\mathcal{R}|$ *edge-route incidence matrix* $\mathfrak{A} = (a_{er})$ where

$$a_{er} = \begin{cases} 1 & \text{if } e \in r \\ 0 & \text{otherwise} \end{cases}$$

hence (4.6) reads as follows.

$$\mathbb{L} := \{x \in \mathbb{Z}_+^{|\mathcal{R}|} \mid \underline{lfr} \leqslant \mathfrak{A}x \leqslant \overline{lfr}\}$$

This powerful model of feasible line plans easily permits the inclusion of further operational constraints. For example, a line plan must contain some lines covering a sequence of stations v_1, v_2, v_2, v_4, e.g. in the center of the city in order to provide a condensation of frequencies to 2. The inequality

$$\sum_{\substack{r \in \mathcal{R} \\ \{v_1, v_2, v_3, v_4\} \in r}} x_r \geqslant 2 \tag{4.7}$$

excludes line plans from $\mathbb{L}$ where too many lines turn off at v_2 or v_3 (cf. figure 4.5).

In general, the set $\mathbb{L}$ contains a bunch of feasible line plans and it is not obvious which line plan should be picked out of $\mathbb{L}$. Optimization helps to overcome the question if the operating authority has a concrete objective in mind. If we can assign some value $c(L)$ to a feasible line plan $L \in \mathbb{L}$ that represents either the cost or the profit of line plan L, we may determine an *optimal* line plan L^* with respect to c, i.e.

$$L^* := \operatorname{argmin}\{c(L) \mid L \in \mathbb{L}\} \qquad \text{or} \qquad L^* := \operatorname{argmax}\{c(L) \mid L \in \mathbb{L}\}. \tag{4.8}$$

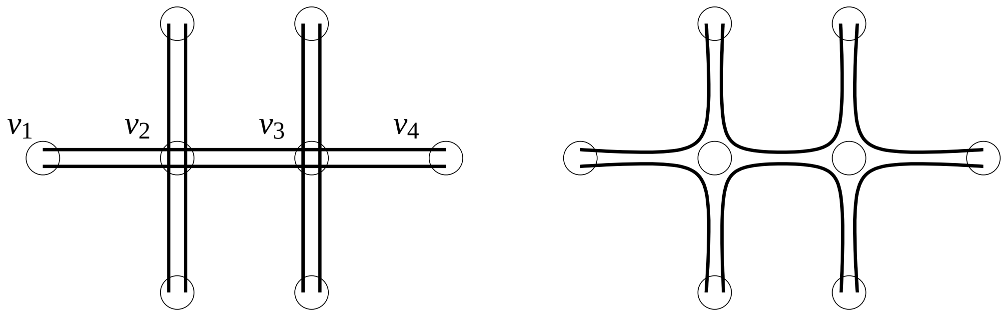


Figure 4.5: A feasible and an infeasible line plan with respect to (4.7)

Problem (4.8) is said to be a *combinatorial optimization problem*, if the *objective* c is linear, i.e. $c(L) = \sum_{l\in L} \tilde{c}(l)$ with $\tilde{c} : L \to \mathbb{R}$. Furthermore if $\tilde{c}(r,\varphi) = \varphi \cdot \tilde{c}(r,1)$ for each line $(r,\varphi) \in L$ we can formulate (4.8) as an *integer linear program* in the following way.

$$x^* := \begin{array}{rl} \text{argmin} & \tilde{c}^T x \\ \text{subject to} & \underline{lfr} \leqslant \mathfrak{A}x \leqslant \overline{lfr} \\ & x \in \mathbb{Z}_+^{|R|} \end{array} \quad \text{or} \quad x^* := \begin{array}{rl} \text{argmax} & \tilde{c}^T x \\ \text{subject to} & \underline{lfr} \leqslant \mathfrak{A}x \leqslant \overline{lfr} \\ & x \in \mathbb{Z}_+^{|R|} \end{array} \tag{4.9}$$

We refer to (4.9) as the *generic line planning problem.*

Due to the hardness results of section 4.5 it is quite improbable (unless $P{=}NP$) that there is an efficient, i.e. polynomial algorithm for (4.9). A fast computation of solutions of models similar to those in (4.9) is essential for our approach to the line planning problem and will be used throughout this thesis. Therefore, we will briefly discuss some well known algorithms for solving integer linear programs and their extensions in the next section.

4.7 Linear programming based branch-and-bound

In this section we discuss the solution approach of an integer linear program

$$z_{IP} = \max\{c^T x \mid \underbrace{Ax \leqslant b,\ x \in \mathbb{Z}_+^n}_{=:S}\}. \tag{4.10}$$

For rational data $A \in \mathbb{Z}^{m\times n}$ and $b \in \mathbb{Z}^m$ exactly one of three alternatives hold

1. (4.10) has an *optimal solution* x_{IP}, i.e. $c^T x_{IP} \geqslant c^T x$ for all $x \in S$.
2. (4.10) is *infeasible*, i.e. $S = \emptyset$.
3. (4.10) is *unbounded*, i.e. for all $\omega \in \mathbb{R}$ there is an $x \in S$ such that $c^T x > \omega$.

We use the notation $z_{IP} = -\infty$ ($z_{IP} = \infty$) for an infeasible (unbounded) instance of (4.10). It can be shown (cf. [52], I.5.4) that one can add constraints $x \leqslant \omega_{A,b}$ to any integer program in order to bound the optimal solution ($\max\{c^T x \mid Ax \leqslant b,\ x \leqslant \omega_{A,b}\} \leqslant 1^T |c| \omega_{A,b}$). Let $\tilde{x}$ be the optimal solution of the enlarged problem then the original instance is unbounded if and only if there is a variable $\tilde{x}_j$ with $((m+n)n\Theta)^n < \tilde{x}_j \leqslant \omega_{A,b}$, where $\Theta := \max_{\substack{i=1,\dots,n \\ j=1,\dots,m}} \{|a_{ij}|, |b_j|\}$. This observation permits us to restrict to instances of (4.10) where $z_{IP} < \infty$. Furthermore we rely on instances with rational data only.

An algorithm that solves (4.10) either produces a *feasible solution* $x_{IP} \in S$ and an upper bound w on the value of all feasible solutions $x \in S$ such that $c^T x_{IP} = w$ or decides that the given instance of (4.10) is infeasible ($z_{IP} = \Leftrightarrow\infty$). Many integer linear programming algorithms focus on the *dual step* by systematically reducing the upper bound but generally not producing an $x \in S$ until $w = z_{IP}$. *Relaxation algorithms* are of this type. At each iteration a relaxation of (4.10) is solved and if the relaxation does not yield an optimal solution of (4.10), the relaxation is refined. We discuss relaxation algorithms that use the *linear programming relaxation* (LP relaxation) of (4.10), i.e.

$$z_{LP} = \max\{c^T x \mid Ax \leqslant b,\ x \in \mathbb{R}^n_+\}.$$

The subsequent discussion requires the knowledge of basic concepts of *linear programming* and the *simplex method*. Novices in this field are referred e.g. to the book of CHVÁTAL [21]. Commercial codes for integer linear programs use the linear programming relaxation together with an enumerative approach. We say $\{S^i \mid i = 1,\dots,k\}$ is a *partition* of S if $\bigcup_{i=1}^k S^i = S$ and $S^i \cap S^j = \emptyset$ for $i, j = 1,\dots,k,\ i \neq j$. Let

$$z^i_{IP} = \max\{c^T x | x \in S^i\} \tag{4.11}$$

then $z_{IP} = \max_{i=1,\dots,k} z^i_{IP}$. This approach reflects the well known concept of *divide and conquer*. The partition is frequently done recursively as shown in the tree of figure 4.6. Here the *sons* of a given node represent a partition of the feasible region of their father, e.g. $S^{1_1}, S^{1_2}, S^{1_3}$ are sons of S^1. In order to prevent a total enumeration, which exhausts any computational resources for large scale problems, we must avoid partitioning S into too many subsets. If no further partition of a feasible region S^i is necessary, we say that the *enumeration tree* can be *pruned* at the corresponding node. Let

$$z^i_{LP} = \max\{c^T x \mid A^i x \leqslant b^i,\ x \in \mathbb{R}^n_+\} \tag{4.12}$$

be the linear programming relaxation of (4.11) with $S^i = \{A^i x \leqslant b^i,\ x \in \mathbb{Z}^n_+\}$.

PROPOSITION 4.5
The enumeration tree can be pruned at the node corresponding to S^i if any of the following three conditions holds:

1. *(4.12) is infeasible, i.e. $z^i_{LP} = \Leftrightarrow\infty$.*
2. *The optimal solution x^i_{LP} of (4.12) satisfies $x^i_{LP} \in S^i$.*
3. *$z^i_{LP} \leqslant \underline{z}_{IP}$, where $\underline{z}_{IP}$ is the value of a known feasible solution to (4.10). Note that if (4.12) is solved by the* dual simplex method, *we may prune once the value of the current* basic solution *is less than $\underline{z}_{IP}$.*

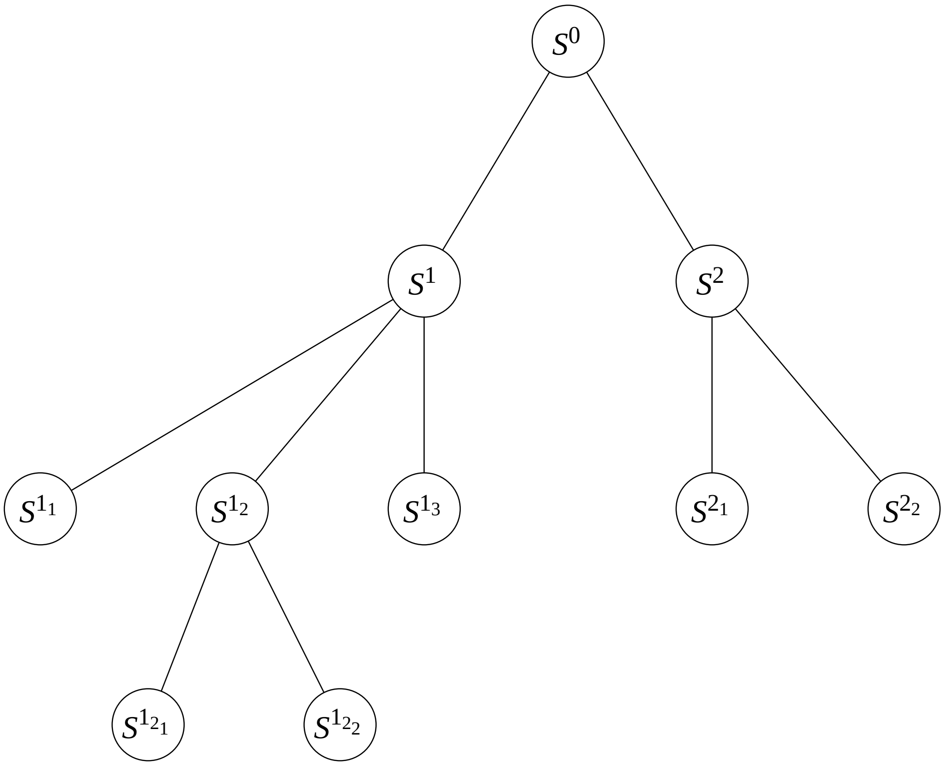


Figure 4.6: A branch-and-bound tree

The enumerative relaxation algorithm that uses linear programming relaxation and the rule of proposition 4.5 for pruning nodes is called linear programming based *branch-and-bound* (B&B) or *implicit enumeration.* We give an outline of the linear programming based branch-and-bound algorithm for solving (4.10). Γ represents a collection of integer programs (IP^i) each of which is of the form mentioned in (4.11). Associated with each problem in Γ is a (local) upper bound $\overline{z}^i_{IP} \geqslant z^i_{IP}$.

Step 1 (Initialization)
$\Gamma = \{\text{IP}^0\}$, $A^0 = A$, $b^0 = b$, $\overline{z}^0_{IP} = \infty$, $\underline{z}_{IP} = -\infty$.

Step 2 (Termination)
If $\Gamma = \emptyset$ then $\underline{z}_{IP}$ is the optimal value of (4.10). Note that $\underline{z}_{IP} = -\infty$ represents infeasibility of the instance.

Step 3 (Selection and Relaxation)
Select and delete a problem IP^i from Γ. Solve the corresponding linear programming relaxation (4.12). Let z^i_{LP} be the optimal value and x^i_{LP} be the optimal solution if one exists.

Step 4 (Pruning)

- if $z^i_{LP} \leqslant \underline{z}_{IP}$ go to step 2.
- if $x^i_{LP} \notin S^i$ go to step 5.
- else ($x^i_{LP} \in S^i$ and $z^i_{LP} > \underline{z}_{IP}$) let $\underline{z}_{IP} = z^i_{LP}$. Delete all problems with $\overline{z}^i_{IP} \leqslant \underline{z}_{IP}$ from Γ and go to step 2.

Step 5 (Partition)
Let $\{S^{i_j}\}_{j=1}^k$ be a partition of S^i. Add problems $\{\text{IP}^{i_j}\}_{j=1}^k$ to Γ, where $\overline{z}^{i_j}_{IP} = z^i_{LP}$. Go to step 2.

It remains to fill in the necessary details of how to select a node from Γ and how to perform a partition of S^i. But first of all, let us focus on *provable good solutions* even if the branch-and-bound algorithm is interrupted due to an exhaustive use of resources (time and space). The branch-and-bound algorithm described above solves an instance of (4.10) in *finite time*, but not even a rough estimation of the number of processed nodes and hence an estimation of the running time can be given a priori. The branch-and-bound algorithm must regularly terminate in order to prove the optimality of a solution (or the infeasibility of an instance). From the practical point of view a feasible solution of (4.10) and a guarantee that this solution is at most p% worse than an optimum solution is sufficient. The *general lower bound* $\underline{z}_{IP}$ together with the *global upper bound* $\overline{z}_{IP} = \max_{\text{IP}^i \in \Gamma} \overline{z}^i_{IP}$ leads to an *optimality gap* which monotonically decreases during the branch-and-bound procedure. We can terminate the algorithm if this gap becomes close enough or gives at least a *performance guarantee* $\frac{z_{IP} - \underline{z}_{IP}}{z_{IP}} \leqslant \frac{\overline{z}_{IP} - \underline{z}_{IP}}{\overline{z}_{IP}}$ (for the sake of simplicity assume $c(x) \geqslant 0$ for all $x \in S$ and $z_{IP} > 0$) of the current best solution of value $\underline{z}_{IP}$ if resources are exhausted.

4.7.1 Node selection

The *node selection* rule for branch-and-bound algorithms is related to enumeration strategies in graph search algorithms [32]. We define the *level* of a node corresponding to S^i as the number of predecessors in the branch-and-bound tree. The strategy where always the node with the largest (smallest) level is selected from Γ is known as the *depth-first-search* (*breadth-first-search*). The majority of commercial implementations use a rule named *best upper bound* as a default strategy. After a pruning step, choose a node from Γ with maximum upper bound $\bar{z}^i_{IP}$. A couple of node selection rules especially for integer linear programs arising from combinatorial optimization problems have been introduced, cf. [41] for some node selection investigations for the TSP.

4.7.2 Partition

Since we use a linear programming relaxation at each node, the partition is done by adding linear constraints. An obvious way to do this is to take $S^i = S^{i_1} \cup S^{i_2}$ with

$$S^{i_1} = S^i \cap \{x \in \mathbb{R}^n_+ \mid dx \leqslant d_0\} \qquad \text{and} \qquad S^{i_2} = S^i \cap \{x \in \mathbb{R}^n_+ \mid dx \geqslant d_0 + 1\}$$

where $(d, d_0) \in \mathbb{Z}^{n+1}$. If x^i_{LP} is the solution of the relaxation of (4.12), we choose (d, d_0) so that $d_0 < d^T x^i_{LP} < d_0 + 1$. This yields $x^i_{LP} \notin S^{i_1} \cup S^{i_2}$ and therefore gives the possibility that $\max\{z^{i1}_{IP}, z^{i2}_{IP}\} < z^i_{IP}$ which may result in an improved global upper bound $\bar{z}_{IP}$.

In practice, only a very special choice of (d, d_0) is used, where $d = e_j = (0, \ldots, 0, 1, 0, \ldots, 0)$ (1 at the jth position) for some $j = 1, \ldots, n$. The solution x^i_{LP} of the relaxation of (4.12) will be infeasible in the resulting relaxations if $x^i_{LP,j} \notin \mathbb{Z}_+$ and $d_0 = \lfloor x^i_{LP,j} \rfloor$. An important practical advantage of this partition is that only simple lower- and upper bound constraints are added to the linear programming relaxation, which can be handled implicitly by the simplex method without increasing the size of the basis.

Even if we restrict on partitioning arising from this strategy, named *variable dichotomy*, there is a variety of different possibilities due to a large number of fractional values in the optimal solution of a linear programming relaxation. Here is a brief itemization of *variable selection rules*.

- Maximum (minimum) infeasible selection: Choose a variable j that has a fractional value, i.e. $x_j \Leftrightarrow \lfloor x_j \rfloor$, closest to 0.5 (0 or 1).

- Strong branching rule: Select a set D of promising variables, e.g. by the maximum/minimum infeasible selection. Let $A^i x \leqslant b^i$ be the constraint set of the linear programming relaxation and x^i_{LP} the optimal solution. Solve for each variable $j \in D$ the two linear programs (or at least some iterations of the dual simplex method).

  $$z^j_d = \max\{c^T x \mid A^i x \leqslant b^i,\ x_j \leqslant \lfloor x^i_{LP,j} \rfloor\}, \qquad z^j_u = \max\{c^T x \mid A^i x \leqslant b^i,\ x_j \geqslant \lfloor x^i_{LP,j} \rfloor + 1\}$$

  Choose a variable x_{j^*}, $j^* \in D$ with

  $$j^* = \operatorname{argmin}_{j \in D}\{\alpha \max(z^j_d, z^j_u) + \min(z^j_d, z^j_u)\}$$

  with $\alpha \in \mathbb{Z}_+$ [10]. A slight modified description of the strong branching rule can be found in [74].

4.8 Improving the branch-and-bound algorithm

One crucial part of the branch-and-bound algorithm is the efficient solution of linear programs. Linear programs corresponding to relaxations after a partition step do not need to be solved from scratch. For example, with the simplex method the optimum basis of the linear programming relaxation of the father node is *dual feasible* and probably a good starting point for the dual simplex method. In the last decade a couple of fast implementations of the simplex algorithm as well as interior point methods have been introduced and currently merged into commercial products like *CPLEX*, *OSL*, and *XPRESS-MP*[2]

Another substantial part of the branch-and-bound algorithm is the size of the branch-and-bound tree which is regulated by partitioning and pruning. Remember that the pruning criterion of *value dominance* can be applied if $z^i_{LP} \leqslant \underline{z}_{IP}$. This condition can be fulfilled more frequently if good lower bounds, i.e. feasible solutions, and improved relaxations are available. Furthermore a predefined performance guarantee of ε supports a relaxation of the value dominance criterion, i.e. $(1+\varepsilon)z^i_{LP} \leqslant \underline{z}_{IP}$.

Heuristics play an important role in the generation of good feasible solutions. The field of heuristics covers some well known approaches to hard combinatorial problems. *Meta heuristics*, including *simulated annealing*, *tabu search*, and *genetic algorithms* can be applied to almost every problem (with more or less success). *Rounding heuristics* try to find an integer solution starting from an optimal solution of the linear programming relaxation. Furthermore, we have a bunch of *problem specific heuristics*, e.g. [45] contains some heuristics for the TSP.

Methods from polyhedral optimization together with thorough *preprocessing* and *probing* result in a tighter linear programming relaxation with an increased value z^i_{LP}. In the subsequent section we discuss the main aspects of improving the linear programming relaxation in detail.

Beyond the pruning conditions the size of the branch-and-bound tree substantially depends on the strategy of the different selections of the algorithm. One effective variable selection scheme is based on *priority orders*. While solving an integer linear program that has variables representing different types of decisions the overall solution progress may depend on which type of variables is chosen for branching first. We assign a *branching priority* to each variable before starting the branch-and-bound algorithm. Variables with higher priorities will be selected and branched upon in the branch-and-bound tree before variables with lower priorities will. Some integer linear programming formulations provide *special ordered sets* (SOSs), i.e. a set of variables $\{x_i \mid i = 1, \ldots, k\}$ for which at most *one* variable may be non-zero in a feasible solution. The partition step of the branch-and-bound algorithm may take advantage of these sets by partitioning the feasible region using the inequality $\sum_{i=1,\ldots,k} x_i \leqslant 0$ instead of a variable dichotomy.

In the remaining part of this section we discuss some aspects of *proper modeling*. For combinatorial and other problems there is not necessarily a *canonical* integer linear program formulation. In general, there is a bunch of formulations representing the same problem but providing linear programming relaxations of significantly different quality (cf. [29, 59] for the maximum clique problem). Furthermore, some formulations *hide* information of the problem which ag-

[2] A list of commercial as well as public domain linear programming solvers is contained in the linear programming frequently asked questions at URL
`http://www.mcs.anl.gov/home/otc/Guide/faq/linear-programming-faq.html`

gravate automatic preprocessing. For example, the objective $c^T x$ of $\max\{c^T x \mid Ax \leqslant b, x \in \mathbb{Z}^n_+\}$ could be hidden in the constraint set by $\max\{z \mid Ax \leqslant b, c^T x = z, x \in \mathbb{Z}^n_+, z \in \mathbb{R}\}$. Proper model formulation is a key ingredient for solving hard combinatorial problems. The reader is referred to the book of WILLIAMS [81].

Even if we choose the best formulation, increase the linear programming relaxation value, and add problem specific information in order to improve the selection schemes, the linear programming based branch-and-bound method may completely fail while other relaxation methods yield good solutions. For example, the *semidefinite programming relaxation* for graph partitioning problems like MAX CUT [35], seems to be more suitable than linear programming relaxations [44]. The branch-and-bound algorithm and especially the linear programming based branch-and-bound method represent *one* particular approach to solve hard combinatorial problems.

4.9 Improving the linear programming relaxation

We review two major approaches for tightening the linear programming relaxation of an integer linear program. *Preprocessing* and *probing* techniques [65, 28] focus on the *reformulation* of the initial constraint set by fixing variables, identifying redundancy and improving bounds and coefficients. In contrast to the reformulation of existing parts of the formulation, *constraint generation* techniques [76, 26] try to generate *new* inequalities in order to elaborate a tighter formulation.

4.9.1 Preprocessing and probing

Consider the integer linear program (4.10) with additional lower and upper bounds $l \leqslant x \leqslant u$. We analyze each of the inequalities trying to establish some results which yield a tighter formulation. Assume that the inequality under consideration $a^{iT} x \leqslant b_i$ is of the form

$$\sum_{j \in I^+} a^i_j x_j - \sum_{j \in I^-} a^i_j x_j \leqslant b_i$$

with $I^+ = \{j \mid a^i_j > 0\}$ and $I^- = \{j \mid a^i_j < 0\}$. Furthermore, let $A^i x \leqslant b^i$ denote the system of inequalities obtained from $Ax \leqslant b$ by deleting row $a^{iT} x \leqslant b_i$.

Infeasibility: It is obvious that the feasible region $\{x \mid Ax \leqslant b, l \leqslant x \leqslant u, x \in \mathbb{Z}^n_+\}$ is empty if the optimum value z of

$$z = \min\{\sum_{j \in I^+} a^i_j x_j - \sum_{j \in I^-} a^i_j x_j \mid A^i x \leqslant b^i, l \leqslant x \leqslant u, x \in \mathbb{Z}^n_+\} \tag{4.13}$$

exceeds b_i. Unfortunately, the integer linear program (4.13) is as hard to solve as the original problem (4.10), but any lower bound $\underline{z}$ of (4.13) that fulfills $\underline{z} > b_i$ also indicates the infeasibility of (4.10). For example, the linear programming relaxation of (4.13) would provide such a lower bound but its computation might be too expensive. Therefore we construct another lower bound by completely disregarding the constraints $A^i x \leqslant b^i$ and conclude that (4.10) is infeasible if

$$\underline{z} = \sum_{j \in I^+} a^i_j l_j \Leftrightarrow \sum_{j \in I^-} a^i_j u_j > b_i.$$

Redundancy: The constraint $a^{iT} x \leqslant b_i$ is redundant and can be eliminated from the formulation if the optimum value z of

$$z = \max\{ \sum_{j \in I^+} a^i_j x_j \Leftrightarrow \sum_{j \in I^-} a^i_j x_j \mid A^i x \leqslant b^i, l \leqslant x \leqslant u, x \in \mathbb{Z}^n_+\} \tag{4.14}$$

does not exceed b_i. Similarly to the infeasible case, an upper bound $\overline{z}$ of (4.14) with $z \leqslant \overline{z} \leqslant b_i$ is sufficient. Therefore $a^{iT} x \leqslant b_i$ is redundant if

$$\overline{z} = \sum_{j \in I^+} a^i_j u_j \Leftrightarrow \sum_{j \in I^-} a^i_j l_j \leqslant b_i.$$

Improving bounds: Consider a variable $k \in I^+$ and the following integer linear program.

$$z_k = \min\{ \sum_{j \in I^+ \setminus \{k\}} a^i_j x_j \Leftrightarrow \sum_{j \in I^-} a^i_j x_j \mid A^i x \leqslant b^i, l \leqslant x \leqslant u, x \in \mathbb{Z}^n_+\}$$

Clearly, $x_k \leqslant \lfloor (b_i \Leftrightarrow z_k)/a^i_k \rfloor$ for each $x \in S$ and hence by disregarding $A^i x \leqslant b^i$ we derive a new upper bound

$$u'_k = \min\{u_k, \lfloor (b_i \Leftrightarrow (\sum_{j \in I^+ \setminus \{k\}} a^i_j l_j \Leftrightarrow \sum_{j \in I^-} a^i_j u_j))/a^i_k \rfloor \}.$$

Carrying out a similar procedure we obtain a new lower bound for a variable $k \in I^-$.

$$l'_k = \max\{l_k, \lceil ((\sum_{j \in I^+} a^i_j u_j \Leftrightarrow \sum_{j \in I^- \setminus \{k\}} a^i_j l_j) \Leftrightarrow b_i)/a^i_k \rceil \}.$$

Fixing of variables: Consider a variable $k \in I^+$ and the integer linear program

$$z_k = \min\{ \sum_{j \in I^+} a^i_j x_j \Leftrightarrow \sum_{j \in I^-} a^i_j x_j \mid A^i x \leqslant b^i, l \leqslant x \leqslant u, l_k + 1 \leqslant x_k, x \in \mathbb{Z}^n_+\}.$$

If $z_k > b_i$ then $x_k = l_k$ for each $x \in S$. Consequently, we can fix x_k to l_k if

$$a^i_k (l_k + 1) + \sum_{j \in I^+ \setminus \{k\}} a^i_j l_j \Leftrightarrow \sum_{j \in I^-} a^i_j u_j > b_i.$$

Similar, for $k \in I^-$ we can fix x_k to u_k if

$$\sum_{j \in I^+} a^i_j l_j \Leftrightarrow a^i_k (u_k \Leftrightarrow 1) \Leftrightarrow \sum_{j \in I^- \setminus \{k\}} a^i_j u_j > b_i.$$

Especially for integer linear programs with tight bounds l and u and in particular for *binary linear programs*, i.e. $l \equiv 0$ and $u \equiv 1$, the procedure of fixing variables has shown to be very effective. For binary linear programs there is another effective probing method based on *logical implications*. Consider two variables $k_1, k_2 \in I$ and the following binary linear program.

$$z = \min\{ \sum_{j \in I^+} a^i_j x_j \Leftrightarrow \sum_{j \in I^-} a^i_j x_j \mid A^i x \leqslant b^i, x_{k_1} = 1, x_{k_2} = 1, 0 \leqslant x \leqslant 1, x \in \mathbb{Z}^n_+ \}$$

If $z > b_i$, or a lower bound $\underline{z}$ on z provides $z \geqslant \underline{z} > b_i$ then we have the following logical implications.

$$x_{k_1} = 1 \Rightarrow x_{k_2} = 0 \quad \text{and} \quad x_{k_2} = 1 \Rightarrow x_{k_1} = 0$$

After partitioning in the branch-and-bound algorithm we can take advantage of these implications. Furthermore, they provide the generation of particular inequalities which are presented in the next section.

The preprocessing and probing techniques described above do not take the objective into account. The following fixing of variables results in a formulation that is not necessarily a relaxation of S, because we exclude some elements of S which can never be an optimal solution. Assume that the integer linear program (4.10) provides a special ordered set $\{x_j \mid j = 1, \ldots, k\}$. Furthermore, let A_{i_1} and A_{i_2} be the columns of the constraint matrix corresponding to variables i_1, i_2 belonging to the special ordered set. If $A_{i_1} \leqslant A_{i_2}$ and $c_{i_1} \geqslant c_{i_2}$ then x_{i_2} can be fixed to 0 because there is always an optimal solution x^* of (4.10) with $x^*_{i_2} = 0$.

4.9.2 Constraint generation

For integer linear programs some general classes of valid inequalities or cuts are known from the literature. A GOMORY cut [36] is constructed with respect to a fractional optimal solution of the corresponding linear programming relaxation. The successive addition of GOMORY cuts is called the GOMORY *fractional cutting plane algorithm* and yields an optimal solution to (4.10) in a finite number of iterations. Hence the enumeration part of the branch-and-bound algorithm can be skipped. In spite of the finite convergence the practical use of GOMORY cuts is quite uncertain. Especially for integer linear programs arising from combinatorial optimization problems a pure GOMORY fractional cutting plane algorithm has been shown to be rather weak [6].

For binary linear programs there is a bunch of useful valid inequalities which have been introduced in the literature. We briefly discuss the classes of *clique* and *cover* inequalities.

In section 4.9.1 we introduced logical implications $x_i = \alpha \Rightarrow x_j = \beta$ with $\alpha, \beta \in 0, 1$. Each of the four combinations of logical implications can be written in the form $\tilde{x}_i = 1 \Rightarrow \tilde{x}_j = 0$, where $\tilde{x}_i \in \{x_i, \overline{x_i} = 1 \Leftrightarrow x_i\}$, $\tilde{x}_j \in \{x_j, \overline{x_j}\}$. Note that $\tilde{x}_i = 1 \Rightarrow \tilde{x}_j = 0$ implies $\tilde{x}_j = 1 \Rightarrow \tilde{x}_i = 0$ because

$x \in \{0,1\}^n$. Hence a logical implication identifies two variables (original or complemented), that cannot be 1 at the same time in any feasible solution resulting in a valid inequality of the form $\tilde{x}_i + \tilde{x}_j \leqslant 1$. Logical implications joined in sequence may yield a larger set of variables that cannot be 1 at the same time. Therefore, we construct a graph $G_L = (V,E)$ where V contains nodes representing original and complemented variables. The edges in E represent the logical implications and link two nodes $v_{\tilde{x}_i}$, $v_{\tilde{x}_j}$ if and only if both variables cannot be 1 in any feasible solution. Any *clique* C in G_L, i.e. a set of nodes C where $uv \in E$ for all $u,v \in V$, $u \neq v$, represents a set of original variables X_C or complemented variables $\overline{X_C}$ that cannot be simultaneously 1 in any feasible solution. From clique C we derive the following *clique cut*.

$$\sum_{x_i \in X_C} x_i + \sum_{\overline{x_i} \in \overline{X_C}} \overline{x_i} \leqslant 1 \Leftrightarrow \sum_{x_i \in X_C} x_i \Leftrightarrow \sum_{\overline{x_i} \in \overline{X_C}} x_i \leqslant 1 \Leftrightarrow |\overline{X_C}|$$

Due to the large number of cliques in G_L *violated clique cuts*, i.e. $\sum_{x_i \in X_C} x_i^* + \sum_{\overline{x_i} \in \overline{X_C}} \overline{x_i^*} > 1$ corresponding to a fractional optimal solution x^* of the linear programming relaxation should be added to the constraint set of the formulation only. Let x^* be the optimal solution of the linear programming relaxation. Assume G_L contains all logical implications, then there is a violated clique cut if and only if G_L contains a clique C of weight $\sum_{x_i \in X_C} x_i^* + \sum_{\overline{x_i} \in \overline{X_C}} (1 \Leftrightarrow x_i^*) > 1$. This recognition problem, known as separation of clique cuts, turns out to be $\mathcal{NP}$-complete, hence for practical use of clique cuts the separation will be based on heuristics.

The problem

$$\max\{c^T x \mid a^T x \leqslant b, x \in \{0,1\}^n\} \tag{4.15}$$

with $a \geqslant 0$ is the binary linear formulation of a *0-1 knapsack problem.* The constraint set of the linear description of (4.15) has been widely studied in the literature (cf. [80] as a starting point). Due to complemented variables we can rewrite every individual constraint of a general binary linear program

$$\max\{c^T x \mid Ax \leqslant b, x \in \{0,1\}^n\} \tag{4.16}$$

using the notation of section 4.9.1.

$$\sum_{i \in I^+} |a^i_j| \tilde{x}_j \leqslant b_i \Leftrightarrow \sum_{i \in I^-} a^i_j$$

where $\tilde{x}_j = x_j$ if $a^i_j > 0$ and $\tilde{x}_j = \overline{x_j}$ if $a^i_j < 0$. Every cut of the corresponding 0-1 knapsack problem represents a cut of (4.16) and can be added to the constraint set after reformulation using the original variables.

The set of *cover inequalities* has been efficiently applied to general binary linear programs. Let $a^T x \leqslant b$, $a \geqslant 0$ be the constraint of the 0-1 knapsack problem. A minimal set of indices C is a *cover* if $\sum_{j \in C} a_j > b$. We can derive the following valid inequality from a cover C.

$$\sum_{j \in C} x_j \leqslant |C| \Leftrightarrow 1 \tag{4.17}$$

The separation of cover cuts results in the solution of the following problem. Given the fractional solution x^* of the linear programming relaxation. We want to find a cover C with $\sum_{j \in C} a_j > b$ and $\sum_{j \in C} x_j^* > |C| \Leftrightarrow 1$. Introducing a vector $z \in \{0,1\}^n$ to represent the unknown set C, we attempt to choose z such that

$$\sum_{j=1}^{n} a_j z_j > b \quad \text{and} \quad \sum_{j=1}^{n} x_j^* z_j > \sum_{j=1}^{n} z_j \Leftrightarrow 1 \quad \Leftrightarrow \quad \sum_{j=1}^{n} a_j z_j \geqslant b+1 \quad \text{and} \quad \sum_{j=1}^{n} (1 \Leftrightarrow x_j^*) z_j < 1.$$

We find a violated cover inequality if the optimal solution value of

$$\min\{\sum_{j=1}^{n} (1 \Leftrightarrow x_j^*) z_j \mid \sum_{j=1}^{n} a_j z_j \geqslant b+1, z \in \{0,1\}^n\} \tag{4.18}$$

is less than 1. The binary linear program (4.18) can be easily identified as a 0-1 knapsack problem. The corresponding recognition problem is $\mathcal{NP}$-complete but can be solved in pseudo polynomial-time by *dynamic programming*.

Constraints can be generated whenever a linear programming relaxation is solved. The generation strategy determines the *type* of the generic linear programming based branch-and-bound algorithm.

- Pure branch-and-bound: No constraint generation.
- Cutting plane: Exhaustive cut generation at the root node of the branch-and-bound tree. No branching is necessary.
- Cut-and-branch: Cut generation only/mainly at the root node. No/minor cut generation at other nodes of the branch-and-bound tree.
- Branch-and-cut: Cut generation at all nodes.

A particular solution method for solving linear programs with a huge number of variables, known as *delayed column generation* [21], extends this listing of linear programming based branch-and-bound algorithms by

- Branch-and-price.

The simplex method with delayed column generation is based on the fact that most of the variables have a value of zero in a basic solution. These “superfluous” variables are left out from the formulation and will be generated on demand. For a current basis the optimality test of the simplex method requires the solution of the *pricing problem*. The pricing consists of identifying a variable, which may be not explicitly included in the linear program, that enters the basis to improve the objective value. Instead of enumerating the columns and computing its reduced cost, we may solve the pricing problem by an alternative optimization problem. For some problems with a huge number of variables, arising from DANZIG-WOLFE decomposition (e.g.

multi-commodity flows [1]) or from a particular problem formulation (e.g. cutting stock problem [21]), the pricing problem provides a structure of well known combinatorial optimization problems (shortest path, minimum-cost flow, knapsack, ...).

In principle, the solution of the linear programs in the branch-and-bound procedure can be achieved by delayed column generation. The only problem we encounter in this approach is the handling of new constraints resulting from problem partitioning in the pricing problem. Usually, the solution of the pricing problem is based on the particular structure of the constraint matrix. In general, the addition of variable dichotomy constraints destroys this structure. Hence, we cannot apply the solution method for the pricing problem associated to the linear programming relaxation for the problems of nodes inside the branch-and-bound tree different from the root node. Furthermore, the advanced preprocessing techniques which may substantially change the original constraint matrix must be carefully applied in combination with branch-and-price algorithms. Nevertheless, with a partition rule that cooperates with the pricing algorithm or preserves the structure of the constraint matrix, also known as *compatible branching rule*, the branch-and-price approach, also known as *branch-and-cut-and-price* when constraint generation is enabled, can be successfully applied to integer linear programs with a tremendous number of variables (cf. [77, 78]).

Depending on the particular type of the algorithm, several implementation details which significantly influence the overall performance of the solution process must be valuated. For example the trade-off between branching and cutting must be considered. A set of fast branching steps may yield a better solution process than expensive generation of weak cuts and vice versa. Local cuts, i.e. inequalities that are only valid for the feasible region corresponding to the current branch-and-bound node, must be distinguished from global valid inequalities. Furthermore inactive cuts, i.e. $\pi^T x < \pi_0$ for all points x of the relaxation must be eliminated from the formulation in order to keep the size of the problem manageable.

General preprocessing and probing techniques as well as general cuts have been applied with moderate success to integer linear programs. Problem specific preprocessing and cuts are vast superior to general techniques but their identification requires a lot of creative mathematical work.

Before closing this chapter we would like to mention that most of the techniques and results presented for integer linear programs can be transfered to *mixed integer linear programs*, i.e.

$$\max\{c^T x + h^T y \mid Ax + Gy \leqslant b,\ x \in \mathbb{Z}^n_+,\ y \in \mathbb{R}_+\}.$$

Chapter 5

Line planning with respect to direct travelers

5.1 Introduction

In chapter 5 and 6 we concentrate on particular objectives of the line planning problem. In the previous chapter we presented a first optimization model introduced as the generic line planning problem in which we could directly associate a certain cost or benefit to a single line that linearly depends on the frequency of the line. However, for the objectives provided by the practitioners we need a more powerful modeling of the line planning problem.

The marketing departments of the railroad companies focus on a *product* with a certain level of quality at reasonable cost. The quality of the "product" railroad is given besides safe and convenient trains by the quality of the line plan and the train schedule. This quality is mainly based on short travel times for the customers. At this early stage of planning there is no train schedule, hence we cannot determine the exact waiting time while changing lines. Changing of lines itself is a major inconvenience, hence one possible way of estimating the quality of a line plan is to focus on the number of necessary *changes*. But even for a line plan that gives a total minimum number of changes there may be some passengers with unacceptable large number of changes. Furthermore, the train schedule may provide two connections for one origin-destination pair of different quality. For example, the first connection requires two changes with short waiting times while the second requires a single change with a long waiting period. Some travelers, e.g. passengers with much luggage, may prefer the latter alternative, but other travelers will use the first alternative because it will result in a reduced total travel time. This variation in the behavior of the passengers produces a significantly deviation in the estimation of the quality. If the line plan contains lines that provide a *direct connection* on a short route for the passengers of a particular origin-destination pair, all these travelers will use this direct connection. Hence one certain way of estimating the quality of a line plan is to compute the number of *direct travelers*. We will present a formulation of the line planning problem with respect to the number of direct travelers in section 5.2. DIENST suggests a branch-and-bound algorithm for a relaxed version of this problem which is described in section 5.3. In section 5.4 we give a new formulation based on an integer linear program of tremendous size which can be significantly reduced by relaxing some capacity constraints (section 5.5). With a problem specific preprocessing and some strong valid inequalities we can tighten the formulation and succeed in solving the relaxed problem within a reasonable amount of computation time. Moreover, this solution provides a performance guarantee for the initial formulation. Section 5.6 summarizes the polyhedral background of the cutting planes applied to the integer linear program of section 5.5. Finally, we present some potential extensions of the direct traveler approach in section 5.8.

5.2 Problem description

Similar to the generic line planning problem, the direct traveler approach requires some information about the infrastructure of the supply network given by an undirected graph $G = (V, E)$ and a set R of possible routes. Throughout the thesis we concentrate on periodic transportation systems, hence all data is based on a basic time interval $[0, \dots, \tau)$. Therefore, we can make use of the concept of the line frequency requirement (cf. chapter 4) which provides lower and upper bounds $\underline{lfr}$, $\overline{lfr}$ for the number of trains for each edge $e \in E$. Hence, the frequency of a particular line on route r is bounded by $\varphi^r_{\max} = \min_{e \in r} \overline{lfr}(e)$. Again, a feasible line plan consists of a set of routes with corresponding frequencies that fulfill the line frequency requirement. The valuation of a line plan with respect to the number of direct travelers also requires some information about the customers of the transportation system. The volume of traffic of the supply network, computed by the system split procedure (cf. section 3.2), is given by an origin-destination matrix $T \in \mathbb{Z}_+^{|V| \times |V|}$, i.e. $T^{a,b}$ denotes the number of travelers commuting between nodes a and b. Note that the system split procedure also computes the traffic load ld for all edges $e \in E$.

Let $V_T^2 \subset \{(a,b) \mid a, b \in V\}$ be the set of node pairs with $T^{a,b} > 0$. For each origin-destination pair $a, b \in V_T^2$, we have a subset $R_{a,b} \subset R$ of routes that provides the passengers commuting between a and b with a pleasant travel path. The notion *pleasant* depends on the network and the origin-destination pair, but a pleasant travel path necessarily contains nodes a and b (cf. the discussion of pleasant travel path in chapter 4). Furthermore, the trains of the supply network have a fixed capacity C (cf. section 3.1). We can determine the valuation of a given line plan with respect to the number of direct travelers by solving a complex integer multi-commodity flow problem. Hence, the problem of establishing a feasible line plan that maximizes the number of direct travelers subject to the capacity of the lines, is related to a particular *integer multi-commodity flow network design problem* (cf. [9] for an introduction to edge oriented multi-commodity flow design problems).

	nsic	nsir	nsar	dbagic	dbagir	sbb1	sbb2	sbb3
$\lvert V \rvert$	23	86	385	100	307	144	78	57
$\lvert E \rvert$	31	114	428	118	398	168	87	62
$\lvert V_T^2 \rvert$	210	2147	11240	3136	9215	1968	447	447

Table 5.1: Reference numbers of the railroad instances

	bvagtram	bvagbus	vbzsbahn	vbztram	vbzbus
$\|V\|$	70	335	53	149	199
$\|E\|$	72	357	63	160	233
$\|V_T^2\|$	1307	12728	941	4484	7704

Table 5.2: Reference numbers of the urban public transport instances

The models for line planning described in this chapter rely on the assumption that the lower bound for the line frequency requirement combined with the train capacity C provides sufficient capacity for the total volume of traffic, i.e. $\underline{lfr}(e) \cdot C \geqslant ld(e)$. Therefore, we concentrate on the capacity constraint for direct travelers, only. Nevertheless, these capacity constraints are most important as we will see in the following example. The line plan depicted in figure 5.1 provides a sufficient capacity for all travelers of origin-destination pair a,c but only 50 of these travelers have a direct connection. Depending on the type of relaxation of the capacity contraints for the direct travelers, we obtain several models of different quality.

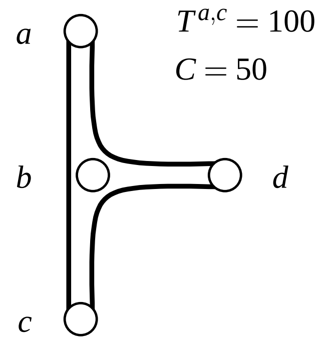


Figure 5.1: Capacity constraint for direct travelers

Before going into the details of the various models, we give a brief description of our data pool. We collected real world data from three different European railroad companies. We have three supply networks from Nederlandse Spoorwegen (ns), the Dutch railroad company, including an InterCity, an InterRegio, and an AggloRegio network (cf. figure 3.2). From the Deutsche Bahn AG (dbag), the German railroad company, we gathered an InterCity and an InterRegio network (cf. figure 5.2). Finally, we have collected three long-distance instances from the Schweizer Bundesbahnen (sbb), the Swiss railroad company. Furthermore, we are aware of several networks arising from urban public transport systems. We have data from the cities of Braunschweig (Braunschweiger Verkehrs AG, bvag) and Zürich (Verkehrsbetriebe Zürich, vbz). The latter data set consists of a fast train system (S-Bahn), a tram and a bus network. The instances of the city of Braunschweig include a tram and a bus network. Tables 5.1 and 5.2 summarize the reference numbers of the various instances.

5.3 A branch-and-bound algorithm

DIENST [27] proposes a branch-and-bound algorithm for the line planning problem with respect to the number of direct travelers based on the following simplification of the original problem. First of all, DIENST assumes an infinite train capacity. Therefore, if the line plan $L = \{(r,\varphi) \mid r \in R' \subset R, \varphi \in \{1,\dots,\varphi^r_{\max}\}\}$ contains a line (r,φ) with $r \in R_{a,b}$, all passengers of the origin-destination pair $a,b \in V_T^2$ are provided with a direct connection. Hence, we can easily compute

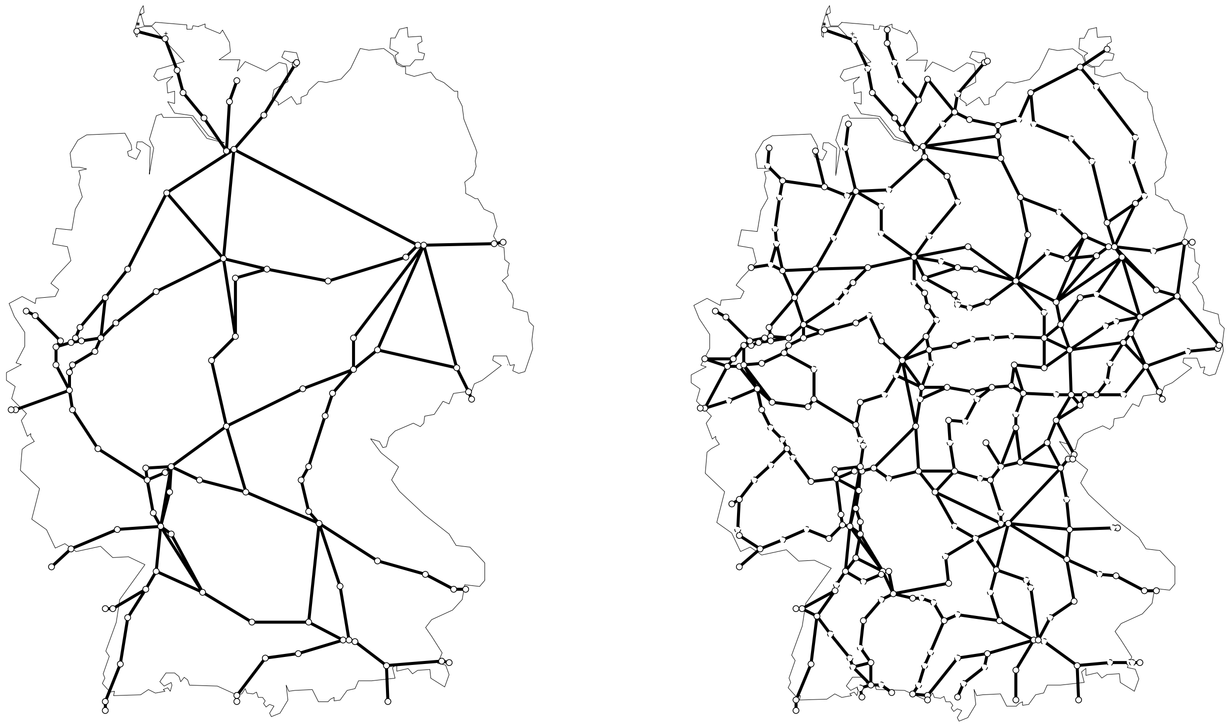


Figure 5.2: The supply networks of the instances `dbagic` and `dbagir`

the number of direct travelers D_L^* of a line plan L as follows.

$$D_L^* = \sum_{\substack{a,b\in V_T^2 \\ R'\cap R_{a,b}\neq\emptyset}} T^{a,b} \tag{5.1}$$

Furthermore, the maximal frequency of a line $\varphi_{\max}$ is limited to 1 and additionally the lower bound for the line frequency requirement $\underline{lfr}$ is firstly omitted to overcome infeasibility problems. In a subsequent procedure DIENST tries to satisfy the line frequency requirement of violated edges $|\{r|r\in R', e\in r\}| < \underline{lfr}(e)$ by adding lines. If the set of possible routes contains routes on single edges $r = e$ for all $e \in E$ this can easily be done, but in general this problem is as hard as finding a feasible line plan itself (cf. theorem 4.2).

The type of *relaxation* used within the branch-and-bound algorithm is based on the following fact. Consider a (partial) line plan $L' = \{(r,1) \mid r \in R'\}$. Note that L' is uniquely determined by R', hence we omit the frequency in the subsequent discussion.

If the lines in R' satisfy the upper bound for the line frequency requirement $\overline{lfr}$ of an edge $e \in E$ with equality, there is no other line containing e in any line plan including R'. Furthermore, if all pleasant travel paths of an origin-destination pair include edge e but R' does not provide a pleasant direct connection for origin-destination pair $a,b \in V_T^2$, there is no direct connection for these $T^{a,b}$ travelers in any line plan including R'. Hence

$$\overline{D}_{R'} = \sum_{a,b\in V_T^2} T^{a,b} \Leftrightarrow \sum_{a,b\in I_{R'}} T^{a,b} \tag{5.2}$$

with

$$I_{R'} = \left\{ (a,b) \in V_T^2 \;\middle|\; R' \cap R_{a,b} = \emptyset, \exists e \in E : |\{r \in R' | e \in r\}| = \overline{lfr}(e), \forall r \in R_{a,b} : e \in r \right\}$$

represents an upper bound for the number of direct travelers of any line plan including R'. With this relaxation we can establish a branch-and-bound algorithm, which is quite similar to the branch-and-bound method based on a linear programming relaxation (cf. section 4.7).

Step 1 (Initialization)
$\Gamma = \{\mathrm{P}^0\}$, $R^0 = \emptyset$, $\overline{R}^0 = R$, $\overline{z}_P^0 = \sum_{a,b \in V_T^2} T^{a,b}$, $\underline{z}_P = 0$.

Step 2 (Termination)
If $\Gamma = \emptyset$ then $\underline{z}_P$ is the maximum number of direct travelers of any feasible line plan.

Step 3 (Selection and Relaxation)
Select and delete a problem P^i from Γ. Compute the upper bound $\overline{z}_P^i$ for the number of direct travelers in a line plan including R^i using (5.2) and the number of direct travelers $\underline{z}_P^i$ in R^i applying (5.1).

Step 4 (Pruning)

- if $\overline{z}_P^i \leqslant \underline{z}_P$ go to step 2.
- if $\underline{z}_P^i > \underline{z}_P$ let $\underline{z}_P = \underline{z}_P^i$. Delete all problems with $\overline{z}_P^i \leqslant \underline{z}_P$ from Γ and go to step 2.

Step 5 (Partition)
Select a line $r^* \in \overline{R}^i \setminus R^i$ with $\overline{lfr}(e) > |\{r \in R^i \mid e \in r\}|$ for all $e \in r^*$. Add problems $R^{i_1} = R^i \cup \{r^*\}$, $\overline{R}^{i_1} = \overline{R}^i \setminus \{r^*\}$ and $R^{i_2} = R^i$, $\overline{R}^{i_2} = \overline{R}^i \setminus \{r^*\}$ with upper bounds $\overline{z}_P^{i,1,2} = \overline{z}_P^i$ to Γ. Go to step 2.

In an ancient ALGOL/FORTRAN implementation of this branch-and-bound algorithm the selection of the *branching route* r^* in step 5 is done by a greedy choice. A route is chosen that maximizes the gain with respect to $\underline{z}_P^i$, i.e. the number of direct travelers in R^i. KREUZER [42] applies this implementation to the instances of Nederlandse Spoorwegen (cf. table 5.3). He limits the number of branch-and-bound nodes to 10000. For the InterCity network `nsic` only, DIENST's implementation regularly terminates with an optimal solution. For the InterRegio `nsir` and the AggloRegio `nsar` network the performance guarantee (gap) provided after 10000 nodes is acceptable, but remember the large computation times[1] and the remarkable relaxations of the capacity constraints.

5.4 A revised direct traveler approach

The results provided by the approach of DIENST are quite unsatisfactory with respect to the level of relaxation and the large computation times. We present a model for the line planning problem

[1] CPU seconds on an HP 720 workstation.

	nsic	nsir	nsar
gap	0.0%	1.9%	4.1%
CPU seconds[1]	30	5400	39600

Table 5.3: Computational results of DIENST's implementation of the B&B algorithm

with the particular objective of direct travelers which is based on an integer linear programming formulation. The fundamentals of this approach, which can be found in [12] as well, consist of a combination of the generic line planning problem and a particular multi-commodity flow formulation of the direct traveler valuation.

Let us focus on the latter part. Consider a feasible line plan $L = \{(r,\varphi) \mid r \in \mathcal{R}' \subset \mathcal{R}, \varphi \in \{1,\ldots,\varphi^r_{\max}\}\}$ and the volume of traffic given by an origin-destination matrix T. Furthermore, suppose that we are aware of the pleasant routes $\mathcal{R}_{a,b} \subset \mathcal{R}$ for each origin-destination pair $a,b \in V_T^2$ and of the train capacity C. We already mentioned in section 5.2 that we concentrate on capacity issues of direct travelers, only. Consider a line $(r,\varphi) \in L$. This line can be used for a direct connection by travelers of origin-destination pairs $a,b \in V_T^2$ with $r \in \mathcal{R}_{a,b}$. We introduce a variable $y_{r,a,b} \in \mathbb{Z}_+$, which denotes the number of direct travelers of origin-destination pair $a,b \in V_T^2$ in line $l = (r,\varphi)$ on route r. Travelers of origin-destination pair a,b use a particular section $r_{a,b}$ of r and therefore, contribute to the load of line (r,φ) on the edges $e \in r_{a,b}$, only.

The number of direct travelers of origin-destination pair $a,b \in V_T^2$ is trivially subject to

$$\sum_{r \in \mathcal{R}_{a,b}} y_{r,a,b} \leqslant T^{a,b}. \tag{5.3}$$

Furthermore, we must obey the line capacity of $l = (r,\varphi)$ given by $C \cdot \varphi$ for each edge $e \in r$. Consider edge $e \in r$ and all origin-destination pairs $a,b \in V_T^2$ with $r \in \mathcal{R}_{a,b}$ and $e \in r_{a,b}$. The direct travelers of all these origin-destination pairs in line l are subject to the line capacity on edge e. This results in the following inequality.

$$\sum_{\substack{a,b \in V_T^2 \\ r \in \mathcal{R}_{a,b}, e \in r_{a,b}}} y_{r,a,b} \leqslant C \cdot \varphi \tag{5.4}$$

The objective of maximizing the number of direct travelers can easily be formulated by

$$\max \sum_{a,b \in V_T^2} \sum_{r \in \mathcal{R}_{a,b}} y_{r,a,b}. \tag{5.5}$$

The relationship to a multi-commodity flow problem is quite obvious. Inequality (5.3) represents the bounds for the supply and demand of commodity a,b and (5.4) describes the bundle capacity constraint. The flow conservation constraint is implicitly given by the variables $y_{r,a,b}$ that corresponds to paths connecting the terminal nodes a and b of commodity a,b (cf. [1] for an introduction to multi-commodity flows).

Now, let us recall the linear formulation of a feasible line plan. For each possible route $r \in \mathcal{R}$ we introduce an integer variable $x_r \in \mathbb{Z}_+$, which denotes the frequency of route r in the line plan. With the bounds for the line frequency requirement $\overline{lfr}$, $\underline{lfr}$ we obtain

$$\underline{lfr}(e) \leqslant \sum_{\substack{r \in \mathcal{R} \\ r \ni e}} x_r \leqslant \overline{lfr}(e). \tag{5.6}$$

We can easily combine the parts (5.3)-(5.5) and (5.6) by replacing φ in (5.4) by x_r. The entire model reads as follows.

$$\text{(LOP)} \quad \begin{aligned}
\max \quad & \sum_{a,b \in V_T^2} \sum_{r \in \mathcal{R}} y_{r,a,b} \\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}\,,\, r \ni e} x_r \geqslant \underline{lfr}(e) && \forall\, e \in E && (5.7)\\
& \sum_{r \in \mathcal{R}\,,\, r \ni e} x_r \leqslant \overline{lfr}(e) && \forall\, e \in E && (5.8)\\
& \sum_{r \in \mathcal{R}_{a,b}} y_{r,a,b} \leqslant T^{a,b} && \forall\, a,b \in V_T^2 && (5.9)\\
& \sum_{\substack{a,b \in V_T^2 \\ \mathcal{R}_{a,b} \ni r, r_{a,b} \ni e}} y_{r,a,b} \leqslant C \cdot x_r && \forall\, r \in \mathcal{R}\,, \quad \forall\, e \in r && (5.10)\\
& x \in \mathbb{Z}_+^{|\mathcal{R}|},\ y \in \mathbb{Z}_+^{\sum_{a,b \in V_T^2} |\mathcal{R}_{a,b}|} && && (5.11)
\end{aligned}$$

The instances from the data pool are provided with the information required by the model (LOP) with the exception of the set of possible routes $\mathcal{R}$. The determination of this set is subject to various rules depending on the particular transportation network and several operational constraints (cf. section 4.5). The only relevant information concerning infrastructure we are aware of consists of the classification property of stations. The termination of lines is allowed at classification yard $V' \subset V$, only. Due to the absence of further relevant infrastructure information, we apply the concept of *routes on shortest paths*. In this concept, also used in various papers of transportation science [27, 42, 62, 69], the set $\mathcal{R}$ consists of routes representing a *shortest path* in the network $G = (V,E)$ with respect to some edge weight $w : E \to \mathbb{Z}_+$. Convenient edge weights $w(e)$ represent the ride time or the distance between the terminals of e. In order to apply the minimal information about the infrastructure we allow routes on shortest paths connecting classification yards only. If we assume a unique shortest path connecting two classification yards the cardinality of $\mathcal{R}$ is $|V'|(|V| \Leftrightarrow 1)/2$. Furthermore, if a shortest path with respect to w connecting a and b represents a pleasant travel path for origin-destination pair $a,b \in V_T^2$, the set of routes on shortest paths $\mathcal{R}$ provides an easy determination of $\mathcal{R}_{a,b}$, i.e.

$$\mathcal{R}_{a,b} = \{r \in \mathcal{R} \mid a \in r \text{ and } b \in r\},$$

because every subpath $r_{a,b}$ of a shortest path r is a shortest path.

The application of the concept of routes on shortest paths has certain advantages, but in general the model (LOP) operates with any set of routes $\mathcal{R}$. For the cost optimal approach, described in chapter 6, we are provided with a set of possible routes designed by practitioners from Nederlandse Spoorwegen. This set also contains routes on shortest paths and combinations of shortest paths that satisfy the operational constraints. The cardinality of this set is significantly smaller compared to the set of routes on shortest paths connecting classification yards. We can conclude that our choice of routes gives a reasonable approximation of a real set of routes in various aspects.

With the above definition of R and $R_{a,b}$ we can establish the (LOP) model for our real world data. The compact formulation of (LOP) results in an integer linear program of tremendous size (cf. table 5.4 and 5.5). Even the solution of the initial linear programming relaxation of the largest instances requires an exhaustive use of resources (cf. table 5.6 and 5.7). All computational experiments reported in this chapter were performed on an SGI Power Challenge with 4 gigabyte of core memory running IRIX 6.2. We applied the commercial mixed integer linear programming solver CPLEX version 5.0 in order to solve the integer linear programs. For the initial linear programming relaxation of the (LOP) formulation we applied the CPLEX barrier method. BIXBY [10] also solved the pure linear programming relaxation without any integer programming preprocessing of the `ns` and `dbag` instances using the primal simplex method of CPLEX in 107755 CPU seconds on the same machine (compared to 13399 CPU seconds for the barrier method[2])

Although we could solve the smallest instances from our data pool (cf. table 5.6 and 5.7), most of the problems are provided with an unsatisfactory performance guarantee. Furthermore, the usage of resources (time and space) is absolutely inacceptable for a practically relevant approach.

	nsic	nsir	nsar	dbagic	dbagir	sbb1	sbb2	sbb3
# constraints	979	27186	86629	32905	216265	21596	4314	3065
# variables	1888	110637	566514	133539	772271	69920	8144	5575
# non-zeros	6165	523252	3538200	734063	4113508	365626	33260	21172

Table 5.4: Size of the (LOP) model (railroad)

	bvagtram	bvagbus	vbzsbahn	vbztram	vbzbus
# constraints	1478	55537	821	4140	15834
# variables	4308	313399	1854	13659	64655
# non-zeros	36307	3247850	8930	108865	469946

Table 5.5: Size of the (LOP) model (urban transport)

[2]Table 5.6 presents the total solution time. The time for solving the pure linear programming relaxation is 0.32 (`nsic`), 190.35 (`nsir`), 6285.04 (`nsar`), 403.34 (`dbagic`), and 6519.64 (`dbagir`) CPU seconds.

	nsic	nsir	nsar	dbagic	dbagir	sbb1	sbb2	sbb3
LP relaxation in the first B&B node	8391594	21287060	25231171	7866185	6178077	45045	47476	11164
best bound	8206670	21287060	25231171	7865981	6178077	44953	47236	11156
best solution	8206670	-	-	-	-	43846	47236	11156
gap	0.0%	-	-	-	-	2.0%	0.0%	0.0%
# B&B nodes	724	5	1	14	1	112	6	37
CPU seconds	377	86400	86400	86400	86400	86400	37	107

Table 5.6: Computational results with the (LOP) model (railroad)

	bvagtram	bvagbus	vbzsbahn	vbztram	vbzbus
LP relaxation in the first B&B node	138483	160784	4962379	4566535	3398852
best bound	138483	160784	4954579	4458336	3386588
best solution	138483	-	4954579	4144512	2777609
gap	0.0%	-	0.0%	7.6%	21.9%
# B&B nodes	1	1	740	20364	211
CPU seconds	6	86400	84	86400	86400

Table 5.7: Computational results with the (LOP) model (urban transport)

5.5 A relaxation of (LOP)

The relaxation of the (LOP) formulation described in this section is related to the well known relaxation methods for multi-commodity flow problems. There are various techniques for relaxing the bundle constraints of multi-commodity flow problems, including Lagrangian relaxation (cf. [1], chapter 16), in order to decompose the problem in several single commodity flow problems. We also apply a relaxation of the bundle constraint related inequalities (5.10) by relaxing

$$\sum_{\substack{a,b \in V_T^2 \\ R_{a,b} \ni r, r_{a,b} \ni e}} y_{r,a,b} \leqslant C \cdot x_r \qquad \forall\, r \in R\,, \quad \forall\, e \in r$$

to

$$y_{r,a,b} \leqslant C \cdot x_r \quad \forall\, a,b \in V_T^2,\ \forall\, r \in R_{a,b}. \tag{5.12}$$

Instead of bounding the sum of relevant variables from above by $C \cdot x_r$ we have the same upper bound for each single element of the sum, which obviously gives a relaxation of (LOP). In terms of travelers and train capacity the resulting formulation obeys the capacity constraints for the direct travelers of a single origin-destination pair, only. The number of direct travelers of an origin-destination pair a,b is subject to the provided capacity of lines on routes in $R_{a,b}$ disregarding other travelers using the same line.

Furthermore, we aggregate all constraints of type (5.12) of one origin-destination pair $a,b \in$

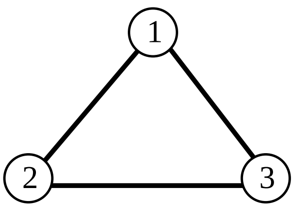
V_T^2, which results in the following inequality.

$$\sum_{r\in R_{a,b}} y_{r,a,b} \leqslant C \sum_{r\in R_{a,b}} x_r \tag{5.13}$$

In the resulting model the y variables of one particular origin-destination pair $a,b \in V_T^2$ occur always in the form $\sum_{r\in R_{a,b}} y_{r,a,b}$. Therefore, we substitute $\sum_{r\in R_{a,b}} y_{r,a,b}$ by one single variable $y_{a,b}$, which represents the number of direct travelers in all suitable lines. The entire formulation derived by the relaxation and the aggregation reads as follows.

$$\text{(lop)} \qquad \begin{aligned} \max \quad & \sum_{a,b\in V_T^2} y_{a,b} \\ \text{s.t.} \quad & \sum_{r\in R,\, r\ni e} x_r \geqslant \underline{lfr}(e) \qquad \forall\, e \in E && (5.14)\\ & \sum_{r\in R,\, r\ni e} x_r \leqslant \overline{lfr}(e) \qquad \forall\, e \in E && (5.15)\\ & y_{a,b} \leqslant T^{a,b} \qquad \forall\, a,b \in V_T^2 && (5.16)\\ & y_{a,b} \leqslant C \sum_{r\in R_{a,b}} x_r \qquad \forall\, a,b \in V_T^2 && (5.17)\\ & x \in \mathbb{Z}_+^{|R|},\ y \in \mathbb{Z}_+^{|V_T^2|} && (5.18) \end{aligned}$$

In the (lop) model we can additionally relax the integrality of the y variables to $y_{a,b} \in \mathbb{R}_+$. According to the objective of (lop) we easily derive $y_{a,b} = \min\{T^{a,b}, C \cdot \sum_{r\in R_{a,b}} x_r\}$. The integrality of x, T, and C implies the integrality of y. A similar relaxation can be done for the (LOP) model with the $y_{r,a,b}$ variables which significantly reduces the number of integer variables. But in contrast to the (lop) model the integrality of the y variables is not implied by the integrality of x, T, and C as one can see in the example depicted in figure 5.3



$T^{1,2} = T^{1,3} = T^{2,3} = 1$, $C = 1$
$r = 1 \Leftrightarrow 2 \Leftrightarrow 3 \Leftrightarrow 1$, $x_r = 1$. $R_{1,2} = R_{1,3} = R_{2,3} = \{r\}$
$y \in \mathbb{Z}_+^3$: optimal solution $y_{r,1,2} = 1$, $y_{r,1,3} = y_{r,2,3} = 0$.
optimal solution value: 1
$y \in \mathbb{R}_+^3$: optimal solution $y_{r,1,2} = y_{r,1,3} = y_{r,2,3} = 0.5$.
optimal solution value: 1.5

Figure 5.3: Counterexample for $x \in \mathbb{Z}_+$ implies $y \in \mathbb{Z}_+$ for (LOP)

Note that the solution of the relaxed model (lop) still provides a feasible line plan, but the objective value of an optimal solution of (lop) gives an upper bound for the number of direct travelers, only.

The resulting integer linear programs derived from the (lop) model are of substantially reduced size (cf. table 5.8 and 5.9). Moreover, solving the initial linear programming relaxation

	nsic	nsir	nsar	dbagic	dbagir	sbb1	sbb2	sbb3
# constraints	241	2258	7951	2681	7350	2017	342	221
# variables	463	5715	11540	6579	25209	4663	1218	870
# non-zeros	2615	134323	602110	163393	936085	97345	14367	9802

Table 5.8: Size of the (lop) model (railroad)

	bvagtram	bvagbus	vbzsbahn	vbztram	vbzbus
# constraints	1173	7517	471	3770	4992
# variables	1191	9283	517	3843	5773
# non-zeros	4989	333448	2315	17672	75764

Table 5.9: Size of the (lop) model (urban transport)

is quite fast even for the largest instances (cf. table 5.10 and 5.11). The entire solution times are quite reasonable on the high-end SGI workstation, too. But if we think of an integration of this method into an interactive *railroad management system*, the requirement of computational resources must be reduced. Furthermore, solving the (lop) model requires a precise tuning of the parameters of the branch-and-bound algorithm implemented in CPLEX. The determination of *optimal* parameter settings requires several time consuming experiments and significantly depends on the particular instance. This becomes evident in our experiments with different mixed integer linear programming solvers. We applied several mixed integer linear programming codes[3] to the `nsir` instance of (lop). We always used the default strategy of the particular solver and set a time limit of one CPU hour. Table 5.12 summarizes the results of our investigations.

In the remaining part of this section we focus on improvements of the (lop) formulation based on a problem specific preprocessing and on constraint generation.

[3]CPLEX: `http://www.cplex.com`, XPRESS-MP: `http://www.dash.co.uk`,
MINTO: `http://www.gatech.akula.isye.gatech.edu/~mwps/projects/minto.html`,
MOPS: `http://www.fu-berlin.de/w3/w3suhl/mops.htm`, OSL: `http://www.ibm.com/osl`

	nsic	nsir	nsar	dbagic	dbagir	sbb1	sbb2	sbb3
LP relaxation in the first B&B node	8206670	27101632	37120193	7451191	6100286	44920	47299	11462
optimal solution	8206670	27065722	37059015	7451191	6097010	44920	47299	11462
# B&B nodes	1	11	13	1	70	1	1	1
CPU seconds	0.10	15.58	464.46	12.64	138.90	5.21	0.38	0.20

Table 5.10: Computational results with the (lop) model (railroad)

	`bvagtram`	`bvagbus`	`vbzsbahn`	`vbztram`	`vbzbus`
LP relaxation					
in the first B&B node	138483	153479	4954579	4406569	3212151
optimal solution	138483	153479	4954579	4406569	3212151
# B&B nodes	1	1	1	1	1
CPU seconds	0.77	37.09	0.09	4.20	13.29

Table 5.11: Computational results with the (lop) model (urban transport)

	best bound	best solution	gap	CPU seconds
CPLEX 2	28864238	24106947	19.7%	3600.00
CPLEX 3	27065722	27065722	0.0%	685.24
CPLEX 4	27065722	27065722	0.0%	444.18
CPLEX 5	27065722	27065722	0.0%	1207.28
MINTO 2.0	28864238	-	-	3600.00
MINTO 3.0	27065722	27065722	0.0%	58.77
MOPS 3	28811396	2586945	11.4%	3600.00
OSL 1.2	28864238	24022094	20.2%	3600.00
XPRESS-MP 10.28	28810248	25730246	12.0%	3600.00

Table 5.12: Computational results with different mixed integer linear programming solvers for the `nsir` instance of the (lop) model

5.5.1 Preprocessing

The first preprocessing technique focuses on eliminating some of the constraints (5.14) and (5.15). Assume that the supply network $G = (V, E)$ contains a *chain* $v = u_0 \Leftrightarrow u_1 \Leftrightarrow \ldots \Leftrightarrow u_{k-1} \Leftrightarrow u_k = w$ connecting two classification yards v and w. A chain is a path in G, where u_i, $i \in \{1, \ldots, k \Leftrightarrow 1\}$ represents non-classification yards of degree two. For each edge $e = u_{i-1}u_i$, $i \in \{1, \ldots, k\}$ the left hand side of the inequalities (5.14) and (5.15) is identical. Therefore, we can replace the $2k$ inequalities by

$$\sum_{r \in R\,,\, r \ni e} x_r \geqslant \max_{i \in \{1,\ldots,k\}} \underline{lfr}(u_{i-1}u_i) =: \underline{lfr}^{vw} \quad \text{and} \quad \sum_{r \in R\,,\, r \ni e} x_r \leqslant \min_{i \in \{1,\ldots,k\}} \overline{lfr}(u_{i-1}u_i) =: \overline{lfr}^{vw}$$

and reduce the number of constraints in the (lop) formulation.

Another preprocessing technique focuses on eliminating and aggregating some y variables. Consider an origin-destination pair $a, b \in V_T^2$ with $ab \in E$. First of all, note that if $T^{a,b} \leqslant C \cdot \underline{lfr}(ab)$ then $y_{a,b} = T^{a,b}$ in any optimal solution, because there are at least $\underline{lfr}(ab)$ trains with sufficient capacity that provide a direct connection for travelers of origin-destination pair a, b. Moreover, if the stations a and b belong to the set of classification yards and there is a route $ab \in R$, then $y_{a,b} = \min\{T^{a,b}, C \cdot \overline{lfr}(ab)\}$ in any optimal solution, because we can increase the frequency of x_{ab} in order to provide sufficient capacity for $y_{a,b}$. Hence, we can eliminate $y_{a,b}$ from the problem. Furthermore, if ab represents an edge of a chain $v = u_0 \Leftrightarrow u_1 \Leftrightarrow \ldots \Leftrightarrow u_{k-1} \Leftrightarrow u_k = w$ and this chain belongs to $\mathcal{R}$, we derive by the same argument stated above that $y_{a,b} = \min\{T^{a,b}, C \cdot \overline{lfr}^{vw}\}$ in any optimal solution. Hence, we can eliminate $y_{a,b}$ from the problem. Another class of suitable candidates for elimination corresponds to origin-destination pairs $u_i u_j$ of a chain $v = u_0 \Leftrightarrow u_1 \Leftrightarrow \ldots \Leftrightarrow u_{k-1} \Leftrightarrow u_k = w$. Similar to the cases above, we can eliminate $y_{u_i u_j}$ from the problem if $T^{u_i,u_j} \leqslant \underline{lfr}^{vw}$ or $v = u_0 \Leftrightarrow u_1 \Leftrightarrow \ldots \Leftrightarrow u_{k-1} \Leftrightarrow u_k = w \in \mathcal{R}$.

The most promising reduction technique is based on aggregating y variables corresponding to origin-destination pairs a_1, b_1 and a_2, b_2 with $\mathcal{R}_{a_1,b_1} = \mathcal{R}_{a_2,b_2}$ and $r_{a_1,b_1} \cap r_{a_2,b_2} \neq \emptyset$ for each $r \in \mathcal{R}_{a_1,b_1} = \mathcal{R}_{a_2,b_2}$. Such a situation frequently occurs for origin-destination pairs a, b_1 and a, b_2 where b_1 and b_2 belong to the inner nodes of a chain $v = u_0 \Leftrightarrow u_1 \Leftrightarrow \ldots \Leftrightarrow u_{k-1} \Leftrightarrow u_k = w$. We replace the origin-destination pairs a, b_1 and a, b_2 by a new symbolic pair $a_{1,2}, b_{1,2}$ with $\mathcal{R}_{a_{1,2},b_{1,2}} := \mathcal{R}_{a_1,b_1} = \mathcal{R}_{a_2,b_2}$ and $T^{a_{1,2},b_{1,2}} := T^{a_1,b_1} + T^{a_2,b_2}$. We discuss the influence of this aggregation on the model (lop) and (LOP). Therefore, consider a feasible line plan represented by $x^* \in \mathbb{Z}_+^{|\mathcal{R}|}$. Let $\overline{y}^*$ ($\underline{y}^*$) be the optimal solution of (lop) ((LOP)) with x variables fixed to x^* of the instance with unchanged origin-destination data. Let y^* be the optimal solution of (lop) with x variables fixed to x^* of the instances with the new symbolic origin-destination pair $a_{1,2}, b_{1,2}$.

CLAIM 5.1
$1^T \overline{y}^* \geqslant 1^T y^* \geqslant 1^T \underline{y}^*$.

PROOF The first inequality is easily derived from the fact that $\overline{y}^*_{a,b} = y^*_{a,b}$ for any origin-destination pair $a, b \notin \{(a_1, b_1), (a_2, b_2)\}$ and

$$\min\{T^{a_1,b_1}, C \sum_{r \in \mathcal{R}_{a_1,b_1}} x_r\} + \min\{T^{a_2,b_2}, C \sum_{r \in \mathcal{R}_{a_2,b_2}} x_r\} \geqslant$$
$$\min\{T^{a_1,b_1} + T^{a_2,b_2}, C \sum_{r \in \mathcal{R}_{a_{1,2},b_{1,2}}} x_r\} = \min\{T^{a_{1,2},b_{1,2}}, C \sum_{r \in \mathcal{R}_{a_{1,2},b_{1,2}}} x_r\} = y^*_{a_{1,2},b_{1,2}}.$$

Similarly, for the second inequality the relaxation of (5.10) provides $\sum_{r \in \mathcal{R}_{a,b}} \underline{y}^*_{r,a,b} \leqslant y^*_{a,b}$ for $a, b \notin \{(a_1, b_1), (a_2, b_2)\}$. Now, consider origin-destination pairs a_1, b_1 and a_2, b_2. With the assumption that for all $r \in \mathcal{R}_{a_1,b_1} = \mathcal{R}_{a_2,b_2}$ we have an edge $e \in r_{a_1,b_1} \cap r_{a_2,b_2}$, there is an inequality of type (5.10) which gives

$$C \cdot x_r \geqslant \sum_{\substack{a,b \in V_T^2 \\ r \in \mathcal{R}_{a,b}, e \in r_{a,b}}} \underline{y}^*_{r,a,b} \geqslant \underline{y}^*_{r,a_1,b_1} + \underline{y}^*_{r,a_2,b_2}.$$

Hence, we have $\sum_{r \in \mathcal{R}_{a_{1,2},b_{1,2}}} \underline{y}^*_{r,a_1,b_1} + \underline{y}^*_{r,a_2,b_2} \leqslant C \sum_{r \in \mathcal{R}_{a_{1,2},b_{1,2}}}$ and with (5.9) we can derive

$$\sum_{r \in \mathcal{R}_{a_{1,2},b_{1,2}}} \underline{y}^*_{r,a_1,b_1} + \underline{y}^*_{r,a_2,b_2} \leqslant \min\{T^{a_1,b_1} + T^{a_2,b_2}, C \sum_{r \in \mathcal{R}_{a_{1,2},b_{1,2}}} x_r\} = y^*_{a_{1,2},b_{1,2}},$$

which completes the proof. □

	constraints			variables			non-zeros		
	original	preproc.	reduction	original	preproc.	reduction	original	preproc.	reduction
`nsar`	7951	3099	61.0%	11540	6688	42.0%	602110	227698	62.2%
`dbagir`	6245	7350	15.0%	25209	24104	4.4%	936085	718397	23.2%

Table 5.13: Size reduction provided by preprocessing techniques

COROLLARY 5.2

Let $\overline{z}^*$ ($\underline{z}^*$) be the optimal solution value of (lop) ((LOP)) of the instance with unchanged origin-destination data. Let z^* be the optimal solution value of (lop) with the new symbolic origin-destination pair $a_{1,2}, b_{1,2}$. Then we have $\overline{z}^* \geqslant z^* \geqslant \underline{z}^*$.

The aggregation of origin-destination pairs a_1, b_1 and a_2, b_2 reduces the size of the resulting integer linear program and provides a closer relaxation of (LOP).

The preprocessing results derived above provide a particular *shrinking of nodes* of a chain $v = u_0 \Leftrightarrow u_1 \Leftrightarrow \ldots \Leftrightarrow u_{k-1} \Leftrightarrow u_k = w$ to $v \Leftrightarrow u \Leftrightarrow w$ (cf. figure 5.4) with $\underline{lfr}(vu) = \underline{lfr}(uw) = \underline{lfr}^{vw}$ and $\overline{lfr}(vu) = \overline{lfr}(uw) = \overline{lfr}^{vw}$. The travelers of an origin-destination pair u_i, u_j whose corresponding y variable cannot be eliminated, can be added to the origin-destination pair v, w. The travelers of origin-destination pair u_i, a, $i \in \{1, \ldots, k \Leftrightarrow 1\}$ and $a \notin \{u_0, \ldots, u_k\}$, can be added to the origin-destination pair v, w if $R_{u_i,a} = R_{u_j,a}$ and $r_{u_i,a} \cap r_{u_j,a} \neq \emptyset$ for all $j \in \{1, \ldots, k \Leftrightarrow 1\}$.

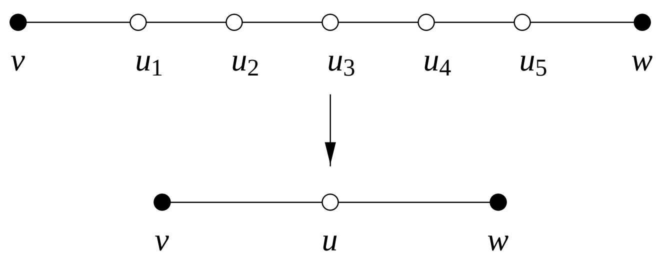


Figure 5.4: Shrinking of non-classification yards

We apply the preprocessing techniques to the two largest instances `nsar` and `dbagir`. Table 5.13 summarizes the significant reduction of the size of the corresponding integer linear program.

5.5.2 Constraint generation

The most effective technique for solving (mixed) integer linear programs is based on improving the linear programming relaxation by valid inequalities or cuts. A first class of valid inequalities is related to the constraints (5.14) and (5.15) and considers x variables, only. Therefore, this class of cuts can be applied to any formulation including inequalities (5.14) and (5.15) (cf. section 5.7 and 6.5.2). The cuts focus on eliminating fractional solutions similar to the fractional solution depicted in figure 5.5.

PROPOSITION 5.3

Let $E' \subset E$, $\alpha^r_{E'} := |r \cap E'|$, $\alpha^{\max}_{E'} = \max\{\alpha^r_{E'} \mid r \in R\}$, and $\alpha^{\min}_{E'} = \min\{\alpha^r_{E'} \mid r \in R, \alpha^r_{E'} \geqslant 2\}$.

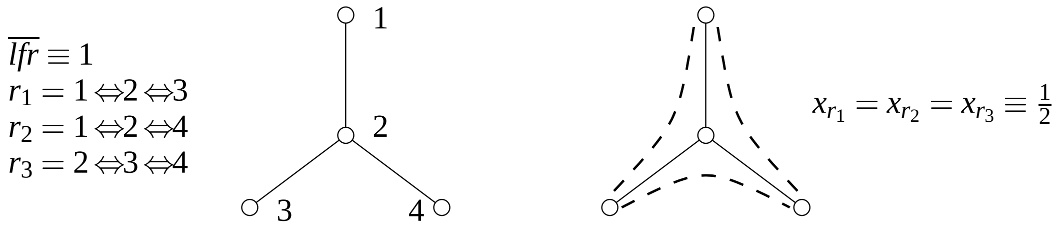


Figure 5.5: A fractional solution

The inequalities

$$\sum_{r \in R, \alpha^r_{E'} \geqslant 1} x_r \geqslant \left\lceil \frac{\underline{lfr}(E')}{\alpha^{\max}_{E'}} \right\rceil \tag{5.19}$$

$$\sum_{r \in R, \alpha^r_{E'} \geqslant 2} x_r \leqslant \left\lfloor \frac{\overline{lfr}(E')}{\alpha^{\min}_{E'}} \right\rfloor \tag{5.20}$$

are valid for (lop).

PROOF From inequalities (5.14) we easily derive

$$\sum_{r \in R, \alpha^r_{E'} \geqslant 1} \alpha^r_{E'} x_r \geqslant \underline{lfr}(E')$$

and by replacing $\alpha^r_{E'}$ by $\alpha^{\max}_{E'}$ we obtain

$$\sum_{r \in R, \alpha^r_{E'} \geqslant 1} x_r \geqslant \frac{\underline{lfr}(E')}{\alpha^{\max}_{E'}}.$$

The left hand side is always integer and hence we can round up the right hand side to the next integer greater or equal than $\underline{lfr}(E')/\alpha^{\max}_{E'}$ which leads to (5.19). We omit the proof of inequality (5.20) which is quite similar to the proof of (5.19). □

With $E' = E$, we derive $x_{r_1} + x_{r_2} + x_{r_3} \leqslant \lfloor 3/2 \rfloor = 1$ from (5.20) for the example depicted in figure 5.5. Due to the exponential number of cuts of type (5.19) and (5.20) we must focus on the dynamic generation and separation. For a given fractional solution, the separation question is, whether there is a subset E' that leads to a violated inequality of the appropriate type or not. Unfortunately, we have the following result.

PROPOSITION 5.4
The separation problem for the cuts of type (5.19) and (5.20) is NP-complete.

PROOF Obviously, the separation problem belongs to NP. We prove the completeness by polynomially transforming the NP-complete CLIQUE problem [35] to the separation of (5.20).

An instance of CLIQUE consists of a graph $G = (V,E)$ and an integer k. The question is, whether there exists a clique of size at least k, i.e. a subset $V' \subset V$ with $|V'| \geqslant k$ such that every two nodes in V' are joined by an edge in E. Now, based on G and k we construct an instance of the separation problem with size polynomially bounded in the size of the input G and k. Let $\tilde{G}$ be a star graph (cf. section 4.5.1) with node set $\tilde{V} = V \cup \{w\}$ and edge set $\tilde{E} = \{vw \mid v \in V\}$. Furthermore, we have the set of possible routes, given by $\mathcal{R} = \{u \Leftrightarrow w \Leftrightarrow v \mid uv \in E\}$. Finally, we define $\underline{lfr} \equiv 1$ and the fractional solution

$$x \equiv \begin{cases} \frac{k}{k(k-1)-2} & \text{if } k \in 2\mathbb{Z}_+ \\ \frac{k-2}{k(k-1)-2} & \text{otherwise.} \end{cases}$$

G contains a clique of size at least k, if and only if, there is a violated inequality of type (5.20). We prove this claim for $k \notin 2\mathbb{Z}_+$. The proof for $k \in 2\mathbb{Z}_+$ is quite similar. Suppose G contains a clique $V' \subset V$ with $|V'| = k$. Then, with $\tilde{E}' = \{vw \mid v \in V'\}$, we have

$$\sum_{r \in \mathcal{R}, \alpha^r_{\tilde{E}'} \neq 0} x_r = \frac{k(k \Leftrightarrow 1)}{2} \frac{k \Leftrightarrow 2}{k(k \Leftrightarrow 1) \Leftrightarrow 2} > \frac{k \Leftrightarrow 2}{2} = \left\lfloor \frac{\overline{lfr}(\tilde{E}')}{2} \right\rfloor$$

which proves the violation of the corresponding inequality. Conversely, assume we have a violated cut of type (5.20) derived from the set $\tilde{E}'$, with $|\tilde{E}'| = k$. Let $V' \subset V$ be the corresponding node set in G. Assume V' is not a clique. Hence, there are at most $k(k \Leftrightarrow 1)/2 \Leftrightarrow 1$ edges joining the nodes of V' and therefore we have

$$\sum_{r \in \mathcal{R}, \alpha^r_{\tilde{E}'} \neq 0} x_r \leqslant \frac{k(k \Leftrightarrow 1) \Leftrightarrow 2}{2} \frac{k \Leftrightarrow 2}{k(k \Leftrightarrow 1) \Leftrightarrow 2} = \frac{k}{2} \Leftrightarrow 1 = \left\lfloor \frac{\overline{lfr}(\tilde{E}')}{2} . \right\rfloor$$

This contradicts the assumption of a violated cut corresponding to $\tilde{E}'$, and therefore proves the statement. A similar proof applies for the separation of (5.19). □

Due to a small maximum degree ($\max\{|\delta(v) \mid v \in V\} \leqslant 9$) of all instances in our data pool, we explicitly add all valid inequalities corresponding to $E' \subset \delta(v)$ with $|E'| \geqslant 3$ and $\underline{lfr}(E') \notin 2\mathbb{Z}_+$. This moderately increases the size of the resulting mixed integer linear program. Moreover, this subset seems to represent the practical and theoretical (cf. section 5.6) most efficient cuts suggested by proposition 5.3.

The second class of valid inequalities is derived from an observation in dimension two and leads after an obvious transformation into higher dimensions to a powerful class of cuts for (lop).

PROPOSITION 5.5
Consider the set of feasible solutions, given by

$$P = \{(x,y) \mid ay \leqslant b_1 x + c_1,\ ay \leqslant b_2 x + c_2,\ x \in \mathbb{Z},\ y \in \mathbb{R}\}$$

with $a, b_1, b_2, c_1, c_2 \in \mathbb{R}$ *and* $b_1 > b_2$. *The inequality*

$$ay \leqslant (\Delta + b_2)x \Leftrightarrow \Delta\lceil\eta\rceil + c_2 \tag{5.21}$$

is valid for P *with* $\eta = \frac{c_2 - c_1}{b_1 - b_2}$ *and* $\Delta = \lfloor\eta\rfloor(b_2 \Leftrightarrow b_1) + c_2 \Leftrightarrow c_1$.

PROOF Let $(x^*,y^*) \in \{(x,y) \mid ay \leqslant b_1x+c_1,\ ay \leqslant b_2x+c_2,\ x \in \mathbb{Z},\ y \in \mathbb{R}\}$. First of all, assume $\lfloor\eta\rfloor = \lceil\eta\rceil$, i.e. $b_1 \Leftrightarrow b_2 = 1$, then we have $\Delta = 0$ and (5.21) is equal to $ay \leqslant b_2x+c_2$. For the case $\lfloor\eta\rfloor \neq \lceil\eta\rceil$ from

$$0 > \left\lfloor \frac{c_2 \Leftrightarrow c_1}{b_1 \Leftrightarrow b_2} \right\rfloor \Leftrightarrow \frac{c_2 \Leftrightarrow c_1}{b_1 \Leftrightarrow b_2} > \Leftrightarrow 1 \overset{\cdot(b_2-b_1)}{\Leftrightarrow} 0 < \underbrace{\lfloor\eta\rfloor(b_2 \Leftrightarrow b_1) + c_2 \Leftrightarrow c_1}_{=\Delta} < b_1 \Leftrightarrow b_2$$

we derive that $0 < \Delta < b_1 \Leftrightarrow b_2$. Now, the validity for the case with $x \leqslant \lfloor\eta\rfloor$ is derived by the following observation.

$$\begin{aligned}(\Delta+b_2)x \Leftrightarrow \Delta\lceil\eta\rceil + c_2 &\overset{1+\lfloor\eta\rfloor=\lceil\eta\rceil}{=} \Delta(x \Leftrightarrow \lfloor\eta\rfloor) + b_2x \Leftrightarrow \Delta + c_2 \\ &\overset{\Delta<b_1-b_2}{\geqslant} (b_1 \Leftrightarrow b_2)(x \Leftrightarrow \lfloor\eta\rfloor) + b_2x \Leftrightarrow \Delta + c_2 \\ &= b_1x + \underbrace{(b_1 \Leftrightarrow b_2)\lfloor\eta\rfloor \Leftrightarrow \Delta + c_2}_{=c_1} \geqslant ay\end{aligned}$$

Finally, the inequalities

$$(\Delta+b_2)x \Leftrightarrow \Delta\lceil\eta\rceil + c_2 \overset{x\geqslant\lceil\eta\rceil}{\geqslant} b_2x + c_2 + \Delta x \Leftrightarrow \Delta x = b_2x + c_2 \geqslant ay$$

prove the validity of (5.21) for $x \geqslant \lceil\eta\rceil$ and complete the proof. □

With $y = y_{a,b}$ and $x = \sum_{r\in R_{a,b}} x_r$ we apply proposition 5.5 to the inequalities $y_{a,b} \leqslant T^{a,b}$ and $y_{a,b} \leqslant C\sum_{r\in R_{a,b}} x_r$.

COROLLARY 5.6
The inequality

$$y_{a,b} \leqslant \Delta \sum_{r\in R_{a,b}} x_r \Leftrightarrow \Delta \left\lceil \frac{T^{a,b}}{C} \right\rceil + T^{a,b} \tag{5.22}$$

with $\Delta = \Leftrightarrow \left\lfloor \frac{T^{a,b}}{C} \right\rfloor C + T^{a,b}$ *is valid for (lop).*

Note, that for $T^{a,b} < C$ we have $\Delta = T^{a,b}$ and hence (5.22) reads as $y_{a,b} \leqslant T^{a,b}\sum_{r\in R_{a,b}} x_r$. This inequality obviously dominates inequality (5.18) of the origin-destination pair $a,b \in V_T^2$ and hence we can *replace* (5.18) by (5.22) instead of *adding* (5.22) to (lop). The idea of proposition 5.5 is quite simple and could be easily implemented in any general mixed integer linear programming preprocessor.

We apply the preprocessing techniques and the derived cuts to the instances from our data pool and obtain significant improvements. We could solve any instance in less than 43 seconds (cf. tables 5.14 and 5.15). Moreover, with the improvements of the mixed integer programs all tested mixed integer linear programming solvers provide the optimal solution in reasonable computation times (cf. table 5.16).

	nsic	nsir	nsar*	dbagic	dbagir*	sbb1	sbb2	sbb3
# constraints	250	2302	3163	2702	6319	2036	360	228
# variables	463	5715	6688	6579	24104	4663	1218	870
# non-zeros	2780	145604	240181	173876	798231	101302	16560	10277
LP relaxation								
in the first B&B node	8206670	27065722	19725243	7451191	4127768	44920	47299	11462
optimal solution	8206670	27065722	19725243	7451191	4126854	44920	47299	11462
# B&B nodes	1	1	1	1	6	1	1	1
CPU seconds	0.09	6.91	7.91	7.31	42.95	4.21	0.26	0.16

Table 5.14: Computational results with the improved (lop) model (railroad)

* The objective function value must be adjusted by 14189981 (nsar) and by 1970156 (dbagir) due to the elimination of some origin-destination pairs resulting from the preprocessing.

	bvagtram	bvagbus	vbzsbahn	vbztram	vbzbus
# constraints	1173	7531	474	3770	4998
# variables	1191	9283	517	3843	5773
# non-zeros	4989	337503	2361	17672	76580
LP relaxation					
in the first B&B node	138483	153479	4954579	4406569	3212151
optimal solution	138483	153479	4954579	4406569	3212151
# B&B nodes	1	1	1	1	1
CPU seconds	0.83	32.75	0.09	4.36	12.89

Table 5.15: Computational results with the improved (lop) model (urban transport)

5.6 Polyhedral aspects

In this section we concentrate on polyhedral properties of the integer linear programs associated with the line planning problem. We derive some results for the polytope associated with the generic line planning problem with supply network $G=(V,E)$, $\underline{lfr}=\overline{lfr}=: lfr \geqslant 1$, and a set of possible routes R. The set of feasible line plans $\mathbb{L}$ is given by

$$\mathbb{L}=\{x \in \mathbb{Z}_+^{|R|} \mid \mathfrak{A}x = lfr\}$$

where $\mathfrak{A}$ represents the edge-route incidence matrix (cf. section 4.6). The first problem arising in a polyhedral analysis is the determination of the dimension of conv $\mathbb{L}$. For general instances of the generic line planning problem, the associated recognition problem is NP-complete, because deciding if $\mathbb{L}=\emptyset$ or not, is already NP-complete (cf. corollary 4.3). Therefore, we concentrate

	CPLEX 2	CPLEX 3	CPLEX 4	CPLEX 5	MINTO 2.0	MINTO 3.0	MOPS 3	OSL 1.2	XPRESS-MP 10.28
CPU seconds	29.66	20.84	24.44	21.09	17.25	32.02	40.83	99.71	24.88

Table 5.16: Computation time for the nsir instance of the improved (lop) model

on a particular class of instances, where $\mathcal{R}$ contains the single edge routes $r = e$ for each edge $e \in E$.

PROPOSITION 5.7
$\dim(\operatorname{conv} \mathbb{L}) = |\mathcal{R}| \Leftrightarrow |E|$.

PROOF The linear equation system $\mathfrak{A}x = lfr$ belongs to the linear description $Ax \leqslant b$ of $\operatorname{conv} \mathbb{L}$. Hence $(A^=, b^=)$ includes $(\mathfrak{A}, lfr)$ which provides a rank of $|E|$ for $(A^=, b^=)$, because the submatrix corresponding to the single edge routes represents an $|E| \times |E|$ identity matrix. With the property $\operatorname{rg}(A^=, b^=) + \dim(P) = n$ for a polytope $P \subset \mathbb{R}^n$ we derive $\operatorname{conv} \mathbb{L} \leqslant |\mathcal{R}| \Leftrightarrow |E|$. In order to achieve $\dim(\operatorname{conv} \mathbb{L}) \geqslant |\mathcal{R}| \Leftrightarrow |E|$ we construct $|\mathcal{R}| \Leftrightarrow |E| + 1$ affinely independent points in $\operatorname{conv} \mathbb{L}$. For each route $r^* \in \mathcal{R} \setminus E$ consider the point $x^{r^*} \in \operatorname{conv} \mathbb{L}$ with $x^{r^*}_{r^*} = 1$, $x^{r^*}_r = 0$ for each $r \in \mathcal{R} \setminus \{E \cup r^*\}$, and $x^{r^*}_e = lfr(e) \Leftrightarrow \sum_{r \in \mathcal{R} \setminus E} x^{r^*}_r$. These $|\mathcal{R}| \Leftrightarrow |E|$ points together with $x^E \in \operatorname{conv} \mathbb{L}$ where $x^E_r = 0$ for each $r \in \mathcal{R} \setminus E$ and $x^E_e = lfr(e)$ represent a set of $|\mathcal{R}| \Leftrightarrow |E| + 1$ linearly independent points of $\operatorname{conv} \mathbb{L}$. Linear independence implies affine independence which completes the proof. □

The constraint set of the integer linear program associated with the generic line planning problem consists of $\mathfrak{A}x = lfr$ and the non-negativity inequalities $x_r \geqslant 0$. The following proposition discussed the facet-defining property of these inequalities.

PROPOSITION 5.8
$x_r \geqslant 0$ *is facet-defining for* $\operatorname{conv} \mathbb{L}$ *if* $|r| > 1$.

PROOF We must construct $|\mathcal{R}| \Leftrightarrow |E|$ affinely independent points $x \in \operatorname{conv} \mathbb{L}$ with $x_r = 0$. Obviously, the points x^E, x^{r^*} with $r^* \in \mathcal{R} \setminus \{E \cup r\}$ constructed in the proof of proposition 5.7 provide $x^E_r = 0$, $x^{r^*}_r = 0$ and are linear independent which completes the proof. □

In proposition 5.3 we derive a class of valid inequalities for the (lop) model which also represent valid inequalities for $\operatorname{conv} \mathbb{L}$. These inequalities are facet-defining under certain assumptions.

THEOREM 5.9
Let $E' \subset \delta(v)$ *for a particular node* $v \in V$, $lfr(e) = 1$ *for each* $e \in E'$, $\eta := |E'| = lfr(E')$ *odd, and* $\alpha_r = |E' \cap r|$ *for each* $r \in \mathcal{R}$. *The valid inequality derived from (5.20) of proposition 5.3 reads as follows.*

$$\sum_{\substack{r \in \mathcal{R} \\ \alpha_r = 2}} x_r \leqslant \left\lfloor \frac{\eta}{2} \right\rfloor \tag{5.23}$$

Inequality (5.23) is facet-defining if $\mathcal{R}$ *and* $\mathbb{L}$ *satisfy the following properties.*

1. *There is a subset* $\mathcal{R}' \subset \mathcal{R}$ *of cardinality* $\eta(\eta \Leftrightarrow 1)/2$ *with* $|r \cap E'| = 2$ *for each* $r \in \mathcal{R}'$ *and* $r \cap r' = e \in E'$ *for each pair* $r, r' \in \mathcal{R}'$, $r \neq r'$.
2. *For each* $r^* \in \mathcal{R} \setminus \{\mathcal{R}' \cup E\}$ *there is a line plan* $\hat{x}^{r^*} \in \mathbb{L}$ *with* $\hat{x}^{r^*}_{r^*} = 1$, $\hat{x}^{r^*}_r = 0$ *for each* $r \in \mathcal{R} \setminus \{\mathcal{R}' \cup E \cup r^*\}$ *and* $\sum_{r \in \mathcal{R}, \alpha_r = 2} \hat{x}^{r^*}_r = \lfloor \eta/2 \rfloor$.

Properties 1 and 2 are not an exceptional restriction to $\mathcal{R}$ and $\mathbb{L}$, e.g. they are obviously satisfied if $ee' \in \mathcal{R}$ for each pair $e, e' \in E'$, $e \neq e'$.

PROOF OF THE THEOREM Obviously, with property 2 we have $|\mathcal{R}| \Leftrightarrow |E| \Leftrightarrow \eta(\eta \Leftrightarrow 1)/2$ linear independent points in the face represented by (5.23). In the following we will construct $\eta(\eta \Leftrightarrow 1)/2$ additional points that lead to $|\mathcal{R}| \Leftrightarrow |E|$ linear independent points satisfying $\sum_{r \in \mathcal{R}, \alpha_r = 2} \hat{x}_r = \lfloor \eta/2 \rfloor$.

With $E' = \{e_0, e_1, \ldots, e_{\eta-1}\}$ we present a particular partitioning of the routes $\mathcal{R}'$. According to property 1, for each pair $e_i, e_j \in E'$ we have exactly one route $r_{e_ie_j} \in \mathcal{R}'$ with $\{r_{e_ie_j} \cap E'\} = \{e_i, e_j\}$. The route $r_{e_ie_j} \in \mathcal{R}'$ is uniquely determined by the edges e_i and e_j, therefore we concentrate on the index and omit r. Consider the partitioning of routes $\mathcal{R}'$ defined by

$$\begin{array}{llllll} \{e_0e_1, & e_2e_3, & e_4e_5, & \ldots, & e_{2\eta-2}e_{2\eta-1}\} & =: \; S^{1\cdot\beta} \\ \{e_0e_2, & e_1e_3, & e_2e_4, & \ldots, & e_{2\eta-3}e_{2\eta-1}\} & =: \; S^{2\cdot\beta} \\ \quad\ldots & & & & & \\ \{e_0e_{\lfloor\eta/2\rfloor}, & e_1e_{\lfloor\eta/2\rfloor+1}, & e_2e_{\lfloor\eta/2\rfloor+2}, & \ldots, & e_{2\eta-\lfloor\eta/2\rfloor-1}e_{2\eta-1}\} & =: \; S^{\lfloor\eta/2\rfloor\cdot\beta}. \end{array}$$

All indices in this and subsequent formulas are taken modulo η. The idea of this partitioning becomes clear if we focus on a particular visualization of the sequence for the example of $|E'| = 5$ (cf. figure 5.6).

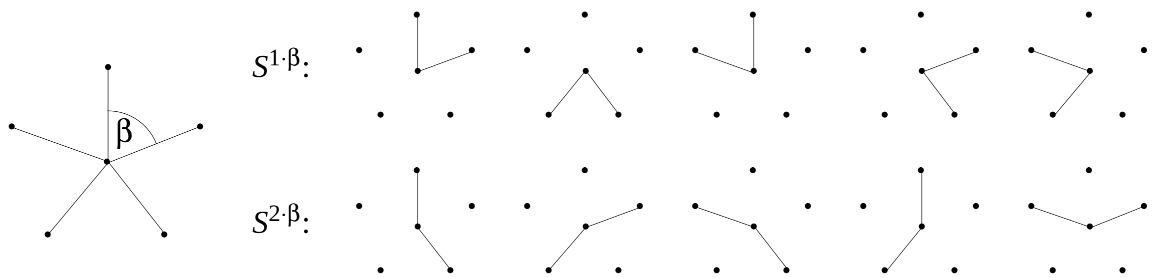


Figure 5.6: Visualization of the partitioning of $\mathcal{R}'$

Furthermore, we rely on a particular order of the η routes in $S^{\kappa\cdot\beta}$, $\kappa = 1, \ldots, \lfloor\eta/2\rfloor$, given by

$$e_0e_\kappa =: r^\kappa_0,\; e_1e_{\kappa+1} =: r^\kappa_1,\; e_2e_{\kappa+2} =: r^\kappa_2,\; \ldots,\; e_{2\eta-\kappa-1}e_{2\eta-1} =: r^\kappa_{\eta-1}.$$

Now, we construct $\eta(\eta \Leftrightarrow 1)/2$ points $x^{\kappa,\iota}$, $\kappa = 1, \ldots, \lfloor\eta/2\rfloor$, $\iota = 0, \ldots, \eta \Leftrightarrow 1$ as follows

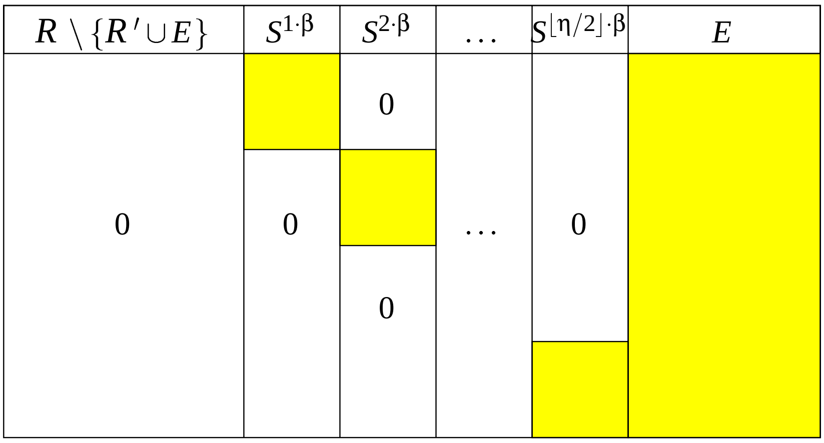
$$x^{\kappa,\iota}_r := \begin{cases} 1 & \text{if } r \in \{r^\kappa_\iota, r^\kappa_{\iota+1}, \ldots, r^\kappa_{\iota+\lfloor\eta/2\rfloor-1}\} \\ 0 & \text{if } r \in \mathcal{R} \setminus \{\mathcal{R}' \cup E\} \end{cases}$$

For the remaining components $x^{\kappa,\iota}_e$ we define

$$x^{\kappa,\iota}_e := lfr(e) \Leftrightarrow \sum_{\substack{r \in \mathcal{R}' \\ e \in r}} x^{\kappa,\iota}_r.$$

By property 1, we easily derive $x^{\kappa,\iota} \in \text{conv}\,\mathbb{L}$ and that $x^{\kappa,\iota}$ belongs to the face represented by (5.23). It remains to prove that the $\eta(\eta \Leftrightarrow 1)/2$ points $\{x^{\kappa,\iota}\}_{\kappa=1,\dots,\lfloor\eta/2\rfloor,\,\iota=0,\dots,\eta-1}$ are linear independent. With $x_r^{\kappa,\iota} = 0$ for $r \in R \setminus \{R' \cup E\}$ this provides the linear independence for all $|R| \Leftrightarrow |E|$ points and completes the proof.

We have the following matrix representation of the points $\{x^{\kappa,\iota}\}_{\kappa=1,\dots,\lfloor\eta/2\rfloor,\,\iota=0,\dots,\eta-1}$.



Therefore, it is sufficient to prove that the points $\{x^{\kappa^*,\iota}\}_{\iota=0,\dots,\eta-1}$ for an arbitrary but fixed $\kappa^* \in \{1,\dots,\lfloor\eta/2\rfloor\}$ are linear independent. Each relevant $\eta \times \eta$ submatrix B corresponding to columns of $S^{\kappa^*\cdot\beta}$ is given by

$$b_{i,j} := \begin{cases} 1 & \text{if } j \in \{i, i+1, \dots, i+\lfloor\eta/2\rfloor \Leftrightarrow 1\} \\ 0 & \text{otherwise} \end{cases}$$

for $i, j \in \{0,\dots,\eta \Leftrightarrow 1\}$. Figure 5.7 depicts the matrix B for $\eta = 5$. The matrix B has rang η which gives the linear independence of the points $\{x^{\kappa^*,\iota}\}_{\iota=0,\dots,\eta-1}$, because $\tilde{B}$ defined by

$$\tilde{b}_{i,j} := \frac{1}{\lfloor\eta/2\rfloor} \cdot \begin{cases} 1 & \text{if } i \in \{j, j+1, \dots, j+\lfloor\eta/2\rfloor \Leftrightarrow 1\} \\ \Leftrightarrow\lfloor\eta/2\rfloor + 1 & \text{if } i = j + \lfloor\eta/2\rfloor \\ 1 & \text{if } i \in \{j+\lfloor\eta/2\rfloor+1, \dots, j+\eta \Leftrightarrow 2\} \\ \Leftrightarrow\lfloor\eta/2\rfloor + 1 & \text{if } i = j+\eta \Leftrightarrow 1 \end{cases}$$

for $i, j \in \{0,\dots,\eta \Leftrightarrow 1\}$ represents the inverse matrix of B. Figure 5.7 depicts the matrix $\tilde{B}$ for $\eta = 5$. We can verify this claim by computing $B \cdot \tilde{B} =: \hat{B}$. Obviously, $\hat{b}_{\sigma,\sigma} = 1$ and $\hat{b}_{\sigma,\rho}$, $\sigma, \rho \in \{0,\dots,\eta \Leftrightarrow 1\}$ is given by

$$\hat{b}_{\sigma,\rho} = \sum_{i \in \{\sigma,\sigma+1,\dots,\sigma+\lfloor\eta/2\rfloor-1\}} \tilde{b}_{i,\rho} = (\lfloor\eta/2\rfloor \Leftrightarrow 1) \cdot 1/\lfloor\eta/2\rfloor \Leftrightarrow 1 \cdot (\lfloor\eta/2\rfloor \Leftrightarrow 1)/\lfloor\eta/2\rfloor = 0.$$

Hence, we obtain that $\hat{B}$ represents the $\eta \times \eta$ identity matrix which completes the proof of theorem 5.9. □

$$B = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \tilde{B} = \frac{1}{2} \cdot \begin{pmatrix} 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \end{pmatrix}$$

Figure 5.7: Matrices B and $\tilde{B}$ for $\eta = 5$.

COROLLARY 5.10
With the assumptions of theorem 5.9 the valid inequality

$$\sum_{\substack{r \in R \\ r \cap E' \neq \emptyset}} x_r \geqslant \left\lceil \frac{\eta}{2} \right\rceil \tag{5.24}$$

derived from (5.19) of proposition 5.3 represents the same facet as inequality (5.23).

PROOF The $|R| - |E|$ linear independent points, constructed in the proof of theorem 5.9, also satisfy $\sum_{r \in R, r \cap E' \neq \emptyset} x_r = \lfloor \eta/2 \rfloor + 1$ which provides the equality of the faces represented by (5.23) and (5.24). □

In the remaining part of this section we discuss polyhedral aspects of the (lop) model with continuous y variables, i.e. $y \geqslant 0$ instead of $y \in \mathbb{Z}_+^{|V_T^2|}$. We focus on two polytopes Q and R defined by

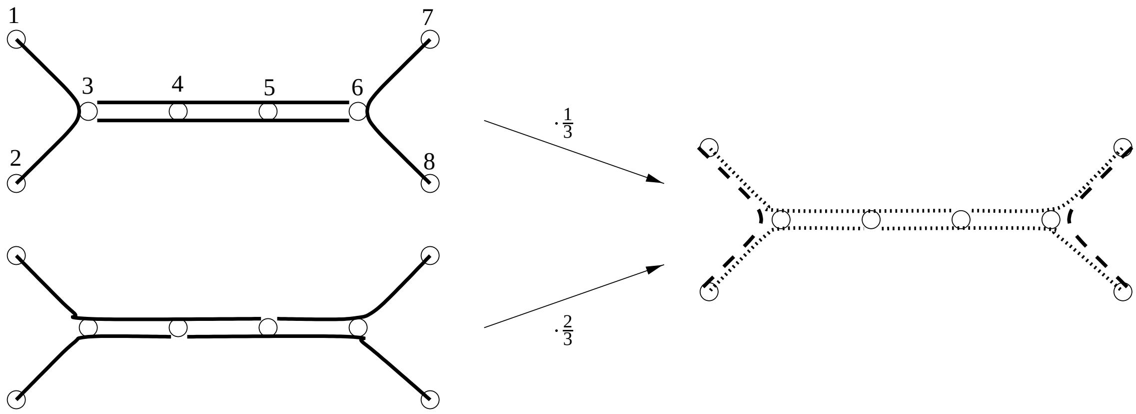
$$\begin{aligned} Q &:= \operatorname{conv}\{(x,y) \mid y_{a,b} \leqslant C \sum_{r \in R_{a,b}} x_r,\ 0 \leqslant y \leqslant T,\ x \in \mathbb{L}\} \\ R &:= \{(x,y) \mid y_{a,b} \leqslant C \sum_{r \in R_{a,b}} x_r,\ 0 \leqslant y \leqslant T,\ x \in \operatorname{conv} \mathbb{L}\}. \end{aligned}$$

Q is the polytope associated with the mixed integer linear program (lop) and we obviously have $Q \subset R$. As described in section 5.5 we may relax the integrality of the y variables to $y \geqslant 0$. Integrality of the x variables and the problem data (C and T) implies the integrality of the y variables in an optimal solution and therefore we may conjecture that $Q = R$. Unfortunately, we have a counter example for $Q = R$ depicted in figure 5.8. Nevertheless, in the following proposition we derive a related property of a slightly modified polytope R^k.

PROPOSITION 5.11
With $k = \max\{\sum_{r \in R_{a,b}} x_r \mid x \in \mathbb{L},\ a,b \in V_T^2\}$*, each extreme point* (x^*, y^*) *of*

$$R^k := \{(x,y) \mid y_{a,b} \leqslant \min\{T^{a,b}, C\} \sum_{r \in R_{a,b}} x_r,\ 0 \leqslant y,\ (k \cdot x) \in \operatorname{conv} \mathbb{L}\}.$$

satisfies $k \cdot x^* \in \mathbb{Z}_+^{|R|}$.



We have the following traveler data $T^{1,2} = T^{3,6} = T^{7,8} = T^{1,5} = T^{2,4} = T^{4,8} = T^{5,7} = 1$. With a train capacity $C = 1$ the first line plan $\{(r = 1 \Leftrightarrow 3 \Leftrightarrow 2, \varphi = 1), (3 \Leftrightarrow 4 \Leftrightarrow 5 \Leftrightarrow 6, 2), (7 \Leftrightarrow 8, 1)\}$ permits a maximum number of 3 direct travelers. Also in the line plan $\{(1 \Leftrightarrow 3 \Leftrightarrow 4 \Leftrightarrow 5, 1), (2 \Leftrightarrow 3 \Leftrightarrow 4, 1), (4 \Leftrightarrow 5 \Leftrightarrow 6 \Leftrightarrow 8, 1), (5 \Leftrightarrow 6 \Leftrightarrow 7, 1)\}$ we have 3 direct travelers. After the convex combination the line plan on the right permits a number of direct travelers of $10/3$ (dash (dotted) routes have frequency $1/3$ ($2/3$)). This obviously leads to $Q \neq R$.

Figure 5.8: Counterexample for $Q = R$.

PROOF Consider the set S of points of R^k defined by

$$\begin{aligned} S \;:=\; & \{(x,y) \mid (k \cdot x) \in \mathbb{L}, \\ & y_{a,b} = 0 \text{ for all } a,b \in I,\ y_{a,b} = \min\{T^{a,b}, C\} \sum_{r \in R_{a,b}} x_r \text{ for all } a,b \in V_T^2 \setminus I,\ \emptyset \subseteq I \subseteq V_T^2\}. \end{aligned}$$

S contains at most $|\mathbb{L}| \cdot 2^{|V_T^2|}$ points and each point $(x^{j^*}, y^{j^*}) \in S = \{(x^j, y^j)\}_{j=1,\dots,\kappa}$ satisfies $k \cdot x^{j^*} \in \mathbb{Z}_+^{|R|}$. We prove that S contains all extreme points of R^k by representing an arbitrary point $(x,y) \in R^k$ by a convex combination of points in S. With $\mathbb{L} = \{k \cdot x^1, \dots, k \cdot x^\iota\}$, let $x = \sum_{i=1}^{\iota} \tilde{\lambda}_i x^i$, $1^T \tilde{\lambda} = 1$, $\tilde{\lambda} \geqslant 0$ be the convex combination of x. With

$$y_{a,b} \leqslant \min\{T^{a,b}, C\} \sum_{r \in R_{a,b}} x_r = \sum_{i=1}^{\iota} \tilde{\lambda}_i \min\{T^{a,b}, C\} \sum_{r \in R_{a,b}} x_r^i$$

we can easily extend the convex combination $x = \sum_{i=1}^{\iota} \tilde{\lambda}_i x^i$ to a convex combination of $(x,y) = \sum_{j=1}^{\kappa} \lambda_i (x^j, y^j)$, $1^T \lambda = 1$, $\lambda \geqslant 0$, and $(x^j, y^j) \in S$, which completes the proof. □

COROLLARY 5.12
If $k = \max\{\sum_{r \in R_{a,b}} x_r \mid x \in \mathbb{L},\ a,b \in V_T^2\} \leqslant 1$, *we have*

$$Q = \{(x,y) \mid y_{a,b} \leqslant C \sum_{r \in R_{a,b}} x_r,\ y_{a,b} \leqslant \Delta \sum_{r \in R_{a,b}} x_r \Leftrightarrow \Delta \lceil T^{a,b}/C \rceil + T^{a,b},\ 0 \leqslant y \leqslant T,\ x \in \operatorname{conv} \mathbb{L}\} =: \tilde{R}$$

with $\Delta = \left\lfloor \frac{T^{a,b}}{C} \right\rfloor C + T^{a,b}$.

PROOF The validity of $Q = \tilde{R}$ is easily derived from $R^1 = \tilde{R}$ and the result of proposition 5.11. With $k = 1$, we have $y_{a,b} \leqslant C$ and hence we can modify the traveler data by $T^{a,b} = \min\{T^{a,b}, C\}$. In this case, the cuts derived by corollary 5.6 replace the capacity constraint $y_{a,b} \leqslant C \sum_{r \in R_{a,b}} x_r$ by $y_{a,b} \leqslant T^{a,b} \sum_{r \in R_{a,b}} x_r$, which proves $R^1 = \tilde{R}$. □

A relaxed version of corollary 5.12 which provides equality for the optimal solution of the corresponding mixed-integer programs can be proved without any polyhedral analysis. The assumption $\max\{\sum_{r \in R_{a,b}} x_r \mid x \in \mathbb{L},\ a,b \in V_T^2\} \leqslant 1$ permits us to directly include the values $T^{a,b}$ to the objective function by assigning the cost coefficient

$$c_r := \sum_{\substack{a,b \in V_T^2 \\ R_{a,b} \ni r}} T^{a,b}$$

to the variable x_r. The y variables become superfluous and hence the line planning problem with respect to the number of direct travelers reduced to the generic line planning problem. Summarizing, we have

$$\begin{aligned} \max\{1^T y \mid (x,y) \in Q\} &= \max\{c^T x | (x,y) \in Q\} = \max\{c^T x | x \in \operatorname{conv} \mathbb{L}\} = \max\{c^T x | (x,y) \in \tilde{R}\} \\ &= \max\{1^T y \mid (x,y) \in \tilde{R}\}. \end{aligned}$$

5.7 Back to the (LOP) model

In section 5.5 and 5.6 we focus on the solution and the structural properties of the relaxed model (lop). With problem specific preprocessing and strong cuts we succeed in solving the relaxation for our real-world instances in reasonable time. An optimal solution $(x^{\text{lop}}, y^{\text{lop}})$ of (lop) provides a feasible line plan represented by x^{lop}. The model (lop) is derived from relaxing the bundle constraint related inequalities (5.10). Therefore, the optimal solution value $1^T y^{\text{lop}}$ of (lop) is an upper bound for the value $1^T y^{\text{LOP}}$ of an optimal solution $(x^{\text{LOP}}, y^{\text{LOP}})$ of (LOP). Conversely, we may take the optimal line plan x^{lop} of (lop) and compute the number of direct travelers subject to the bundle constraints by fixing the x variables to x^{lop}. Note that $|R^{x^{\text{lop}}}| = |\{r \mid x_r^{\text{lop}} > 0, r \in R\}| \ll |R|$ and therefore the resulting integer linear program $(\text{LOP}^{x^{\text{lop}}})$ is quite small. The model $(\text{LOP}^{x^{\text{lop}}})$ reads as follows.

$$(\text{LOP}^{x^{\text{lop}}}) \quad \begin{aligned} z^{x^{\text{lop}}} = \max \quad & \sum_{a,b \in V_T^2} \sum_{r \in R^{x^{\text{lop}}}} y_{r,a,b} \\ \text{s.t.} \quad & \sum_{r \in R_{a,b} \cap R^{x^{\text{lop}}}} y_{r,a,b} \leqslant T^{a,b} && \forall\, a,b \in V_T^2 \\ & \sum_{\substack{a,b \in V_T^2 \\ R_{a,b} \cap R^{x^{\text{lop}}} \ni r, r_{a,b} \ni e}} y_{r,a,b} \leqslant C \cdot x_r^{\text{lop}} && \forall\, r \in R^{x^{\text{lop}}}, \quad \forall\, e \in r \\ & y \in \mathbb{Z}_+^{\sum_{a,b \in V_T^2} |R_{a,b} \cap R^{x^{\text{lop}}}|} \end{aligned}$$

	nsic	nsir	nsar	dbagic	dbagir	sbb1	sbb2	sbb3
objective value ($z^{x^{\text{lop}}}$)	8206670	20901319	24765845	7372418	6097010	43678	47030	11147
upper bound ($1^T y^{\text{lop}}$)	8206670	27065722	33915224	7451191	6097010	44920	47299	11462
gap	0.0%	29.5%	36.9%	1.1%	0.0%	2.8%	0.6%	2.8%

Table 5.17: Performance guarantee of (LOP) provided by (lop) (railroad)

	bvagtram	bvagbus	vbzsbahn	vbztram	vbzbus
objective value ($z^{x^{\text{lop}}}$)	138483	153479	4936792	4304287	3169827
upper bound ($1^T y^{\text{lop}}$)	138483	153479	4954579	4406569	3212151
gap	0.0%	0.0%	0.4%	2.4%	1.3%

Table 5.18: Performance guarantee of (LOP) provided by (lop) (urban transport)

For all instances from our data pool we solve ($\text{LOP}^{x^{\text{lop}}}$) within a few seconds. Obviously, the optimal solution value $z^{x^{\text{lop}}}$ of ($\text{LOP}^{x^{\text{lop}}}$) provides a lower bound for the objective value $1^T y^{\text{LOP}}$ of an optimal solution $(x^{\text{LOP}}, y^{\text{LOP}})$ of (LOP), because the optimal solution of ($\text{LOP}^{x^{\text{lop}}}$) is feasible for (LOP).

Summarizing, we have a feasible solution of (LOP) and a performance guarantee provided by

$$1^T y^{\text{lop}} \geqslant 1^T y^{\text{LOP}} \geqslant z^{x^{\text{lop}}}.$$

Tables 5.17 and 5.18 summarize the results for our real-world instances. With the exception of nsir and nsar the resulting gaps are quite acceptable.

A detailed analysis of the instances nsir and nsar leads to the result that the $\overline{lfr}$ values together with the train capacity C do not satisfy the traffic load ld for some edges, i.e. $\overline{lfr}(e) \cdot C < ld(e)$, and therefore violate a basic assumption. Nevertheless, we can make use of our analysis by adding the constraints

$$\sum_{\substack{a,b \in V_T^2 \\ e \in r_{a,b} \,\forall\, r \in R_{a,b}}} y_{a,b} \leqslant C \cdot \overline{lfr}(e) \tag{5.25}$$

for every edge $e \in E$ with $\overline{lfr}(e) \cdot C < ld(e)$ to (lop). The optimal solution value of this extended (lop) formulation still provides an upper bound for the optimal solution value of (LOP). For the instances nsir and nsar we obtain significantly improved bounds, which finally lead to reasonable gaps (cf. table 5.19)

	(lop)	($\text{LOP}^{x^{\text{lop}}}$)	(lop)+(5.25)	($\text{LOP}^{x^{\text{lop}}+(5.25)}$)	gap
nsir	27065722	20901319	21082656	21054848	0.1%
nsar	33915224	24765845	25151440	24788405	1.5%

Table 5.19: Performance guarantee of a solution of (LOP) provided by (lop) including (5.25)

	1	2	3	4	5	10	20	30	40
upper bound ($1^T y^{\text{lop}}$)	7451191	8662779	9137901	9385358	9556860	9922811	10054649	10068432	10071184
lower bound ($z^{x^{\text{lop}}}$)	7372418	8201901	8397738	8514912	8468055	8315326	7638692	6973674	6522361
gap	2%	6%	9%	10%	13%	19%	32%	44%	54

Table 5.20: Solutions of J^i, $i \in \{1, 2, 3, 4, 5, 10, 20, 30, 40\}$

	50	100	150	200	300	400	500	1000	C
upper bound ($1^T y^{\text{lop}}$)	10071448	10071448	10071448	10071448	10071448	10071448	10071448	10071448	10071448
lower bound ($z^{x^{\text{lop}}}$)	6239132	5563623	5336942	5219682	5150790	5074161	4962578	4909130	4862313
gap	61%	81%	89%	93%	96%	98%	103%	105%	107%

Table 5.21: Solutions of J^i, $i \in \{50, 100, 150, 200, 300, 400, 500, 1000, C\}$

The instances `nsir` and `nsar` clearly point at the limits of the bounding scheme for the (LOP) model provided by (lop). However, the substantial relaxation, described in section 5.5, provides reasonable results due to the fact that $T^{a,b} \ll C$. In this case, the bundle constraints play a minor role in the (LOP) model. In the following computation we represent the quality of the relaxation (lop) for different values of $C/T^{a,b}$. Consider an instance J of the line planning problem. An instance J^i is derived from J with bounds for the line frequency requirement $\underline{lfr}$, $\overline{lfr}$ and train capacity C given by

$$\underline{lfr}^i = i \cdot \underline{lfr} \quad \overline{lfr}^i = i \cdot \overline{lfr} \quad C^i = \left\lceil \frac{C}{i} \right\rceil .$$

We apply this generation of new instances, that focuses on the "violation" of $T^{a,b} \ll C$, to the InterCity network of the Deutsche Bahn AG (cf. tables 5.20 and 5.21). We observe, that the gap significantly increases for larger values of i. On the one hand, the upper bound provided by (lop) increases. On the other hand, for small values i the value $z^{x^{\text{lop}}}$ of the feasible solutions also increases, but for $i \geqslant 30$ the value $z^{x^{\text{lop}}}$ becomes smaller than the feasible solution of J which clearly can be transformed to a feasible solution of J^i of the same value. For $i \geqslant 40$ the upper bound does not change (this upper bound also represents the total number of travelers in the network). This indicates that the set of optimal solutions of (lop) for instance J^i becomes larger by increasing i and the choice of *the* optimal solution provided by the branch-and-bound procedure is completely random. The valuations of these solutions in the (LOP) model significantly differ and hence we obtain worse feasible solutions. Fortunately, for real-world public transportation networks $T^{a,b} \ll C$ represents a reasonable assumption.

Another upper bounding scheme that does not provide a feasible solution of (LOP) in general is based on the linear programming relaxation of (LOP). With massive computer power we have already computed the linear programming relaxation in the first node of the branch-and-bound tree (cf. section 5.4), which gives a better upper bound compared to the (lop) bound for instances `nsir`, `nsar`, and `sbb3`. We can improve the linear programming relaxation by adding the cuts derived in section 5.5 to (LOP). The valid inequalities presented in proposition 5.3 directly apply to the (LOP) formulation. By aggregating and relaxing the inequalities (5.10) of a particular

	nsic	nsir	nsar	dbagic	dbagir	sbb1	sbb2	sbb3
LP relaxation								
in the first B&B node	8206670	21082102	25143387	7433919	6097924	44920	47236	11159
best bound	8206670	21081002	25143387	7429366	6097924	44920	47236	11156
best solution	8206670	21064537	-	7370519	-	44920	47236	11156
gap	0.0%	0.1%	-	0.8%	-	0.0%	0.0%	0.0%
# B&B nodes	1	6	1	22	1	5	10	22
CPU seconds	1	86400	86400	86400	86400	966	47	89

Table 5.22: Computational results with the improved (LOP) model (railroad)

	bvagtram	bvagbus	vbzsbahn	vbztram	vbzbus
LP relaxation					
in the first B&B node	138483	153479	4379994	3210318	4954579
best bound	138483	153479	4356363	3209757	4954579
best solution	138483	-	4356363	3209757	4954579
gap	0.0%	-	0.0%	0.0%	0.0%
# B&B nodes	1	1	54	24	8
CPU seconds	23	86400	532	9830	5

Table 5.23: Computational results with the improved (LOP) model (urban transport)

origin-destination $a,b \in V_T^2$ we derive the valid inequality

$$\sum_{r \in R_{a,b}} y_{r,a,b} \leqslant C \sum_{r \in R_{a,b}} x_r \tag{5.26}$$

for model (LOP). We apply the cuts of proposition 5.5 to (5.26) and (5.9) with $x = \sum_{r \in R_{a,b}} x_r$ and $y = \sum_{r \in R_{a,b}} y_{r,a,b}$ which results in

$$\sum_{r \in R_{a,b}} y_{r,a,b} \leqslant \Delta \sum_{r \in R_{a,b}} x_r \Leftrightarrow \Delta \left\lceil \frac{T^{a,b}}{C} \right\rceil + T^{a,b}, \tag{5.27}$$

with $\Delta = \lfloor T^{a,b}/C \rfloor C + T^{a,b}$. We add cuts (5.27) and cuts of proposition 5.3 to (LOP) and run this improved formulation again with a time limit of 24 CPU hours on the SGI Power Challenge (cf. tables 5.22 and 5.23). Finally, tables 5.24 and 5.25 present the best known solutions and bounds of model (LOP) provided by the various bounding methods.

	nsic	nsir	nsar	dbagic	dbagir	sbb1	sbb2	sbb3
best bound	8206670	21081002	25143387	7429366	6097010	44920	47236	11156
best solution	8206670	21064537	24788405	7372418	6097010	44920	47236	11156
gap	0.0%	0.1%	1.4%	0.8%	0.0%	0.0%	0.0%	0.0%

Table 5.24: Best known feasible solution and bound of model (LOP) (railroad)

	bvagtram	bvagbus	vbzsbahn	vbztram	vbzbus
best bound	138483	153479	4356363	3209757	4954579
best solution	138483	153479	4356363	3209757	4954579
gap	0.0%	0.0%	0.0%	0.0%	0.0%

Table 5.25: Best known feasible solution and bound of model (LOP) (urban transport)

5.8 Extensions of the models

5.8.1 The software LOP

The approach of line planning presented in section 5.5 and 5.7 is implemented in a computer program named LOP. Given the necessary data (infrastructure and traveler data) LOP computes a feasible solution, i.e. a line plan, and the performance guarantee of this solution. Furthermore, LOP provides the user with a simple graphical user interface (cf. figure 5.9) to visualize the data instance and the computed line plan. LOP is based on the CPLEX callable library for solving the (mixed) integer linear programs. A binary distribution for several UNIX based platforms can be downloaded from the web page of the LOP project[4]. Moreover, we implemented a Web interface to LOP that allows a simple access to the optimization results.

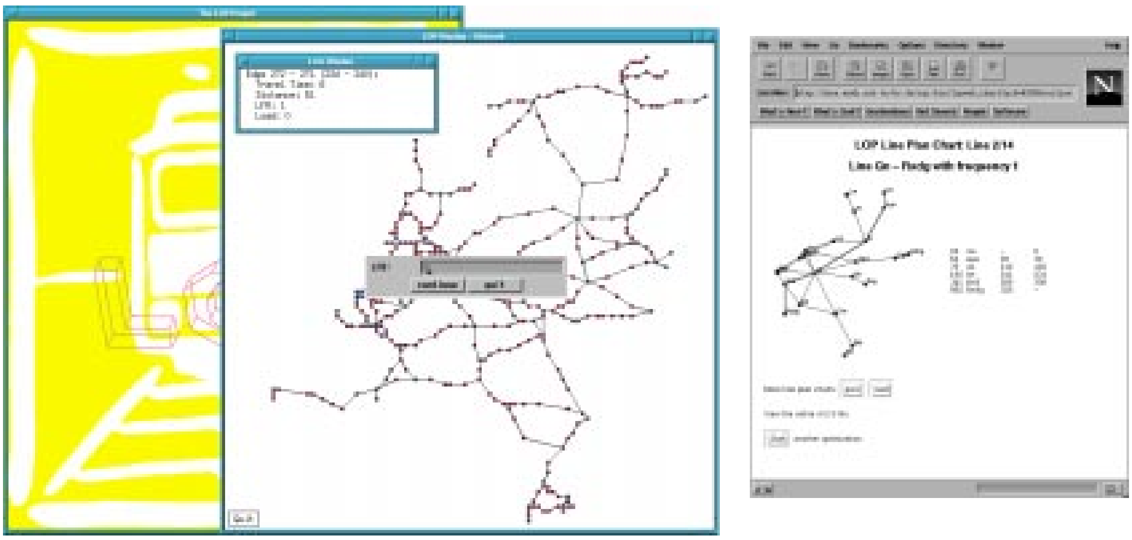


Figure 5.9: Screenshots of the LOP program and the Web interface

5.8.2 A weighted version of (LOP) and (lop)

In the models (LOP) and (lop) we focus on determining a line plan that permits a maximum *number* of direct travelers. The contribution of travelers to the objective is independent of the associated origin-destination pair a, b. A different *weight* of travelers in the objective is of valuable

[4] http://www.math.tu-bs.de/mo/projects/lop/lop.html

interest. For instance, a weight related to the distance between a and b attaches more importance to far distant origin-destination pairs. Moreover, in the (LOP) model we can penalize travel paths which differ from the shortest travel path by assigning a weight

$$w_{r,a,b} = \frac{\text{length of the shortest path connecting } a \text{ and } b}{\text{length of } r_{a,b}}$$

to the variable $y_{r,a,b}$. A careful transformation of these weights to the (lop) model is necessary to keep the upper bound property of an optimal solution of (lop). A general and straight forward transformation that obeys this property is

$$w_{a,b} = \max_{r \in R_{a,b}} w_{r,a,b}.$$

5.8.3 Flexibility versus hardness

In section 4.6 we already present an example of the flexibility of the path formulation for line planning. The reasonable solution times of the improved (lop) model significantly depend on the set of feasible line plans $\mathbb{L}$ and its linear representation $\{x \mid Ax \leqslant b,\ x \in \mathbb{Z}_+^{|R|}\}$. This becomes evident by the following example.

Consider a supply network $G = (V, E)$, bounds for the line frequency requirement $\underline{lfr}$, $\overline{lfr}$, the set of routes R, and the volume of traffic T. With the objective that maximizes the number of direct travelers it is obvious, that the frequencies of lines running via a particular edge $e \in E$ reside close to $\overline{lfr}(e)$. With a generous computation of the $\overline{lfr}$ values, the total line plan may contain too many lines. Due to the flexibility of the model, we can easily overcome this problem by adding one single constraint. The inequality

$$\sum_{r \in R} x_r \leqslant \lambda \tag{5.28}$$

excludes line plans exceeding λ lines (with respect to the frequency). Moreover, we can assign the length of the routes in kilometers to the x variables and can bound the overall length of a feasible line plan. We apply this extension to the (lop) model and solve the `dbagic` instance. The computation times are summarized in table 5.26. The influence of the addition of inequality (5.28) to (lop) dramatically increases the solution time. This increase reflects the theoretical hardness of a sensitivity analysis for (mixed) integer linear programs. One single constraint (the same holds for new variables or changes of coefficients) can destroy the *good nature* of an integer linear program, which becomes perfectly clear by the example of the 0/1 knapsack problem. By

	(lop)	(lop) + (5.28)
# B&B nodes	1	1717
CPU seconds	7.31	733.86

Table 5.26: The inequality (5.28) and the performance of the branch-and-bound algorithm

adding $a^T x \leqslant b$, the formerly trivially solvable program $\max\{c^T x \mid x \in \{0,1\}^n\}$ becomes (NP) hard.

The polyhedral analysis of the (lop) model and the computational investigations point to the importance of the linear description of $\operatorname{conv}\mathbb{L}$. If a particular linear formulation $\{x \mid Ax \leqslant b,\ x \in \mathbb{Z}_+^{|R|}\}$ of $\mathbb{L}$ leads to unacceptable computation times we might try to improve this formulation using the well known techniques of preprocessing and constraint generation.

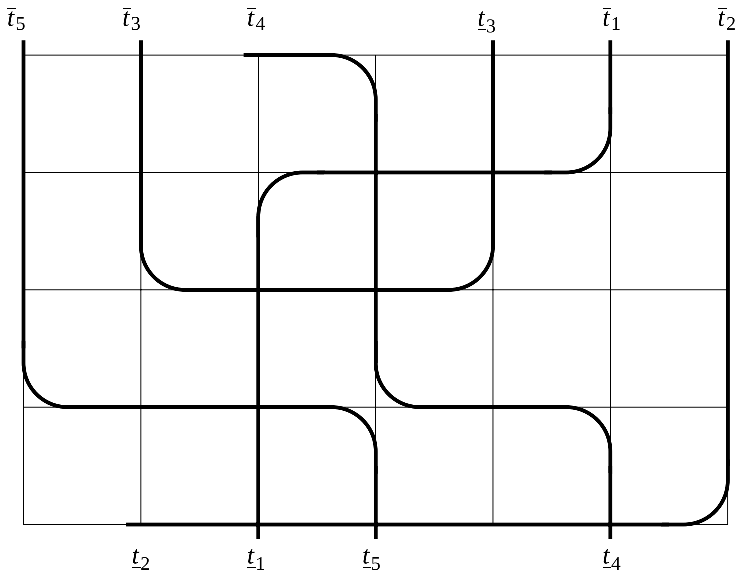


Figure 5.10: A 5×7 grid graph with a packing of path including a knock-knee

5.8.4 Related problems

The path formulation of the feasible line planning problem may be of valuable interest for other applications in the context of routing and design. For a particular *VLSI* (very large scale integration) chip layout problem we briefly discuss an adaption of the line planning formulation. Consider a *grid graph* G (a 5×7 grid graph is depicted in figure 5.10) and pairs of terminal nodes $(\underline{t}_1, \bar{t}_1), \dots, (\underline{t}_k, \bar{t}_k)$ on the outer face of G. The *edge disjoint path packing problem* consists of finding simple edge disjoint paths connecting the nodes of the terminal pairs. For chip layout problems with two writing layers, one for vertical writing and one for horizontal writing (for details cf. [46]), *knock-knees* (cf. figure 5.11) cannot be realized and therefore must be forbidden in feasible path packings. Obviously, this problem can be formulated using the line planning

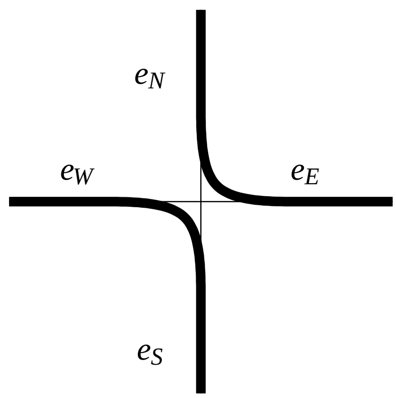


Figure 5.11: A knock-knee

notation. The supply network is given by the grid graph $G=(V,E)$ with $\underline{lfr}\equiv 0$ and $\overline{lfr}\equiv 1$. The set of possible routes R contains all paths connecting $\underline{t}_i$ and $\bar{t}_i$. With the additional constraints

$$\sum_{\substack{r\in R\\ \{e_N,e_E\}\in r}} x_r + \sum_{\substack{r\in R\\ \{e_S,e_W\}\in r}} x_r \leqslant 1 \quad \text{and} \quad \sum_{\substack{r\in R\\ \{e_N,e_W\}\in r}} x_r + \sum_{\substack{r\in R\\ \{e_S,e_E\}\in r}} x_r \leqslant 1 \tag{5.29}$$

where e_N, e_E, e_S, e_W represent incident edges of an inner node of G (the index $X \in \{N(\text{orth}),\ldots,W(\text{est})\}$ of e_X corresponds to the direction of the edge in the grid), we avoid knock-knees in the layout. Now, the edge disjoint path packing problem without knock-knees can be formulated as follows.

$$\max\{1^T x \mid \sum_{\substack{r\in R\\ r\ni e}} x_r \leqslant 1,\ \sum_{\substack{r\in R\\ r=\underline{t}_i\ldots\bar{t}_i}} x_r \leqslant 1, (5.29), x\in\{0,1\}^{|R|}\}$$

The *synchronization of traffic lights* for urban road networks also provides a potential application of the line planning problem. The synchronization of traffic lights with respect to minimum waiting times is closely related to the train schedule planning for railroad systems. In road networks a sequence of traffic lights controlled by a *progressive signal system* (in German: Grüne Welle) provides an analogy of a railroad line. A car with an appropriate velocity that passes this sequence does not need to stop for reasons of a red light. A car that leaves the sequence of suitable controlled traffic lights by turning off, represents a traveler who changes lines. Given a supply network defined by the junctions and road segments with a volume of traffic defined by an origin-destination matrix, we can easily formulate the problem of finding a sequence of lights controlled by a progressive signal system by using the notation of railroad line planning. The set of routes R contains all eligible paths of the road network (excluding paths with forbidden turnings). With the bound for the line frequency requirement set to $\underline{lfr}\equiv 0$ and $\overline{lfr}\equiv 1$, each line of the resulting line plan corresponds to a sequence of traffic lights. With an average velocity, which is a reasonable assumption for an urban road network, we can calculate the traffic light sequence of adjacent traffic lights. If we apply the line planning with respect to the number of direct travelers, we obtain a progressive signal system that provides a thoroughfare for a maximal number of cars. As mentioned above, the synchronization of traffic lights in different lines is closely related to train scheduling problem of railroad planning and is the subject of a paper by PASCOLO et. al. [60].

The examples presented above indicate a potential line planning formulation of the associated problem but obviously do not claim to be superior to well studied special algorithms particularly for the VLSI problem.

In the computational investigations of the (LOP) and (lop) models, the set of possible routes consists of routes on shortest paths. Some of the lines in the resulting line plan can be linked in order to increase the number of direct travelers (cf. figure 5.12). In general, there are several possible combinations of lines. The *optimal* combination of lines with respect to a maximal increase of the number of direct travelers again can be formulated in the line planning notation. We construct a new instance of the line planning problem based on a given line plan $L=\{(r_i,\varphi_i)\mid$

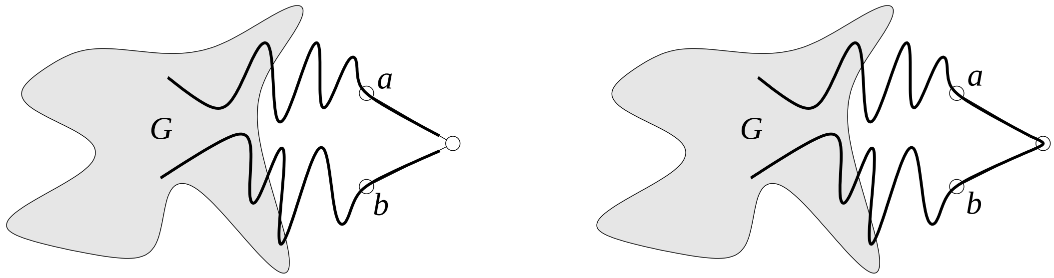


Figure 5.12: A linking of lines that provides a direct connection for origin-destination pair a,b

$r_i \in \mathcal{R}, \varphi_i \in \mathbb{Z}_+, i = 1,\dots,k\}$. The original lines represent the nodes of the new supply network G'. The nodes (r_i,φ_i) and (r_j,φ_j) are connected by an edge in G' if the terminal nodes of r_i and r_j coincide and $\varphi_i = \varphi_j$. The set of possible routes $\mathcal{R}'$ consists of all eligible *linkings* of original lines. With $\underline{lfr} \equiv 0$, $\overline{lfr} \equiv 1$, and the additional constraints

$$\sum_{\substack{r\in\mathcal{R}'\\(r_i,\varphi_i)\in r}} x_r \leqslant 1 \quad \text{for all } (r_i,\varphi_i) \in L$$

to exclude combinations that use an original line more than once, the new line plan given by x represents a suitable combination of lines. We focus on a combination with a maximal increase of direct travelers, therefore we include origin-destination pairs a,b which do not have a direct connection in the original line plan. The terminal stations of the origin-destination pair a,b are no longer included in G' but with $\mathcal{R}'_{a,b}$ consisting of routes in $\mathcal{R}'$ that provide a linking of original lines with a pleasant connection for a,b, we can establish the (LOP) respectively the (lop) model.

5.8.5 Line planning with delayed column generation

In the latter section we discuss the linking of lines after the optimization of the (LOP) and (lop) models with routes on shortest paths. Furthermore, we may increase the objective by adding the routes resulting from possible linking to the set $\mathcal{R}$ and then solve the model. The combination results in a set $\mathcal{R}$ with significantly increased cardinality. If we apply a linking of routes of $\mathcal{R}$ for the second or third time, the size of the resulting $\mathcal{R}$ might be too large to handle all variables x_r in the model explicitly. Moreover, we might define the possible routes by arbitrary (simple) paths of the supply network G. The size of the associated model exceeds any (memory) resource limits even for small supply networks (e.g. the supply network of `nsir` contains more than $2.5 \cdot 10^9$ different simple paths), because $|\mathcal{R}|$ grows exponentially with the size of G.

In the following discussion we develop the fundamentals of a branch-and-price algorithm for particular instances of the (lop) model. The number of y variables is always bounded by $|V|^2$ and can explicitly remain in the (lop) formulation. In general, the cardinality of the set of possible routes $\mathcal{R}$ and hence the number of x variables may exponentially grow with the size of the supply network. Therefore, we concentrate on a dynamical generation of x variables. Before we go into the details of the algorithm, we focus on the pricing problem of the linear programming relaxation of (lop). In the simplex method the *reduced cost* of a variable x_r is given by

$$\tilde{c}_r = \sum_{e \in r} (\underline{\pi}_e \Leftrightarrow \overline{\pi}_e) + C \sum_{\substack{a,b \in V_T^2 \\ R_{a,b} \ni r}} \sigma_{a,b} \tag{5.30}$$

where $\underline{\pi}_e \geqslant 0$, $\overline{\pi}_e \geqslant 0$, and $\sigma_{a,b} \geqslant 0$ represent the *dual variables* corresponding to inequalities (5.14), (5.15), and (5.17). The pricing problem for the x variables consists of finding a route $r \in \mathcal{R}$ with $\tilde{c}_r > 0$ or establishing its nonexistence.

We suggest a solution approach for the pricing problem and give a compatible branching rule for instances of the line planning problem with a set of possible routes $\mathcal{R}$ consisting of *all* simple paths of the supply network connecting two classification yards. Furthermore, the set $R_{a,b}$ is implicitly given by

$$R_{a,b} = \left\{ r \in \mathcal{R} \mid a,b \in r,\ \frac{\text{length of } r_{a,b}}{\text{length of shortest path connecting } a \text{ and } b} \leqslant \alpha \right\} \tag{5.31}$$

with $\alpha \geqslant 1$. Even for this particular class of instances, there is no hope (unless $\mathcal{P}=\mathcal{NP}$) for an efficient pricing algorithm. Such an algorithm would solve the LONGEST PATH problem, which is known to be $\mathcal{NP}$-complete [35]:

Consider an instance of the *longest path problem*, given by a graph $G = (V,E)$, two nodes s and t, and an integer K. Is there a simple path in G connecting s and t that contains at least K edges? There is an obvious polynomially transformation of a longest path instance to an instance of the particular pricing problem. We just have to introduce a copy s' of node s and link s and s' by an edge. The set of classification yards contains s' and t, only. With $\sigma \equiv 0$, $\underline{\pi}_{s's} = 0$, $\overline{\pi}_{s's} = K \Leftrightarrow 1$, and $\underline{\pi}_e = 1$, $\overline{\pi}_e = 0$ for all $e \in E$, we easily derive that G contains a path of at least K edges connecting s and t if and only if the pricing problem recognizes a column (route) of positive reduced cost.

We propose a binary linear program for the pricing problem. The solution of this program provides a simple path connecting two classification yards of maximum reduced cost defined by (5.30).

First of all, we modify the supply network by adding two nodes s and t and edges sv and vt for all nodes $v \in V' \subset V$ corresponding to classification yards. A path connecting s and t (s-t path) represents a path in G connecting two classification yards or an empty path. We introduce binary variables x_e for all edges (including edges incident with s and t) and binary variables x_v for all nodes $v \in V$. The variables x_e and x_v have value 1 if and only if the s-t path contains e and v. The constraints

$$\sum_{e \in \delta(s)} x_e = 1 \tag{5.32}$$

$$\sum_{e \in \delta(t)} x_e = 1 \tag{5.33}$$

$$\sum_{e \in \delta(v)} x_e = 2x_v \qquad \text{for all } v \in V \tag{5.34}$$

provide a feasible region that corresponds to edge and node vectors that contain a simple *s*-*t* path but also may contain isolated cycles. Therefore, we add the following inequalities to the formulation

$$\sum_{e \in C} x_e \leqslant |C| \Leftrightarrow 1 \qquad \text{for all cycles } C \text{ in } G \tag{5.35}$$

and obtain that every binary solution x satisfying (5.32)-(5.35) represents a simple *s*-*t* path. Conversely, every binary representation of an *s*-*t* path satisfies (5.32)-(5.35). Furthermore, with the variables x_e we can establish the first part of the objective defined by (5.30). In order to build the second part of (5.30) we introduce another class of binary variables $x_{u,v}$ for $u, v \in V_T^2$. A variable $x_{u,v}$ achieves value 1 if the *s*-*t* path contains a suitable travel path for origin-destination pair u, v. Obviously, u and v must be included in the *s*-*t* path, which is guaranteed by

$$x_{u,v} \leqslant x_u \quad \text{and} \quad x_{u,v} \leqslant x_v. \tag{5.36}$$

Furthermore, for $x_{u,v} = 1$ the *s*-*t* path must be included in $R_{u,v}$, therefore we add

$$x_{u,v} \leqslant |W| \Leftrightarrow \sum_{e \in W} x_e \quad \text{for all } u\text{-}v \text{ paths } W \text{ with } \frac{\text{length of } W}{\text{length of shortest } u\text{-}v \text{ path}} > \alpha \tag{5.37}$$

which fixes $x_{u,v}$ to 0 for all *s*-*t* path corresponding to routes r with $u, v \in r$ but $r \notin R_{u,v}$. The objective

$$\max \sum_{e \in E} (\underline{\pi}_e \Leftrightarrow \overline{\pi}_e) x_e + \sum_{u,v \in V_T^2} \sigma_{u,v} \cdot x_{u,v} \tag{5.38}$$

completes the binary formulation of the pricing problem. We refer to the binary problem defined by (5.32)-(5.38) as (PRICE). Note that additional constraints for fixing $x_{u,v}$ to 1 if the *s*-*t* path contains a suitable travel path are superfluous because $\sigma_{u,v} \geqslant 0$.

The binary linear program (PRICE) consists of a tremendous number of constraints of type (5.35) and (5.37), but obviously we can generate these constraints on demand and keep the size of the program manageable. Furthermore, with the variable dichotomy given by $x_r = 0$, $x_r = 1, \ldots$ $x_r = \min_{e \in r} \overline{lfr}(e) =: \varphi^r_{\max}$ for problem partitioning in the branch-and-bound algorithm we derive a compatible branching rule. Let P be the problem of a particular node of the branch-and-bound tree, then we obtain a partitioning based on the variable dichotomy of x_r given by

$$P_0 = P + (x_r = 0), \;\; P_1 = P + (x_r = 1), \; \ldots, \; P_{\varphi^r_{\max}} = P + (x_r = \varphi^r_{\max}).$$

The problems P_i, $i = 0, \ldots, \varphi^r_{\max}$ can be processed in the following way. For each $a, b \in V_T^2$ with $r \in R_{a,b}$ all line plans in the feasible region of problem P_i provide a direct connection for $\min\{T^{a,b}, C \cdot i\}$ travelers. Hence we can ignore these travelers for subsequent considerations and prohibit a generation of x_r in the pricing problem by adding

$$x_{ss_r} + x_{tt_r} + \sum_{e \in r} x_e \leqslant |r| + 1$$

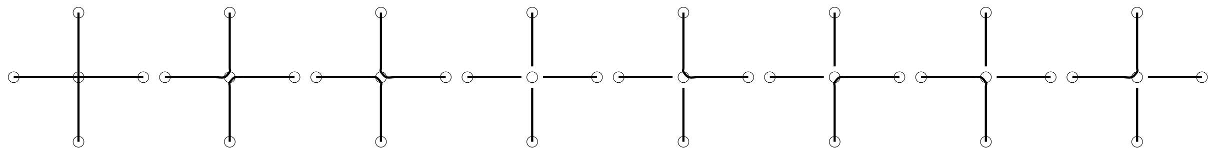


Figure 5.13: Eight alternatives for line plans in the neighborhood of a node with degree 4.

to (PRICE) where s_r and t_r represent the terminal nodes of r.

For the particular case $\underline{lfr} = \overline{lfr} \equiv 1$ we present another compatible branching rule based on problem partitioning corresponding to node splittings. Before we go into the details, we refer to a branch-and-price algorithm for set partitioning problems [8]. In the case $|\{x_r \mid r \in R_{a,b}\}| \leqslant 1$ for all $a, b \in V_T^2$ and for all feasible line plans $x \in \mathbb{L}$, the y variables are superfluous (cf. section 5.6) and hence the line optimization problem with $\underline{lfr} = \overline{lfr} \equiv 1$ becomes a set partitioning problem. BARNHART et. al. [8] present a branching scheme based on submatrix elimination that iteratively forces the constraint matrix of active variables to be *totally balanced* [52], which results in an integer solutions of the linear programming relaxation.

Our branching rule also works for instances with $|\{x_r \mid r \in R_{a,b}\}| > 1$. Consider a node $u \in V$ of degree 4 corresponding to a classification yard. The part of any feasible line plan in the neighborhood of u is represented by one of the alternatives depicted in figure 5.13. Therefore, we can replace the problem partitioning based on variable dichotomy by a branching on the structure of a line plan in the neighborhood of a particular node. Obviously, this branching rule provides a partitioning of the feasible region of P. With the depth-first-search node selection scheme the line plan is uniquely determined after $|V|$ branching steps. Hence, the depth of the branch-and-bound tree is at most $|V|$. Now, how can this branching be combined with the pricing problem? Consider a problem P of a node of the branch-and-bound tree with a feasible region corresponding to line plans that have the structure depicted in figure 5.14 in the neighborhood of $u \in V$. We have to guarantee, that the variable x_r generated by the pricing model must satisfy one of the four alternatives.

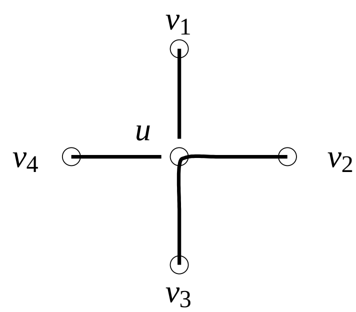


Figure 5.14: One of the eight alternatives

0. r does not contain u.

1. r terminates at u with $r = \ldots v_1 u$.

2. r terminates at u with $r = \ldots v_4 u$.

3. u is an inner node of r with $r = \ldots v_2 u v_3 \ldots$.

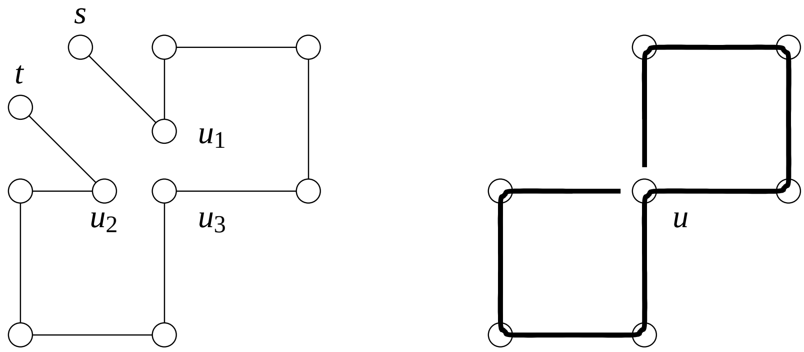


Figure 5.16: A simple *s*-*t* path that corresponds to a walk in G with node repetition.

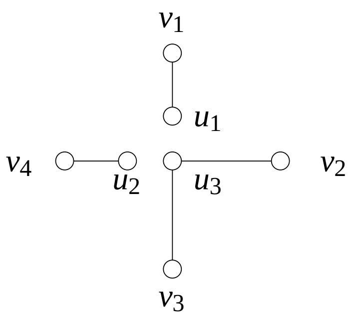


Figure 5.15: The splitting of u into u_1, u_2, and u_3

We can modify the problem data given by G and the origin destination matrix T in order to apply the almost unchanged model (PRICE) for the pricing problem. For the particular instance, we split the node u into three independent nodes u_1, u_2, and u_3 as depicted in figure 5.15. Each node u_i corresponds to one of the alternatives 1-3. Furthermore, we replace the origin-destination pairs $u,v \in V_T^2$ by pairs u_1,v, u_2,v, and u_3,v with an amount of $T^{u_i,v} = T^{u,v}$ travelers for $i = 1,2,3$. Finally, we remove the node u_3 from the set of classification yards. Hence, a simple *s*-*t* path in the graph associated with the model (PRICE) corresponds to a walk r in G that satisfies one of the alternatives 0-3. Unfortunately, the walk r is not necessarily a simple path in G (cf. figure 5.16). With the addition of the special ordered set constraint

$$x_{u_1} + x_{u_2} + x_{u_3} \leqslant 1$$

to the (PRICE) model, we exclude walks with node repetition in G and can apply the solution of (PRICE) to generate new variables.

In this section we developed a pricing model and a compatible branching rule for the *pure* (lop) model, but we can easily integrate the cuts introduced in corollary 5.6. The dual variables of these inequalities result in additional cost coefficients for the $x_{u,v}$ variables. Moreover, with $E' = \delta(v)$ we also may include cuts of proposition 5.3. The dual variables of these inequalities provide cost coefficients for some x_u variables corresponding to nodes $v \in V$ in model (PRICE).

Even if we succeed in solving the hard pricing problem, the costs (in terms of computation time) of a branch-and-price approach must be compared with its benefit. In particular, for real world railroad networks with travel path acceptance factor α close to 1, the branch-and-price approach must give reasonable gains to accept the computational difficulties. For the instance `nsic` we compare the increase of the objective of the (lop) model when switching from at most $|V|^2$ shortest path routes to *all* simple path routes with $R_{a,b}$ given by (5.31). Table 5.27 presents the values of the objective function for different values of α. For $\alpha = 1$ the gain of the instance with routes on all path compared to the instance with routes on shortest paths (objective: 8206670) is less than 0.3% and can be further reduced by applying the linking of lines presented in section 5.8.4. For values of α which exceed any practical limit of a suitable travel path for far distance networks, the gain is less that 6.0% and if $a, b \in r$ is the only requirement for a suitable travel path ($\alpha = \infty$), the gain amounts 7.4%.

	$\alpha = 1.0$	1.1	1.2	1.3	1.4	1.5
objective	8228534	8271831	8288514	8322062	8351004	8408412
gain	0.3	0.8	1.0	1.4	1.8	2.5

	1.6	1.7	1.8	1.9	2.0	$\alpha = \infty$
objective	8482280	8500174	8554428	8584312	8691464	8810707
gain	3.4	3.6	4.2	4.6	5.9	7.4

Table 5.27: Results for the `nsic` instance of (lop) with different values of α.

Chapter 6

Cost optimal line plans

6.1 Introduction

The underlying motivation for line planning with respect to direct travelers was to minimize the inconvenience for passengers in the transportation system, which is estimated by the number of changes. The direct traveler approach, described in the previous chapter, results in a line plan L^* that provides a direct connection for a maximum number of passengers D^*. It is obvious that this line plane is not necessarily *optimal* with respect to the total number of changes. However, a line plan $\hat{L}$ that gives a minimum number of changes $C(\hat{L})$ satisfies

$$\sum_{a,b} T^{a,b} \Leftrightarrow D^* \leqslant C(\hat{L}) \leqslant C(L^*).$$

Even a small gap does not imply a small number of train changes for all travelers, there may well be unacceptable large numbers of train changes for minorities among the travelers. Furthermore, the direct traveler approach often results in lines on *long* routes, where the notion of long routes refers to the number of tracks/edges in the route. Due to the fixed capacity of trains the load of the trains substantially differs along the route. Even if we adjust the capacity of the line in a subsequent simulation, the capacity and hence the number of coaches for the trains is specified by the track with maximum load. Therefore, long lines may lead to a substantial amount of unused train capacity at less busy tracks and thus can be rather costly. Due to the process of privatization of public transportation companies which enforces the efficient utilization of resources, aspects of cost optimal line planning are coming up.

The problem of cost optimal line planning was introduced by CLAESSENS [22] in cooperation with the Dutch railroad company Nederlandse Spoorwegen (NS) and Railned[1]. The problem permits a straightforward *integer nonlinear formulation* (section 6.3). CLAESSENS, VAN DIJK, and ZWANEVELD [23] propose a linearization of the nonlinear model which results in a huge binary linear program. In section 6.4 we discuss this model and the associated algorithm.

[1]Railned is a state organization responsible for capacity planning, management of the infrastructure and for railroad safety.

CLAESSENS et. al. [23] succeed in solving a real-world instance arising from a subnet of the Dutch railroad network (cf. figure 6.1) in approximately one hour on a SUN LX workstation. The model and the algorithm is implemented using GAMS/CPLEX [11, 34]. However, for large real-world data the algorithm provides solutions with a performance guarantee of about 15% in spite of massive computer power [51], which is not accepted by the practitioners of NS. In section 6.5 we give another linearization of the nonlinear formulation of CLAESSENS based on an integer linear program. A reformulation of the program using strong cuts and problem specific preprocessing techniques is applied to four large real-world instances provided by NS. Based on these instances we discuss both linearizations and report about computational investigations. The last section of this chapter concerns the simultaneous line planning of the different supply networks with respect to cost objectives and reviews the practical aspects of cost optimal line planning.

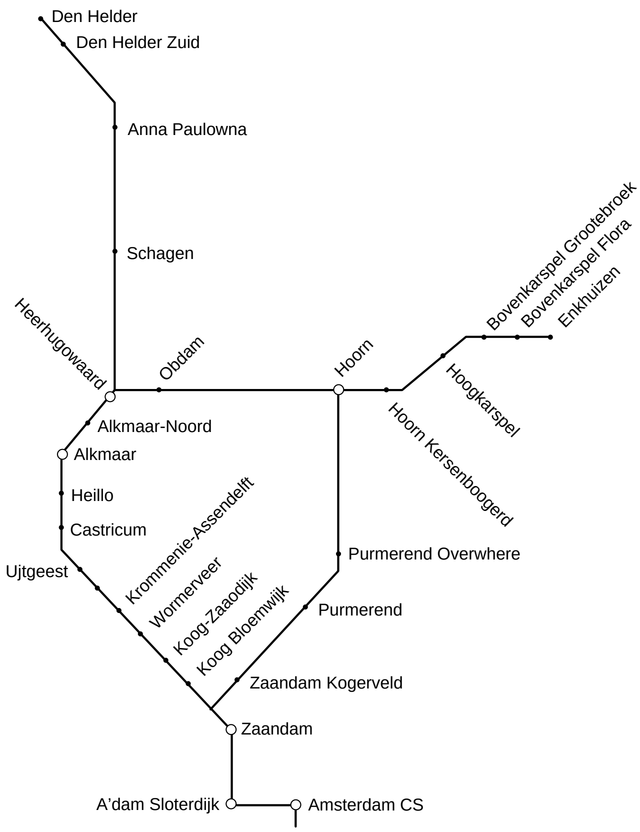


Figure 6.1: Subnetwork in the north-western part of the Netherlands

6.2 Problem description

The overall cost of a transportation system is primarily based on the dispatch of personnel and rolling material. The cost optimal line planning problem focuses on the latter. In contrast to the direct traveler approach with a fixed train capacity, in the cost approach we also determine the number of coaches per train, i.e. the capacity of the operating trains. For each line l in the line plan we compute the number of coaches in the trains serving l. The number of coaches is assumed to be identical for each train serving line l. These identical trains are called *compositions*. A trivial circulation of rolling stock where trains are used for one particular line only (known as *tram-formula*) permits the calculation of operational cost arising from a particular line plan with associated train capacities. The cost are divided into the following categories.

- Fixed cost per coach and motor unit including depreciation cost, capital cost, fixed maintenance cost, and cost of overnight parking.

- Variable cost per coach and motor unit including energy and maintenance cost.

Hence the cost of one particular train on route r with c coaches is determined as follows.

$$\mathfrak{c}^{\mathrm{t_{fix}}} + c \cdot \mathfrak{c}^{\mathrm{c_{fix}}} + d_r \cdot (\mathfrak{c}^{\mathrm{t_{var}}} + c \cdot \mathfrak{c}^{\mathrm{c_{var}}})$$

$\mathfrak{c}^{\mathrm{t_{fix}}}$ ($\mathfrak{c}^{\mathrm{c_{fix}}}$) denotes the fixed cost of a motor unit (coach), $\mathfrak{c}^{\mathrm{t_{var}}}$ ($\mathfrak{c}^{\mathrm{c_{var}}}$) represents the cost of a motor unit (coach) per kilometer, and d_r denotes the length (in kilometers) of route r. With respect to the simple circulation scheme the number of compositions that are necessary to operate a line l on route r with frequency φ within the basic time interval $[0,\ldots,\tau)$ can be determined. The number of necessary compositions depends on the running time between the origin and destination of r, the frequency φ, and the minimum *turn-around time*. This turn-around time is needed for cleaning the train, maintenance, and changing of the crew. The running time plus the turn-around time is divided by τ to obtain the number of compositions Γ_r needed for operating a line *once* per basic time interval. The multiplication of Γ_r and frequency φ with subsequent rounding up to the smallest integer above results in the total number of trains needed for serving line (r,φ). Hence, the operational cost of a line $l = (r,\varphi)$ with c coaches can be roughly estimated by

$$\lceil \varphi \cdot \Gamma_r \rceil (\mathfrak{c}^{\mathrm{t_{fix}}} + c \cdot \mathfrak{c}^{\mathrm{c_{fix}}}) + d_r \cdot \varphi \cdot (\mathfrak{c}^{\mathrm{t_{var}}} + c \cdot \mathfrak{c}^{\mathrm{c_{var}}}). \tag{6.1}$$

The remaining problem characteristics include the parameters of the generic line planning problem including the supply network $G = (V,E)$, the set of possible routes $\mathcal{R}$, the bounds $\underline{lfr}$, $\overline{lfr}$ corresponding to the line frequency requirement and the traffic load ld. Besides the fixed and variable costs introduced above, we have the capacity c_{cap} of a single coach. Furthermore, the number of coaches in an operating line are bounded from below and above by $\underline{c}$ respectively $\overline{c}$. The problem of cost optimal line planning consists of finding a set of lines $L \subset \mathcal{L}$ with a corresponding number of coaches so that the resulting line plan provides sufficient capacity to satisfy the traffic load $ld(e)$ of each edge $e \in E$ and is minimal with respect to the total cost given in terms of (6.1).

6.3 A nonlinear formulation

The cost optimal line planning problem permits an obvious formulation based on an *integer nonlinear program.* We introduce two classes of integer variables. For each possible route $r \in \mathcal{R}$ we have a variable $x_r \in \mathbb{Z}_+$ which denotes the frequency of r and represents the resulting line plan $\{(r,x_r) \mid r \in \mathcal{R}, x_r > 0\}$. Furthermore, $y_r \in \mathbb{Z}_+$ represents the number of coaches per train serving line (r,x_r). Similar to the generic line planning problem, the frequencies of a line plan must satisfy the bounds $\underline{lfr}$, $\overline{lfr}$. In addition to the natural upper bound for the frequency of a line on route r given by $\min_{e \in r} \overline{lfr}(e)$ we introduce a global upper bound $\varphi_{\max}$. This reflects the current policy at NS. InterCity and InterRegio lines have a minimum cycle time of 30 minutes whereas the minimum cycle time of AggloRegio lines is 15 minutes. Together with a basic time interval of 60 minutes this results in $\varphi_{\max} = 2$ for IC and IR supply networks and $\varphi_{\max} = 4$ for

AR networks and provides a set of frequencies $F_0 = \{0, 1, \dots, \varphi_{\max}\}$. Similar to the generic line planning problem and the direct traveler approach a line plan $L = \{(r, x_r) \mid r \in R\,,\, x_r \in F_0\}$ must satisfy the following inequality for each edge $e \in E$.

$$\underline{lfr}(e) \leqslant \sum_{r \in R\,,\, r \ni e} x_r \leqslant \overline{lfr}(e)$$

In contrast to the direct traveler approach with a fixed train capacity, the capacity, i.e. the number of coaches per train is determined by the model itself, in the cost oriented line planning problem. Hence the lower bound $\underline{lfr}$ does not necessarily result in a line plan with capacity that satisfies the traffic load for each edge. Therefore, a feasible line plan must fulfill the following (quadratic) inequality for each edge $e \in E$.

$$\sum_{r \in R\,,\, r \ni e} c_{\text{cap}} \cdot x_r y_r \geqslant ld(e)$$

With respect to the cost structure discussed in the latter section, the objective of a line plan given by $(x, y) \in \mathbb{Z}_+^{2|R|}$ reads as follows.

$$\sum_{r \in R} \lceil x_r \cdot \Gamma_r \rceil \left(\mathfrak{c}^{\text{tfix}} + y_r \cdot \mathfrak{c}^{\text{cfix}}\right) + d_r \cdot x_r \cdot \left(\mathfrak{c}^{\text{tvar}} + y_r \cdot \mathfrak{c}^{\text{cvar}}\right)$$

In all constraints and in the objective the y variables, representing the number of coaches, are multiplied with the corresponding x variables. Hence, we can simply add the constraint

$$\underline{c} \leqslant y_r \leqslant \overline{c}$$

to bound the number of coaches from below and above. A y variable contributes to the objective and the capacity constraint only if the corresponding x variable has a value greater than 0. In order to establish a line plan and the capacity of the lines from a solution of the resulting nonlinear program, we can fix the y_r variables to 0 for $x_r = 0$. Furthermore, we can substitute $y'_r + \underline{c} = y_r$ and transform the non-trivial lower bound $y \geqslant \underline{c}$ to $y' \geqslant 0$.

Summarizing the constraints and the objective we obtain the following integer nonlinear program.

(COSTNLP)

$$\min \quad \sum_{r \in R} \lceil x_r \cdot \Gamma_r \rceil \left(\mathfrak{c}^{\text{tfix}} + (y_r + \underline{c}) \cdot \mathfrak{c}^{\text{cfix}}\right) + d_r \cdot x_r \cdot \left(\mathfrak{c}^{\text{tvar}} + (y_r + \underline{c}) \cdot \mathfrak{c}^{\text{cvar}}\right)$$

$$\text{s.t.} \quad \sum_{r \in R\,,\, r \ni e} x_r \geqslant \underline{lfr}(e) \qquad \forall\, e \in E \tag{6.2}$$

$$\sum_{r \in R\,,\, r \ni e} x_r \leqslant \overline{lfr}(e) \qquad \forall\, e \in E \tag{6.3}$$

$$\sum_{r \in R\,,\, r \ni e} c_{\text{cap}} \cdot x_r (y_r + \underline{c}) \geqslant ld(e) \qquad \forall\, e \in E \tag{6.4}$$

$$y_r \leqslant \overline{c} \Leftrightarrow \underline{c} \qquad \forall\, r \in R \tag{6.5}$$

$$x_r \in F_0, \;\; y_r \in \mathbb{Z}_+ \qquad \forall\, r \in R \tag{6.6}$$

The model has discontinuous terms ($\lceil x_r \cdot \Gamma_r \rceil$) in the objective, quadratic terms in both the objective and in the constraints, and integer variables. Relaxation methods in order to obtain lower bounds within a branch-and-bound algorithms, such as the classical *Lagrangian relaxation* yield poor results [22]. CLAESSENS presents a heuristic based on a relaxation of (COSTNLP) combined with iteratively rounding of x variables. The outline of the algorithm is as follows. A local optimum of the real-valued relaxation of (COSTNLP) is produced with a general nonlinear programming solver. CLAESSENS successfully applied the GAMS/MINOS [11] solver to the real-valued relaxation of a small instance of (COSTNLP). Promising routes, i.e. routes whose product of associated variables $x_r y_r$ exceed a given acceptance level α, get a new lower bound $x_r \geqslant 1$. Some other routes, with $x_r y_r < \beta < \alpha$ are deleted from the problem and the relaxation is solved again. If all routes are deleted or bounded from below by 1 the x variables are rounded upwards to the next integer. The x variables are fixed in the following determination of the y variables. With fixed x variables the problem (COSTNLP) becomes an integer linear program which is related to an integer multi-commodity flow problem. The corresponding recognition problem remains $\mathcal{NP}$-complete.

PROPOSITION 6.1
Given $x \in F_0^{|R|}$ and an integer $\eta \in \mathbb{Z}$. The problem whether there is an $y \in \mathbb{Z}_+^{|R|}$ that is feasible with respect to (6.2)-(6.6) and yields an objective less or equal than η is $\mathcal{NP}$-complete.

PROOF We can polynomially transform the $\mathcal{NP}$-complete FEASIBLE LINE PLAN (FLP) with $\underline{lfr} = \overline{lfr} \equiv 1$ problem (cf. corollary 4.3) to this particular problem. Therefore, we set $F_0 = \{0, 1\}$, $ld(e) = 1$, $\underline{c} = 0$, $\overline{c} = 1$, $\mathfrak{c}^{t_{\text{fix}}} = \mathfrak{c}^{c_{\text{fix}}} = \mathfrak{c}^{t_{\text{var}}} \equiv 0$, $\mathfrak{c}^{c_{\text{var}}} = 1$, and $d_r = |\{e \in E \mid e \in r\}|$. Hence the cost of a route $r \in R$ corresponds to the number of edges in r. It is quite obvious that FLP has a solution if and only if there is a feasible $y \in \mathbb{Z}_+^{|R|}$ with cost equal to $\eta = |E|$. Furthermore, it is easy to see that the recognition problem under consideration belongs to $\mathcal{NP}$.□

Nevertheless, the integer linear program derived from the fixing heuristic could be solved in reasonable time for the particular instance, but the results obtained by the overall procedure are reported to be unsatisfactory [22].

In contrast to the heuristic approach of CLAESSENS which focuses on the fixing of the x variables of (COSTNLP), we may consider the fixing of y variables. With the exception of the discontinuous term in the objective, the fixing of the y variables results in an integer linear program which can be interpreted as follows. The lines have different but fixed capacities and the solution of the program determines a feasible line plan that provides sufficient capacity for each edge at minimum cost. Even if we fix $\underline{lfr} \equiv \overline{lfr} \equiv 1$ which obviously eliminates the discontinuous term, the problem remains $\mathcal{NP}$-complete (cf. corollary 4.3).

6.4 Linearization I

CLAESSENS, VAN DIJK and ZWANEVELD [23] present a linearization of the nonlinear program of the previous section with a tremendous number of binary variables. In the (COSTNLP) formulation all quadratic term have the form $x_r y_r$. A straight forward linearization is obtained by introducing new binary variables $z_{r,\varphi,\gamma}$ with $z_{r,\varphi,\gamma} = 1$, if the line plan contains a line on route r with frequency φ and γ coaches, and $z_{r,\varphi,\gamma} = 0$, otherwise. With $F = F_0 \setminus \{0\}$ the quadratic term $x_r y_r$ is replaced by $\sum_{\varphi\in F} \sum_{\gamma=\underline{c}}^{\overline{c}} \varphi\gamma z_{r,\varphi,\gamma}$ and the terms with separate x_r are replaced by $\sum_{\gamma=\underline{c}}^{\overline{c}} \varphi z_{r,\varphi,\gamma}$. Even the discontinuous term $\lceil \Gamma_r x_r \rceil y_r$ can be replaced by $\sum_{\varphi\in F} \sum_{\gamma=\underline{c}}^{\overline{c}} \lceil \Gamma_r \varphi \rceil \gamma z_{r,\varphi,\gamma}$. The complete formulation derived from this linearization is a binary linear program and reads as follows.

$$
\text{(COSTBLP)} \quad
\begin{aligned}
\min \quad & \sum_{r\in R} \sum_{\varphi\in F} \sum_{\gamma=\underline{c}}^{\overline{c}} \left(\lceil \varphi \cdot \Gamma_r \rceil \left(\mathfrak{c}^{\mathrm{t_{fix}}} + \gamma \cdot \mathfrak{c}^{\mathrm{c_{fix}}} \right) + d_r \cdot \varphi \cdot \left(\mathfrak{c}^{\mathrm{t_{var}}} + \gamma \cdot \mathfrak{c}^{\mathrm{c_{var}}} \right) \right) z_{r,\varphi,\gamma} \\
\text{s.t.} \quad & \sum_{r\in R\,,\, r\ni e} \sum_{\varphi\in F} \sum_{\gamma=\underline{c}}^{\overline{c}} \varphi z_{r,\varphi,\gamma} \geqslant \underline{lfr}(e) \qquad \forall\, e \in E && (6.7) \\
& \sum_{r\in R\,,\, r\ni e} \sum_{\varphi\in F} \sum_{\gamma=\underline{c}}^{\overline{c}} \varphi z_{r,\varphi,\gamma} \leqslant \overline{lfr}(e) \qquad \forall\, e \in E && (6.8) \\
& \sum_{r\in R\,,\, r\ni e} \sum_{\varphi\in F} \sum_{\gamma=\underline{c}}^{\overline{c}} c_{\mathrm{cap}} \cdot \varphi\gamma z_{r,\varphi,\gamma} \geqslant ld(e) \qquad \forall\, e \in E && (6.9) \\
& \sum_{\varphi\in F} \sum_{\gamma=\underline{c}}^{\overline{c}} z_{r,\varphi,\gamma} \leqslant 1 \qquad \forall\, r \in R && (6.10) \\
& z \in \{0,1\}^{|R|\,|F|\cdot(\overline{c}-\underline{c}+1)} && (6.11)
\end{aligned}
$$

Clearly, constraints (6.7)-(6.9) resemble (6.2)-(6.4). With the new set of variables we can delete the constraints (6.5) but we must guarantee that for each route there is at most one combination of frequency and capacity in the line plan, therefore we add constraint (6.10). Compared to the model (COSTNLP) the number of variables dramatically grows in the formulation (COSTBLP). Realistic values for problem parameters are of the following magnitude: $|F| \in \{1,\dots,4\}$, $\underline{c} \in \{2,3\}$, and $\overline{c} \in \{12,\dots,15\}$. For real-world instances the number of variables grows by a factor of 10, whereas the number of constraints keeps unchanged. Even for the small network depicted in figure 6.1 the binary linear program (COSTBLP) consists of 5629 binary variables, 192 constraints, and 111733 non-zeros in the constraints matrix. Therefore, a problem specific preprocessing is applied to an instance of (COSTBLP) in order to reduce the size of the binary linear program before passing it to a general linear programming-based branch-and-bound algorithm.

6.4.1 Reducing the size of the problem

First of all, note that shrinking of nodes (cf. section 5.5.1) can also be applied to networks in the cost approach. If no route terminates at a particular node $v \in V$ of degree 2, we can compose the incident edges e_1, e_2 of v (cf. figure 6.2) and hence reduce the number of constraints in (COSTBLP). The resulting edge $e_{1,2}$ obviously has the following parameters.

$$ld(e_{1,2}) = \max\{ld(e_1), ld(e_2)\},\ \underline{lfr}(e_{1,2}) = \max\{\underline{lfr}(e_1), \underline{lfr}(e_2)\},\ \overline{lfr}(e_{1,2}) = \min\{\overline{lfr}(e_1), \overline{lfr}(e_2)\}$$

In particular, the composition of edges shows to be effective for networks with a substantially smaller number of classification yards compared to the total number of stations, e.g. in Agglo-

Regio networks.

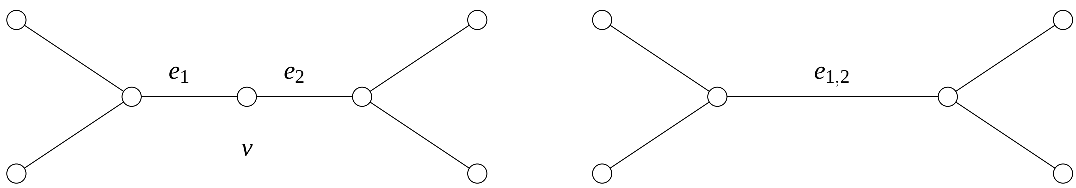


Figure 6.2: Composition of edges

It is quite obvious that variables $z_{r,\varphi,\gamma}$ with $\varphi > \min_{e \in r} \overline{lfr}(e)$ cannot be 1 in any feasible solution. The value of $\varphi^r_{\max} = \min_{e \in r} \overline{lfr}(e)$ was introduced as an implicit lower bound for the frequency of a line on route r, hence we can remove $z_{r,\varphi,\gamma}$ for each $\varphi > \varphi^r_{\max}$ from the binary linear program. The remaining techniques focus on reducing the number of variables by applying dominance rules and implications derived from upper bounds for the number of coaches in an optimum solution.

If the traffic load for all edges of a route r is satisfied by a single line (r, φ^*) with γ coaches, i.e. $c_{\text{cap}} \cdot \varphi\gamma \geqslant \max_{e \in r} ld(e)$, and either no other line nor a larger frequency of r is necessary to fulfill the frequency requirement for each edge $e \in r$, i.e. $\varphi \geqslant \max_{e \in r} \underline{lfr}(e)$ the variables $z_{r,\varphi,\gamma}$ with $\varphi > \varphi^*$ are always 0 in any optimal solution and can be removed from (COSTBLP).

If a line (r, φ) is selected, i.e. $\sum_{\gamma=\underline{c}}^{\overline{c}} z_{r,\varphi,\gamma} = 1$, then in order to fulfill the frequency requirement at least $\max\{0, \underline{lfr}(e) \Leftrightarrow \varphi\}$ other trains of other lines must pass e. Each train consists of at least $\underline{c}$ coaches and transports at least $\max\{0, \underline{lfr}(e) \Leftrightarrow \varphi\}\underline{c} \cdot c_{\text{cap}}$ passengers. Hence there remains an amount of $ld(e) \Leftrightarrow \max\{0, \underline{lfr}(e) \Leftrightarrow \varphi\}\underline{c} \cdot c_{\text{cap}}$ passengers for line (r, φ) and therefore we only need $\lceil (ld(e) \Leftrightarrow \max\{0, \underline{lfr}(e) \Leftrightarrow \varphi\}\underline{c} \cdot c_{\text{cap}})/c_{\text{cap}} \rceil =: \gamma^*_e$ coaches on edge e for line (r, φ). The variables $z_{r,\varphi,\gamma}$ with $\gamma > \max_{e \in r} \gamma^*_r$ are always 0 in an optimal solution and can be removed from the binary linear program.

Another variable elimination scheme is based on a dominance rule. A dominated variable $z_{r,\varphi,\gamma}$ can be replaced by a variable $z_{r,\varphi',\gamma'}$ in any feasible line plan. This replacement also results in a feasible line plan with cost not larger than the original. A variable $z_{r,\varphi,\gamma}$ is dominated by $z_{r,\varphi',\gamma'}$ with $(\varphi, \gamma) \neq (\varphi', \gamma')$ if the following three conditions hold.

First of all the cost coefficient of $z_{r,\varphi',\gamma'}$ must be less or equal than the cost coefficient of $z_{r,\varphi,\gamma}$, i.e.

$$\lceil \varphi' \cdot \Gamma_r \rceil (\mathfrak{c}^{\text{t}_{\text{fix}}} + \gamma' \cdot \mathfrak{c}^{\text{c}_{\text{fix}}}) + d_r \cdot \varphi' \cdot (\mathfrak{c}^{\text{t}_{\text{var}}} + \gamma' \cdot \mathfrak{c}^{\text{c}_{\text{var}}}) \leqslant \lceil \varphi \cdot \Gamma_r \rceil (\mathfrak{c}^{\text{t}_{\text{fix}}} + \gamma \cdot \mathfrak{c}^{\text{c}_{\text{fix}}}) + d_r \cdot \varphi \cdot (\mathfrak{c}^{\text{t}_{\text{var}}} + \gamma \cdot \mathfrak{c}^{\text{c}_{\text{var}}}).$$

Furthermore, the capacity of $z_{r,\varphi',\gamma'}$ is greater of equal than the capacity of $z_{r,\varphi,\gamma}$ or the extra capacity of $z_{r,\varphi,\gamma}$ is superfluous:

$$\varphi\gamma \leqslant \varphi'\gamma' \quad \text{or} \quad \varphi'\gamma' c_{\text{cap}} + \max\{0, \underline{lfr}(e) \Leftrightarrow \varphi'\}\underline{c} \cdot c_{\text{cap}} \geqslant ld(e) \quad \text{for all } e \in r$$

Finally, the line frequency requirement must be satisfied after replacing $z_{r,\varphi,\gamma}$ by $z_{r,\varphi',\gamma'}$. If $\varphi \geqslant \varphi'$ the lower bound $\underline{lfr}$ must be observed, which can be formally expressed by

$$\varphi' + \max\left\{0, \frac{ld(e) \Leftrightarrow \min\{\varphi'\gamma', \varphi\gamma\} c_{\text{cap}}}{\overline{c} \cdot c_{\text{cap}}}\right\} \geqslant \underline{lfr}(e) \quad \text{for all } e \in r.$$

If $\varphi \leqslant \varphi'$, the upper bound $\overline{lfr}$ becomes important, but there is no local condition which gives the feasibility in this case. Extra frequency of other lines may yield cost savings but together with φ' may violate $\overline{lfr}$. Hence the dominance rule is eligible for $\varphi \geqslant \varphi'$ only.

In combination with a general preprocessing scheme introduced in section 4.9.1, the number of variables is reduced by 73% compared to the number of variables in the initial formulation of the instance depicted in figure 6.1.

6.4.2 Improving lower bounds

In section 4.9 we mentioned the importance of good global lower bounds in a branch-and-bound algorithm. The value of the linear programming relaxation corresponding to (COSTBLP) can significantly be increased by tightening parts of the right hand side and adding some cuts to the problem.

The tightening of the right hand side is related to the cuts for the direct traveler model in section 5.5.2. The left hand side of inequality (6.9) gives always $\xi \cdot c_{\text{cap}}$ with $\xi \in \mathbb{Z}_+$. If a feasible solution $z \in \{0,1\}^{|R||F|\cdot(\overline{c}-\underline{c}+1)}$ of (COSTBLP) satisfies (6.9) it also fulfills

$$\sum_{r\in R\,,\, r\ni e}\sum_{\varphi\in F}\sum_{\gamma=\underline{c}}^{\overline{c}} \varphi\gamma z_{r,\varphi,\gamma} \geqslant \left\lceil \frac{ld(e)}{c_{\text{cap}}} \right\rceil =: \widetilde{ld}(e).$$

CLAESSENS et. al. also apply general cuts (clique and cover cuts) mentioned in section 4.9.2 and improve the lower bound by 10% compared to the initial linear programming relaxation value. It is worthwhile to mention that cuts derived from the direct travel model that consists of frequency variables only (e.g. cuts introduced in proposition 5.3) could be directly applied to the model (COSTBLP).

6.4.3 The branch-and-bound algorithm

CLAESSENS, VAN DIJK, and ZWANEVELD solved the preprocessed and tightened binary linear program with a standard linear programming based branch-and-bound algorithm that makes use of special ordered sets which arise from inequalities (6.10) and a particular node selection scheme. The selection of the next subproblem to be investigated is based upon an estimate of the best obtainable integer feasible solution for the subproblem. This estimate is obtained by removing all variables with a fractional value from the objective value of the linear programming relaxation. This node selection is reported to find better feasible solutions compared to the schemes described in section 4.7.1. The authors report about computational investigations for the instance with 28 stations mentioned above. They used CPLEX [25] version 3.0 as a basis for the branch-and-bound algorithm and GAMS [11] for modeling the problem and for implementing the preprocessing techniques. Table 6.1 compares the size and the linear programming relaxation value of the initial and the preprocessed binary linear program. The total preprocessing requires about 10 seconds on a 486DX2-66 PC. Table 6.2 presents the computation time (CPU seconds on a SUN LX-50) of the branch-and-bound algorithm related to the achieved performance guarantee

	variables	constraints	non-zeros	LP relaxation value
initial (COSTBLP)	5629	194	111733	6920
preprocessed (COSTBLP)	1547	139	18192	7577

Table 6.1: Size and linear programming relaxation value of the initial and preprocessed program (COSTBLP) of the instance depicted in figure 6.1

(gap). As usual, most of the solution time is spend on proving optimality of a feasible solution which is found after 850 CPU seconds.

The resulting line plan and the cost saving in comparison to line plans derived from the direct traveler approach are discussed in section 6.7.

6.4.4 Features and limitations of (COSTBLP)

In the following we will see that the binary linear program (COSTBLP) is more flexible than the nonlinear program (COSTNLP). We can easily formulate operational constraints by eliminating variables. In particular the set of possible frequencies and the number of coaches can be adjusted to meet some practical requirements given by Nederlandse Spoorwegen (NS). In the nonlinear program (COSTNLP) as well as in the direct traveler approach the frequency φ of a line (r,φ) is chosen from $0,1,\dots,\varphi_{\max}$. In the AggloRegio network with $\varphi_{\max}=4$, lines with frequency 3 (corresponding to cycle times of 20 minutes) are excluded by the planners at NS. Such *holes* in the set of frequencies can be modeled by additional binary variables and constraints in the (COSTNLP) model. For each route $r \in R$ introduce a new binary variable $u_r \in \{0,1\}$ and the inequalities

$$4u_r \leqslant x_r, \qquad x_r \Leftrightarrow 2u_r \leqslant 2. \tag{6.12}$$

If $u_r = 1$ the frequency x_r is fixed to 4, otherwise ($u_r = 0$) the frequency x_r is bounded from above by 2. Hence (6.12) excludes frequency 3 for lines on route r. In the binary linear program the additional requirement on the frequency can be easily achieved by removing variables $z_{r,3,\gamma}$ from (COSTBLP).

A particularity of the rolling stock of the Nederlandse Spoorwegen also results in a *hole* in the domain of possible coach numbers $\underline{c},\dots,\overline{c}$. Some lines are served by trains that are composed of individual trainsets. Such a trainset named *Koploper* (cf. figure 6.3) consists of one motor coach, middle coaches and a driving trailer. In the front and back of the trainsets there are doors

	LP relaxation	10% gap	5% gap	0% gap
CPU seconds	1	29	77	3989

Table 6.2: Running times of the branch-and-bound algorithm applied to the instance depicted in figure 6.1

underneath the driver's cabin, through which a passageway can be created when trainsets are coupled[2]. This allows passengers to get from one trainset to another. There are 3-coach and 4-coach trainsets in the rolling stock of Nederlandse Spoorwegen. Hence a train that is composed of trainsets consists of $3,4,6,7,8,\ldots$ coaches. With $\underline{c}=3$ we must exclude 5 from the set of possible coach numbers. Similar to the requirement on the frequency we can exclude lines with 5 coaches from the feasible region of (COSTNLP) by additional binary variables and some constraints.

Figure 6.3: Electric trainsets ICM/Plan Z *"Koploper"*

Although the binary linear program is capable of easily including additional operational constraints, the computation time even for the small instance (cf. figure 6.1) is rather large. The solution of the preprocessed formulation (COSTBLP) for the complete Dutch InterCity network `sp97ic` depicted in figure 6.4 provides unacceptable results. The branch-and-bound algorithm (CPLEX version 4.0 on a DEC 633 Alpha) was stopped after the size of the branch-and-bound tree exceeded 415 megabytes and provided a performance guarantee (or gap) of 15% [51].

In the next section we present a new linearization of the nonlinear program which results in an integer linear program of substantially smaller size compared to (COSTBLP).

[2]For technical details we refer to `http://mercurio.iet.unipi.it/ns/4000.html`.

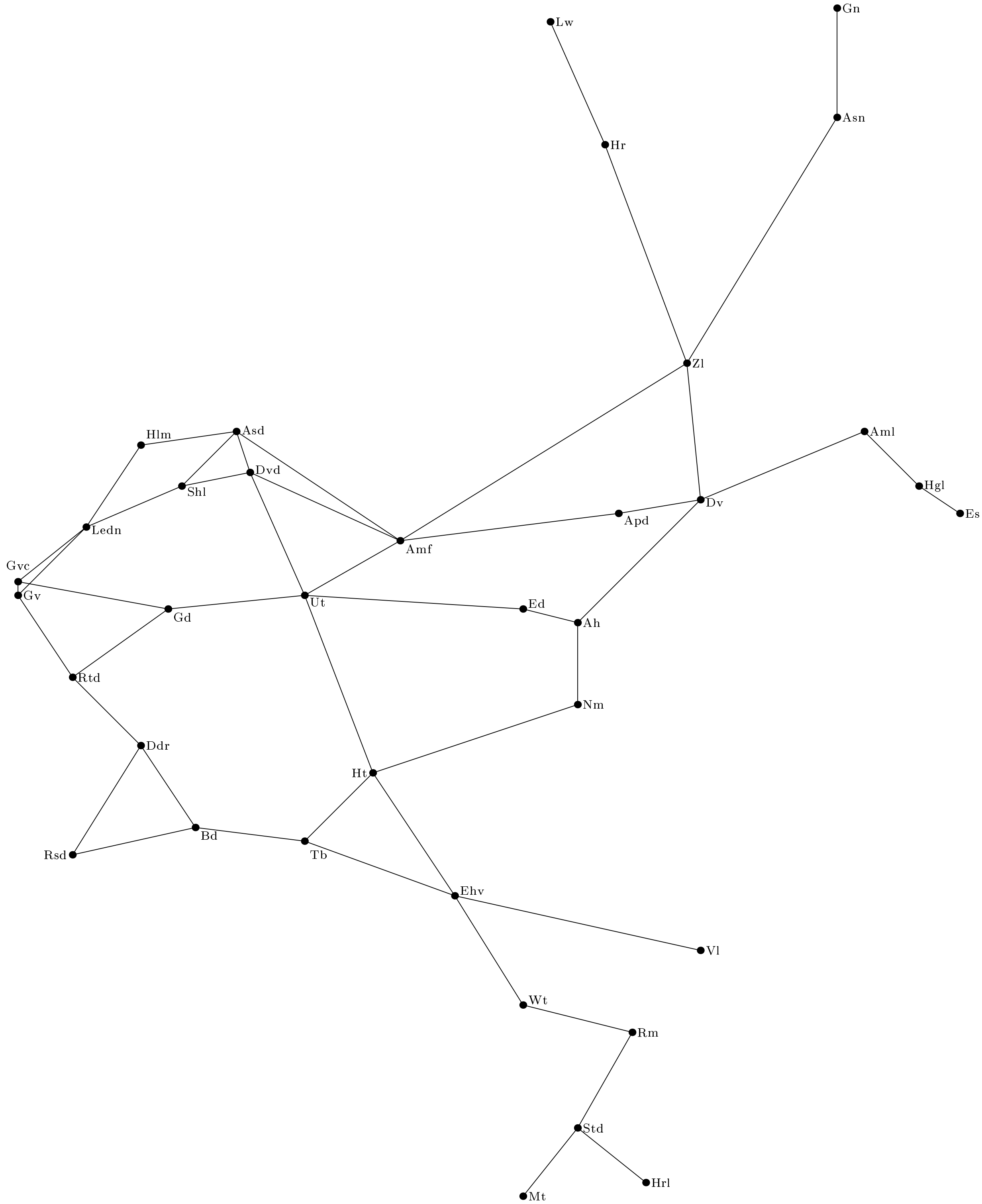


Figure 6.4: The Dutch InterCity network `sp97ic`

6.5 Linearization II

In this section we present a new linearization of the nonlinear program described in section 6.3. Similar to the (COSTNLP) model we have two classes of variables, i.e. the class of variables representing the frequency of a route and the class of variables representing the number of coaches for a particular line. We avoid quadratic terms in the objective as well as in the set of constraints by introducing binary variables for the combination of a particular route and frequency. For each route $r \in \mathcal{R}$ and each frequency $\varphi \in \mathcal{F}$ we have a binary variable $x_{r,\varphi}$ with $x_{r,\varphi} = 1$ if the line plan contains the line on route r with frequency φ and $x_{r,\varphi} = 0$, otherwise. Furthermore, for each route-frequency combination $(r,\varphi) \in \mathcal{R} \times \mathcal{F}$ we have $y_{r,\varphi} \in \mathbb{Z}_+$ representing the number of coaches of the line (r,φ). A quadratic term $x_r y_r$ in the (COSTNLP) model can be replaced by $\sum_{\varphi\in\mathcal{F}} \varphi y_{r,\varphi}$ if we guarantee that $y_{r,\varphi} \geqslant \underline{c}$ if and only if $x_{r,\varphi} = 1$. A separate x_r in the nonlinear program can be substituted by $\sum_{\varphi\in\mathcal{F}} \varphi x_{r,\varphi}$ and the discontinuous term $\lceil x_r \Gamma_r \rceil y_r$ can be replaced by $\sum_{\varphi\in\mathcal{F}} \lceil \varphi \Gamma_r \rceil y_{r,\varphi}$. The complete model reads as follows.

$$\min \quad \sum_{r\in\mathcal{R}} \sum_{\varphi\in\mathcal{F}} \lceil \varphi \cdot \Gamma_r \rceil \left(x_{r,\varphi}\mathfrak{c}^{\mathrm{t_{fix}}} + y_{r,\varphi} \cdot \mathfrak{c}^{\mathrm{c_{fix}}}\right) + d_r \cdot \varphi \cdot \left(x_{r,\varphi}\mathfrak{c}^{\mathrm{t_{var}}} + y_{r,\varphi} \cdot \mathfrak{c}^{\mathrm{c_{var}}}\right)$$

$$\text{s.t.} \quad \sum_{r\in\mathcal{R},\, r\ni e} \sum_{\varphi\in\mathcal{F}} \varphi x_{r,\varphi} \geqslant \underline{lfr}(e) \qquad \forall\, e \in E \tag{6.13}$$

$$\sum_{r\in\mathcal{R},\, r\ni e} \sum_{\varphi\in\mathcal{F}} \varphi x_{r,\varphi} \leqslant \overline{lfr}(e) \qquad \forall\, e \in E \tag{6.14}$$

$$\sum_{r\in\mathcal{R},\, r\ni e} \sum_{\varphi\in\mathcal{F}} c_{\mathrm{cap}} \cdot \varphi y_{r,\varphi} \geqslant ld(e) \qquad \forall\, e \in E \tag{6.15}$$

$$y_{r,\varphi} \leqslant \overline{c} \cdot x_{r,\varphi} \qquad \forall\, r \in \mathcal{R}, \quad \forall\, \varphi \in \mathcal{F} \tag{6.16}$$

$$\underline{c} \cdot x_{r,\varphi} \leqslant y_{r,\varphi} \qquad \forall\, r \in \mathcal{R}, \quad \forall\, \varphi \in \mathcal{F} \tag{6.17}$$

$$\sum_{\varphi\in\mathcal{F}} x_{r,\varphi} \leqslant 1 \qquad \forall\, r \in \mathcal{R} \tag{6.18}$$

$$x \in \{0,1\}^{|\mathcal{R}|\times|\mathcal{F}|}, \quad y \in \mathbb{Z}_+^{|\mathcal{R}|\times|\mathcal{F}|} \tag{6.19}$$

Constraints (6.13)-(6.15) resemble constraints (6.2)-(6.4) of the (COSTNLP) model. In contrast to the (COSTNLP) model the y variables contribute to the objective as well as to the capacity constraint (6.15) without being multiplied with the corresponding x variable. Hence, we have to ensure that $y_{r,\varphi} \in [\underline{c}, \overline{c}]$ if and only if $x_{r,\varphi} = 1$. This is done by the constraints (6.16) and (6.17). Inequality (6.18) resembles constraint (6.10) of (COSTBLP) which guarantees that for each route there is at most one line on route r in the line plan. The binary representation of combinations of routes and frequencies easily permits to model *holes* in the set of frequencies $\mathcal{F}$ by deleting variables, e.g. $x_{r,3}$. The representation of holes in the domain of the y variables must be modeled with additional binary variables and constraints (cf. section 6.4.4). This kind of holes are less important compared to the domain restrictions of the frequencies because the number of coaches in the line plan is adjusted in a subsequent planning step (cf. section 6.7).

Simililarly to the (COSTNLP) model we can save $|\mathcal{R}| \cdot |\mathcal{F}|$ constraints of type (6.17) by an appropriate subsitution of the y variables. The trains of an operating line consists of at least $\underline{c}$

coaches, so let $y_{r,\varphi} \in \mathbb{Z}_+$ be the number of *additional* coaches of line (r,φ). Hence, the total number of coaches of line (r,φ) is $\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}$. If we replace $y_{r,\varphi}$ by $\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}$ in the above formulation, inequality (6.17) reduces to the redundant non-negativity constraint $y_{r,\varphi} \geqslant 0$ and can be deleted. The complete formulation reads as follows.

$$\min \sum_{r\in R}\sum_{\varphi\in F} \lceil \varphi\cdot\Gamma_r \rceil \left(x_{r,\varphi}\mathfrak{c}^{\mathrm{t_{fix}}} + (\underline{c}\cdot x_{r,\varphi}+y_{r,\varphi})\cdot \mathfrak{c}^{\mathrm{c_{fix}}}\right) + d_r\cdot\varphi\cdot\left(x_{r,\varphi}\mathfrak{c}^{\mathrm{t_{var}}} + (\underline{c}\cdot x_{r,\varphi}+y_{r,\varphi})\cdot\mathfrak{c}^{\mathrm{c_{var}}}\right)$$

(COSTILP)

$$\text{s.t.} \quad \sum_{r\in R,\, r\ni e}\sum_{\varphi\in F} \varphi x_{r,\varphi} \geqslant \underline{lfr}(e) \qquad \forall\, e\in E \tag{6.20}$$

$$\sum_{r\in R,\, r\ni e}\sum_{\varphi\in F} \varphi x_{r,\varphi} \leqslant \overline{lfr}(e) \qquad \forall\, e\in E \tag{6.21}$$

$$\sum_{r\in R,\, r\ni e}\sum_{\varphi\in F} c_{\mathrm{cap}}\cdot\varphi(\underline{c}\cdot x_{r,\varphi}+y_{r,\varphi}) \geqslant ld(e) \qquad \forall\, e\in E \tag{6.22}$$

$$y_{r,\varphi} \Leftrightarrow (\overline{c}\Leftrightarrow\underline{c})\cdot x_{r,\varphi} \leqslant 0 \qquad \forall\, r\in R,\quad \forall\,\varphi\in F \tag{6.23}$$

$$\sum_{\varphi\in F} x_{r,\varphi} \leqslant 1 \qquad \forall\, r\in R \tag{6.24}$$

$$x\in\{0,1\}^{|R|\times|F|},\ \ y\in\mathbb{Z}_+^{|R|\times|F|} \tag{6.25}$$

The number of variables grows by a factor of $|F|$ and the number of constraints increases by $|R|\cdot|F|$ compared to the model (COSTNLP). In comparison with the linearization (COSTBLP) the size of the model significantly reduces whereas the quality of the initial linear programming relaxation keeps more or less unchanged. The precise figures can be found in the tables of section 6.6. In the following sections we extend the preprocessing and bound scheme of CLAESSENS et. al. [23]. First of all in section 6.5.1 we apply techniques introduced for the (COSTBLP) model to the (COSTILP) formulation. Finally, we give some advanced variable elimination schemes and some valid inequalities for (COSTILP) in section 6.5.2. Both the variable elimination scheme and the cuts will be introduced for the (COSTILP) model, only. But all the ideas we will mention below are rather based on the underlying problem than on the particular model and can be simply transfered to the (COSTBLP) model.

6.5.1 Preprocessing and lower bounding derived from (COSTBLP)

Besides the network reduction leading to composition of edges and the adjustment of the right hand side $ld(e)$ to $\lceil ld(e)/c_{\mathrm{cap}}\rceil\cdot c_{\mathrm{cap}}$ which can be directly applied to (COSTILP) the variable elimination schemes and dominance rules provide an improved bounding for the variables x and y. Let $Z\subset R\times F\times\{\underline{c},\ldots,\overline{c}\}$ be the set of indices corresponding to variables in the preprocessed model (COSTBLP). Some variables $z_{r,\varphi,\gamma}$ are deleted due to variable elimination schemes and domination rules introduced in section 6.4.1. Let $\gamma^{\max}_{r,\varphi} := \max_{(r,\varphi,\gamma)\in Z}\gamma$ be the largest possible number of coaches for line (r,φ) in any optimal solution. If $\gamma^{\max}_{r,\varphi} = \Leftrightarrow\infty$, i.e. $\{(r,\varphi,\gamma)\mid\gamma\in\{\underline{c},\ldots,\overline{c}\}\}\cap Z=\emptyset$, we can delete $x_{r,\varphi}$ and $y_{r,\varphi}$ from (COSTILP), too. Otherwise, we know that $y_{r,\varphi}$ is bounded from above by $\gamma^{\max}_{r,\varphi}\Leftrightarrow\underline{c}$ in any optimal solution. Hence we can tighten inequality (6.23) by $y_{r,\varphi}\Leftrightarrow(\gamma^{\max}_{r,\varphi}\Leftrightarrow\underline{c})\cdot x_{r,\varphi}\leqslant 0$.

6.5.2 New preprocessing and lower bounding techniques

The objective of the initial linear programming relaxation of the preprocessed (COSTBLP) is slightly improved compared to the objective of the initial linear programming relaxation of (COSTILP) after applying the bounding of y variables. Nevertheless, the substantially smaller size of (COSTILP) permits a faster solution of the linear program compared to (COSTBLP). We can tighten the linear programming relaxation of both models by an advanced variable elimination scheme and various problem specific valid inequalities.

Elimination of variables

In the (COSTBLP) formulation a variable $z_{r,\varphi,\gamma}$ provides the complete information of a possible line and we can easily indicate the influence on the demand of coaches and the frequency requirement which lead to some elimination schemes (cf. section 6.4.1). In the (COSTILP) model the frequency and the number of coaches of a particular line is kept in different variables, but with the bounds $\underline{c}$ and $\overline{c}$ we also can estimate the influence on the problem data and derive some elimination schemes.

Suppose we have a line (r^*,φ^*) in the line plan that already satisfies the frequency requirement of all edges $e \in r^*$, i.e. $\varphi^* \geqslant \max_{e\in r^*} \underline{lfr}(e)$. Furthermore, assume that the frequency φ^* is sufficient to satisfy the demand of coaches of all edges $e \in r^*$, i.e. $\overline{c} \geqslant \max\{\lceil \max_{e\in r^*} \widetilde{ld}(e)/\varphi^*\rceil, \underline{c}\} =: \xi$. Let $\mathfrak{c}^{r^*,\varphi^*,\xi}$ with

$$\mathfrak{c}^{r^*,\varphi^*,\xi} = \underbrace{\lceil \varphi^*\Gamma_{r^*}\rceil(\mathfrak{c}^{\mathrm{tfix}} + \underline{c}\cdot\mathfrak{c}^{\mathrm{cfix}}) + d_{r^*}\varphi^*(\mathfrak{c}^{\mathrm{tvar}} + \underline{c}\cdot\mathfrak{c}^{\mathrm{cvar}})}_{\mathfrak{c}_x^{r^*,\varphi^*}} + \underbrace{(\lceil \varphi^*\Gamma_{r^*}\rceil\mathfrak{c}^{\mathrm{cfix}} + d_{r^*}\varphi^*\mathfrak{c}^{\mathrm{cvar}})}_{\mathfrak{c}_y^{r^*,\varphi^*}} \cdot (\xi \Leftrightarrow \underline{c})$$

be the cost of a line (r^*,φ^*) with ξ coaches. If $\mathfrak{c}^{r^*,\varphi^*,\xi} \leqslant \mathfrak{c}^{r^*,\varphi',\underline{c}} = \mathfrak{c}_x^{r^*,\varphi'}$ with $\varphi' > \varphi^*$ we can replace the line (r^*,φ') by (r^*,φ^*) in any optimal solution without increasing the objective. Hence the variables $x_{r^*,\varphi'}$ and $y_{r^*,\varphi'}$ are superfluous and can be removed from (COSTILP). Even if $\mathfrak{c}^{r^*,\varphi^*,\xi} > \mathfrak{c}^{r^*,\varphi',\underline{c}}$ we can derive an improved bound for the number of coaches in a line (r^*,φ'). If the cost $\mathfrak{c}^{r^*,\varphi',\xi'}$ exceeds $\mathfrak{c}^{r^*,\varphi^*,\xi}$, i.e.

$$\xi' \geqslant \left\lceil \frac{\mathfrak{c}^{r^*,\varphi^*,\xi} \Leftrightarrow \mathfrak{c}_x^{r^*,\varphi'}}{\mathfrak{c}_y^{r^*,\varphi'}} \right\rceil$$

we can replace (r^*,φ') with ξ' coaches by (r^*,φ') with ξ coaches in any optimal solution. Hence we can bound $y_{r^*,\varphi'}$ from above by $\lfloor(\mathfrak{c}^{r^*,\varphi^*,\xi} \Leftrightarrow \mathfrak{c}_x^{r^*,\varphi'})/\mathfrak{c}_y^{r^*,\varphi'}\rfloor$.

Due to the particular structure of $F = \{2^i | i = 0,1,2\}$ for all instances we can give another elimination scheme of lines with large frequency. Suppose the line (r^*,φ^*) satisfies the frequency requirement as well as the demand of coaches for all edges $e \in r^*$. We can replace the line (r^*,φ') equipped with ξ' coaches and $\varphi' > \varphi^*$ in any optimal solution by (r^*,φ^*) with $\min\{\overline{c}, \varphi'/\varphi^*\xi'\}$ coaches if $\lceil\varphi'\Gamma_r\rceil = \varphi'/\varphi^*\lceil\varphi^*\Gamma_r\rceil$, e.g. with the appropriate assumptions, $(r^*,4)$ with ξ' coaches can be replaced by $(r^*,2)$ with $\min\{\overline{c}, 2\xi'\}$ coaches. Hence we can eliminated $x_{r^*,\varphi'}$ and $y_{r^*,\varphi'}$ from (COSTILP).

The fixing of variables described in the remaining part of this section relies on the particular structure of the network, the cost structure, and the set of possible routes. In contrast to the elimination schemes based on three simple conditions described above, we *propose* several promising candidates for fixing. In some cases the validity of a fixing can be *proved* by exploring the consequences of the proposal.

The proposal of line fixings rely on *dead ends* of the supply networks. Suppose the supply network $G = (V, E)$ provides a dead end $u \overset{e_1}{\Leftrightarrow} v \overset{e_2}{\Leftrightarrow} w$ with $\delta(w) = e_2$, $\delta(v) = \{e_1, e_2\}$ (cf. figure 6.5), $\widetilde{ld}(e_2) \geqslant \widetilde{ld}(e_1)$, $\underline{lfr}(e_2) \geqslant \underline{lfr}(e_1)$, and $e_2 \notin R$. With this configuration, every line plan that satisfies the frequency requirement as well as the demand of coaches for e_2 already satisfies the edge e_1. Hence we can eliminate all line on route e_1 from (COSTILP). Furthermore, the frequency and the capacity of a line (r, φ) terminating at v is superfluous for edge e_1 hence we would save some cost if we replaced line (r, φ) by $(r \setminus e_1, \varphi)$ if $r \setminus e_1 \in R$. In a feasible solution of (COSTILP) there may be already a line with route $r \setminus e_1$ and frequency φ'. Frequency φ' may be *incompatible* with frequency φ, i.e. $\varphi' + \varphi \notin F$ or the sum of coaches in both lines may exceed $\overline{c}$. Hence an adjustment will result in an infeasible line plan such as in the following example. Let $G = (\{t, u, v, w\}, \{tu, uv, vw\})$, $R = \{t \Leftrightarrow u \Leftrightarrow v, u \Leftrightarrow v \Leftrightarrow w, v \Leftrightarrow w\}$, $\widetilde{ld}(tu) = 6$, $\widetilde{ld}(uv) = 5$, $\widetilde{ld}(vw) = 25$, $\overline{c} = 5$, $\underline{c} = 3$, and $F = \{1, 2, 4\}$. Every line plan, e.g. $(t \Leftrightarrow u \Leftrightarrow v, 2)$ with 3 coaches, $(u \Leftrightarrow v \Leftrightarrow w, 1)$ with 5 coaches, and $(v \Leftrightarrow w, 4)$ with 5 coaches, contains a line on route $u \Leftrightarrow v \Leftrightarrow w$, hence we cannot eliminate the corresponding variables from (COSTILP). Nevertheless, we apply this elimination proposal to the real-world InterCity network `sp97ic` depicted in figure 6.4. The edges LwHr and HrZl represent an appropriate dead end. In the following we give a proof for the fixing of the lines with route $r_{\text{Hr}-\text{Vl}} = \text{Hr} \Leftrightarrow \text{Zl} \Leftrightarrow \text{Dv} \Leftrightarrow \text{Ah} \Leftrightarrow \text{Nm} \Leftrightarrow \text{Ht} \Leftrightarrow \text{Ehv} \Leftrightarrow \text{Vl}$. The proof requires the reference numbers of the particular instance: We have $F = \{1\}$ (hence we can omit the frequency index) and the minimum and maximum number of coaches in a train is equal to 3 respectively 15. The values of $\underline{lfr}$ and $\widetilde{ld}$ for relevant edges can be found in figure 6.6. The cost of the relevant lines are represented in table 6.3.

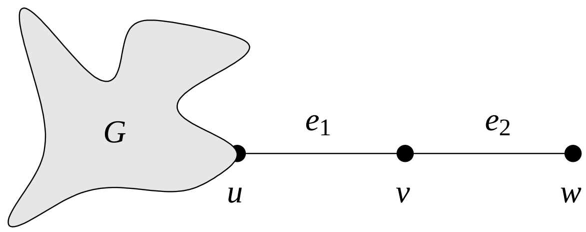


Figure 6.5: A dead end

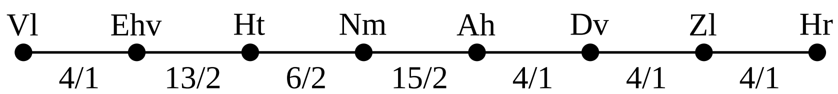


Figure 6.6: Route $r_{\text{Hr}-\text{Vl}}$ with demand of coaches and frequency requirement.

	Hr-Vl	Vl-Zl	Ah-Ehv
c_x^r	27106581	21594240	9662487
c_y^r	4571847	3750272	1758589

Table 6.3: Cost of some lines in the InterCity network `sp97ic`

CLAIM 6.2
An optimal solution of the InterCity instance `sp97ic` *of (COSTILP) does not contain lines on route* $r_{\mathrm{Hr-Vl}}$.

PROOF If a feasible solution contains the line on $r_{\mathrm{Hr-Vl}}$ then we can reduce the cost of the solution by appropriate replacements. Therefore, assume that (x,y) is a feasible solution of (COSTILP) with $x_{\mathrm{Hr-Vl}} = 1$ and $y_{\mathrm{Hr-Vl}} = \xi$.

1. If $x_{\mathrm{Vl-Zl}} = 0$, we obtain a feasible solution with cost savings of $5512341 + 821575\xi$ by the replacement $x_{\mathrm{Hr-Vl}} = y_{\mathrm{Hr-Vl}} = 0$, $x_{\mathrm{Vl-Zl}} = 1$, and $y_{\mathrm{Vl-Zl}} = \xi$.

2. If $x_{\mathrm{Vl-Zl}} = 1$, $y_{\mathrm{Vl-Zl}} \geqslant 1$, and $x_{\mathrm{Ah-Ehv}} = 0$, then the frequency requirement as well as the demand of coaches for the edges EhvVl, AhDv, and DvZl is already satisfied by the line on $r_{\mathrm{Vl-Zl}}$. The additional requirements of the remaining edges of $r_{\mathrm{Vl-Zl}}$ will be satisfied by the new line on $r_{\mathrm{Ah-Ehv}}$ with ξ coaches. We obtain a feasible solution with cost savings of $17444094 + 2813258\xi$ by the replacement $x_{\mathrm{Hr-Vl}} = y_{\mathrm{Hr-Vl}} = 0$, $x_{\mathrm{Ah-Ehv}} = 1$, and $y_{\mathrm{Ah-Ehv}} = \xi$.

3. Let $x_{\mathrm{Vl-Zl}} = 1$, $y_{\mathrm{Vl-Zl}} \geqslant 1$, and $x_{\mathrm{Ah-Ehv}} = 1$. The replacement of $x_{\mathrm{Hr-Vl}} = y_{\mathrm{Hr-Vl}} = 0$, and $y_{\mathrm{Ah-Ehv}} = \min\{12 = \overline{c} \Leftrightarrow \underline{c}, y_{\mathrm{Ah-Ehv}} + \xi\}$ results in a feasible solution with cost savings of at least $2710658 \Leftrightarrow 1758589\xi \geqslant 6003513$.

4. Let $x_{\mathrm{Vl-Zl}} = 1$, $y_{\mathrm{Vl-Zl}} = 0$, and $x_{\mathrm{Ah-Ehv}} = 0$. We set $x_{\mathrm{Ah-Ehv}} = 1$, $y_{\mathrm{Vl-Zl}} = 1$, and $y_{\mathrm{Ah-Ehv}} = \max\{0, \xi \Leftrightarrow 1\}$ and eliminate the line on $r_{\mathrm{Hr-Vl}}$ from the solution by $x_{\mathrm{Hr-Vl}} = y_{\mathrm{Hr-Vl}} = 0$ which yields a saving of at least 13693822.

5. Finally, let $x_{\mathrm{Vl-Zl}} = 1$, $y_{\mathrm{Vl-Zl}} = 0$, and $x_{\mathrm{Ah-Ehv}} = 1$. With the replacement of $x_{\mathrm{Hr-Vl}} = y_{\mathrm{Hr-Vl}} = 0$, $y_{\mathrm{Vl-Zl}} = 1$, and $y_{\mathrm{Ah-Ehv}} = \min\{12, \max\{0, \xi \Leftrightarrow 1\}\}$ we save at least an amount of 23356309.

In all cases we can find an appropriate replacement of the line on route $r_{\mathrm{Hr-Vl}}$ that keeps feasibility and yields a substantial cost reduction. □

Valid inequalities

In this section we collect some valid inequalities for (COSTILP). There is no general method of finding problem specific cuts. Often it is useful to have a close look at the linear description of the feasible region of an integer linear program. For very small instances we can give the feasible region by enumerating all integer solutions. There are finite algorithms (cf. [5, 19, 20]) that compute the linear description of the convex hull of all integer solutions. The linear description of the convex hull for a particular instance may give a clue for a class of valid inequalities for the complete model.

The first class of valid inequalities is similar to a class of cuts already used in the direct traveler approach (cf. proposition 5.3) and excludes some solutions with fractional frequencies. For reasons of convenience and the slight modification we recall the representation and the proof of these cuts.

COROLLARY 6.3
Let $E' \subset E$, $\alpha^r_{E'} := |r \cap E'|$, $\alpha^{\max}_{E'} = \max\{\alpha^r_{E'} \mid r \in R\}$, and $\alpha^{\min}_{E'} = \min\{\alpha^r_{E'} \mid r \in R, \alpha^r_{E'} \geqslant 2\}$. The inequalities

$$\sum_{r \in R, \alpha^r_{E'} \geqslant 1} \sum_{\varphi \in F} \varphi x_{r,\varphi} \geqslant \left\lceil \frac{\underline{lfr}(E')}{\alpha^{\max}_{E'}} \right\rceil \tag{6.26}$$

$$\sum_{r \in R, \alpha^r_{E'} \geqslant 2} \sum_{\varphi \in F} \varphi x_{r,\varphi} \leqslant \left\lfloor \frac{\overline{lfr}(E')}{\alpha^{\min}_{E'}} \right\rfloor \tag{6.27}$$

are valid for (COSTILP).

PROOF From inequalities (6.20) we easily derive

$$\sum_{r \in R, \alpha^r_{E'} \geqslant 1} \sum_{\varphi \in F} \alpha^r_{E'} \varphi x_{r,\varphi} \geqslant \underline{lfr}(E')$$

and by replacing $\alpha^r_{E'}$ by $\alpha^{\max}_{E'}$ we obtain

$$\sum_{r \in R, \alpha^r_{E'} \geqslant 1} \sum_{\varphi \in F} \varphi x_{r,\varphi} \geqslant \frac{\underline{lfr}(E')}{\alpha^{\max}_{E'}}.$$

The left hand side is always integer and hence we can round up the right hand side to the next integer greater or equal than $\underline{lfr}(E')/\alpha^{\max}_{E'}$ which leads to (6.26). We leave out the proof of inequality (6.27) which is quite similar to the proof of (6.26). □

Obviously, we can apply the proof of corollary 6.3 to the constraint (6.22) which reads after adjusting the right hand side as follows.

$$\sum_{r \in R, r \ni e} \sum_{\varphi \in F} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) \geqslant \widetilde{ld}(e) \tag{6.28}$$

We obtain a cut similar to (6.26) that excludes some solutions with a fractional number of coaches.

COROLLARY 6.4
With the notation of corollary 6.3 the inequality

$$\sum_{r\in R\,,\alpha^r_{E'}\geqslant 1}\sum_{\varphi\in F}\varphi(\underline{c}\cdot x_{r,\varphi}+y_{r,\varphi})\geqslant\left\lceil\frac{\widetilde{ld}(E')}{\alpha^{\max}_{E'}}\right\rceil \tag{6.29}$$

is valid for (COSTILP).

Another class of valid inequalities is derived from the following example. Let $v\in V$ with $\delta(v)=\{e_1,e_2\}$ and $\widetilde{ld}(e_1)>\widetilde{ld}(e_2)$. Either the line plan contains a line via e_1 stopping at v or the number of coaches running via e_2 is at least $\widetilde{ld}(e_1)$. This idea will be generalized in the next proposition.

PROPOSITION 6.5
Let $E'\subset E$, $e_0\in E\setminus E'$, $\alpha^r_{E'}:=|r\cap E'|$, and $\widetilde{ld}(e_0)>\sum_{e\in E'}\widetilde{ld}(e)$. The inequality

$$\left(\widetilde{ld}(e_0)\Leftrightarrow\sum_{e\in E'}\widetilde{ld}(e)\right)\sum_{\substack{r\in R\\ r\cap E'=\emptyset,r\ni e_0}}\sum_{\varphi\in F}x_{r,\varphi}+\sum_{r\in R}\sum_{\varphi\in F}\alpha^r_{E'}\varphi(\underline{c}\cdot x_{r,\varphi}+y_{r,\varphi})\geqslant\widetilde{ld}(e_0) \tag{6.30}$$

is valid for (COSTILP).

PROOF First of all, let us assume that $\sum_{r\in R\,,r\cap E'=\emptyset,r\ni e_0}\sum_{\varphi\in F}x_{r,\varphi}\geqslant 1$. Then (6.30) is obviously dominated by (6.28) and hence is valid for (COSTILP). Now assume $\sum_{r\in R\,,r\cap E'=\emptyset}\sum_{\varphi\in F}x_{r,\varphi}=0$, i.e. all routes of lines in the line plan that contain e_0 contain at least another edge $e\in E'$. Therefore, we have

$$\widetilde{ld}(e_0)\leqslant\sum_{r\in R\,,r\ni e_0}\sum_{\varphi\in F}\varphi(\underline{c}\cdot x_{r,\varphi}+y_{r,\varphi})\leqslant\sum_{r\in R}\sum_{\varphi\in F}\alpha^r_{E'}\varphi(\underline{c}\cdot x_{r,\varphi}+y_{r,\varphi})$$

which gives the validity of (6.30). □

Now, consider the configuration $v\in V$, $\delta(v)=\{e_1,e_2\}$ and $\widetilde{ld}(e_2)+\underline{c}\geqslant\widetilde{ld}(e_1)\geqslant\widetilde{ld}(e_2)$. It is easy to see that one of the following three alternatives holds.

- The line plan contains a line via e_2 stopping at v.
- Every line in the line plan contains e_1 and e_2, and hence the number of coaches running via e_2 is at least $\widetilde{ld}(e_1)$ (Proposition 6.5).
- Every line in the line plan that contains e_2 also contains e_1 and there is a line via e_1 stopping at v. Hence the number of coaches via e_1 is at least $\widetilde{ld}(e_2)$ to satisfy the demand of e_2 plus at least $\underline{c}$ coaches of the line via e_1 stopping at v.

The following proposition generalizes this observation.

PROPOSITION 6.6
Let $E' \subset E$, $e_0 \in E \setminus E'$, $\alpha^r_{E'} := |r \cap E'|$, and $\sum_{e \in E'} \widetilde{ld}(e) + \underline{c} > \widetilde{ld}(e_0) > \sum_{e \in E'} \widetilde{ld}(e)$. Furthermore, let $\mu = \widetilde{ld}(e_0) \Leftrightarrow \widetilde{ld}(E')$ and $\nu = \widetilde{ld}(E') + \underline{c} \Leftrightarrow \widetilde{ld}(e_0)$. The inequality

$$\mu\nu \sum_{\substack{r \in R, r \not\ni e_0 \\ r \cap E' \neq \emptyset}} \sum_{\varphi \in F} x_{r,\varphi} + \mu \sum_{r \in R, r \ni e_0} \sum_{\varphi \in F} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) + \nu \sum_{r \in R} \sum_{\varphi \in F} \alpha^r_{E'} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi})$$
$$\geqslant \mu\nu + \mu\widetilde{ld}(e_0) + \nu\widetilde{ld}(E') \quad (6.31)$$

is valid for (COSTILP).

PROOF First of all let us assume $\sum_{r \in R, r \not\ni e_0, r \cap E' \neq \emptyset} \sum_{\varphi \in F} x_{r,\varphi} =: \xi \geqslant 1$. With $\mu \geqslant 1$ and $\nu \geqslant 1$ inequality (6.31) is dominated by (6.28) because $\sum_{r \in R, r \ni e_0} \sum_{\varphi \in F} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) \geqslant \widetilde{ld}(e_0)$ and $\sum_{r \in R} \sum_{\varphi \in F} \alpha^r_{E'} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) \geqslant \widetilde{ld}(E')$. Now consider the case $\xi = 0$ and $\sum_{r \in R, r \ni e_0, r \cap E' = \emptyset} \sum_{\varphi \in F} x_{r,\varphi} =: \rho = 0$, i.e. all routes on lines in the line plan that contain e_0 also contain another edge $e \in E'$. Conversely, a route $r \in R$ included in the line plan with $r \cap E' \neq \emptyset$ contains e_0, too. The capacity of lines containing e_0 must satisfy the demand of $\widetilde{ld}(e_0)$ coaches and hence the number of coaches running along edges of E' is at least $\widetilde{ld}(e_0)$, too. With the definition of μ and ν we obtain

$$\mu \sum_{r \in R, r \ni e_0} \sum_{\varphi \in F} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) + \nu \sum_{r \in R} \sum_{\varphi \in F} \alpha^r_{E'} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) \geqslant \mu\widetilde{ld}(e_0) + \nu\widetilde{ld}(e_0)$$
$$= \mu\widetilde{ld}(e_0) + \nu(\mu + \widetilde{ld}(E'))$$
$$= \mu\nu + \mu\widetilde{ld}(e_0) + \nu\widetilde{ld}(E').$$

It remains to prove the validity of (6.31) for the case $\xi = 0$ and $\rho \geqslant 1$, i.e. every route in the line plan with $r \cap E' \neq \emptyset$ contains e_0, too, but there is at least one line (r^*, φ^*) in the line plan with $e_0 \in r^*$ and $r^* \cap E' = \emptyset$. In order to satisfy the demand of edges in E' the capacity of lines containing e_0 and an edge of E' must be at least $\widetilde{ld}(E')$. The line (r^*, φ^*) consists of at least $\underline{c}$ coaches, hence we have

$$\sum_{r \in R, r \ni e_0} \sum_{\varphi \in F} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) = \sum_{\substack{r \in R, r \ni e_0 \\ r \cap E' \neq \emptyset}} \sum_{\varphi \in F} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) + \sum_{\substack{r \in R, r \ni e_0 \\ r \cap E' = \emptyset}} \sum_{\varphi \in F} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi})$$
$$\geqslant \widetilde{ld}(E') + \underline{c}$$

Again, with the definition of μ and ν we obtain

$$\mu \sum_{r \in R, r \ni e_0} \sum_{\varphi \in F} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) + \nu \sum_{r \in R} \sum_{\varphi \in F} \alpha^r_{E'} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) \geqslant \mu(\widetilde{ld}(E') + \underline{c}) + \nu\widetilde{ld}(E')$$
$$= \mu(\widetilde{ld}(e_0) + \nu) + \nu\widetilde{ld}(E')$$
$$= \mu\nu + \mu\widetilde{ld}(e_0) + \nu\widetilde{ld}(E').$$

which proves the statement of the proposition. □

The following valid inequality relies on the fact that either for each edge $e \in E$ with $\underline{c} \cdot \underline{lfr}(e) < \widetilde{ld}(e) < \underline{c}(\underline{lfr}(e)+1)$ the number of trains via e is at least $\underline{lfr}(e)+1$ or the number of additional coaches of lines via e is at least $\widetilde{ld}(e) \Leftrightarrow \underline{c} \cdot \underline{lfr}(e)$.

PROPOSITION 6.7
For each edge $e \in E$ with $\underline{c} \cdot \underline{lfr}(e) < \widetilde{ld}(e) < \underline{c}(\underline{lfr}(e)+1)$ the inequality

$$\sum_{r \in R, r \ni e} \sum_{\varphi \in F} \xi \varphi x_{r,\varphi} + \min\{\xi, \varphi\} y_{r,\varphi} \geqslant \xi(\underline{lfr}(e)+1) \tag{6.32}$$

is valid for (COSTILP) with $\xi = \widetilde{ld}(e) \Leftrightarrow \underline{c} \cdot \underline{lfr}(e)$.

PROOF Let (x, y) be a feasible solution of (COSTILP). If $\sum_{r \in R, r \ni e} \sum_{\varphi \in F} \varphi x_{r,\varphi} \geqslant \underline{lfr}(e)+1$ the inequality is obviously valid. Now, let us assume that $\sum_{r \in R, r \ni e} \sum_{\varphi \in F} \varphi x_{r,\varphi} = \underline{lfr}(e)$. The inequality (6.32) becomes

$$\sum_{r \in R, r \ni e} \sum_{\varphi \in F} \min\{\varphi, \xi\} y_{r,\varphi} \geqslant \xi$$

and is similar to the inequality (6.28) which reads as $\sum_{r \in R, r \ni e} \sum_{\varphi \in F} \varphi y_{r,\varphi} \geqslant \xi$ after subtracting $\underline{c} \cdot \underline{lfr}(e)$ from the left and right hand side. We can replace φ by $\min\{\varphi, \xi\}$ in (6.28) since y is integer and hence obtain the validity of (6.32) which completes the proof. □

Another class of valid inequalities is originated by the following idea. Due to the lower bound $\underline{c}$ for the number of coaches in a train the requirement on the frequency (6.20) has a particular impact on the total number of coaches in lines via an edge $e \in E$. Suppose the line plan contains a line (r, φ) with $e \in r$. Independent of the particular demand $\widetilde{ld}(e)$ there must be at least $\underline{c}(\underline{lfr}(e) \Leftrightarrow \varphi)$ coaches of other lines via e. This observation is generalized in the following proposition.

PROPOSITION 6.8
Let $e \in E$ and $R' \subset R$ with $e \in r$ for all $r \in R'$ and $\sum_{r \in R'} \sum_{\varphi \in F} x_{r,\varphi} \leqslant 1$ in any feasible solution of (COSTILP). Note that the set of variables $\{x_{r,\varphi} \mid r \in R', \varphi \in F\}$ corresponds to a clique in the graph of logical implications (cf. section 4.9.2). The inequality

$$\sum_{r \in R \setminus R', r \ni e} \sum_{\varphi \in F} \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) \geqslant \widetilde{ld}(e) \left(1 \Leftrightarrow \sum_{r \in R'} \sum_{\varphi \in F} x_{r,\varphi}\right) + \underline{c} \sum_{r \in R'} \sum_{\varphi \in F} (\underline{lfr}(e) \Leftrightarrow \varphi) x_{r,\varphi} \tag{6.33}$$

is valid for (COSTILP).

PROOF Let (x, y) be a feasible solution of (COSTILP). If $\sum_{r \in R'} \sum_{\varphi \in F} x_{r,\varphi} = 0$ inequality (6.33) is obviously valid since (x, y) satisfies (6.28). Otherwise, there is one $x_{r*,\varphi^*} = 1$ with $r \in R'$. The frequency requirement (6.20) becomes $\sum_{r \in R \setminus R', r \ni e'} \sum_{\varphi \in F} \varphi x_{r,\varphi} \geqslant \underline{lfr}(e) \Leftrightarrow \varphi^*$ and therefore we have at least $\underline{lfr}(e) \Leftrightarrow \varphi^*$ trains of lines $R \setminus R' \times F$ in the line plan each with at least $\underline{c}$ coaches. Hence (x,y) fulfills (6.33) and this completes the proof. □

The final class of valid inequalities is effective for edges $e \in E$ with a large demand of coaches $\widetilde{ld}(e)$ but a relatively small lower bound for the frequency requirement. Hence if the sum of frequency in a line plan is close to $\underline{lfr}(e)$ the number of coaches in the operating lines must be close to $\overline{c}$.

Proposition 6.9
Let $e \in E$ and (r^*, φ^*) be a line with $e \in r^*$. If $\widetilde{ld}(e) \Leftrightarrow \overline{c}(\underline{lfr}(e) \Leftrightarrow \max\{1, \underline{lfr}(e) \Leftrightarrow \overline{lfr}(e) + \varphi^*\}) \geqslant 0$ then the inequality

$$\varphi^*(\underline{c}x_{r^*,\varphi^*} + y_{r^*,\varphi^*}) \Leftrightarrow \underbrace{\left(\underline{lfr}(e) \Leftrightarrow \sum_{\substack{(r,\varphi)\in R\times F\setminus(r^*,\varphi^*)\\ r\ni e}} \varphi x_{r,\varphi}\right)}_{=:\xi} \theta \geqslant 0 \qquad (6.34)$$

is valid for (COSTILP) with $\theta = \lceil(\widetilde{ld}(e) \Leftrightarrow \overline{c}(\underline{lfr}(e) \Leftrightarrow \max\{1, \underline{lfr}(e) \Leftrightarrow \overline{lfr}(e) + \varphi^*\}))/\varphi^*\rceil$.

Proof First of all, note that $\xi \in \{\overline{lfr}(e) \Leftrightarrow \underline{lfr}(e), \ldots, \varphi^*\}$. The inequality (6.34) is obviously valid for $\xi \leqslant 0$. Now suppose $\xi \geqslant 1$, i.e. $x_{r^*,\varphi^*} = 1$. If we prove validity of

$$\xi(\underline{c}x_{r^*,\varphi^*} + y_{r^*,\varphi^*}) \Leftrightarrow \xi\theta \geqslant 0 \qquad (6.35)$$

then the validity of (6.34) follows since (6.35) dominates (6.34). We can derive validity of (6.35) from (6.28).

$$\sum_{\substack{(r,\varphi)\in R\times F\\ r\ni e}} \varphi(\underline{c}\cdot x_{r,\varphi} + y_{r,\varphi}) \geqslant \widetilde{ld}(e)$$

$$\varphi^*(\underline{c}x_{r^*,\varphi^*} + y_{r^*,\varphi^*}) \geqslant \widetilde{ld}(e) \Leftrightarrow \overbrace{\sum_{\substack{(r,\varphi)\in R\times F\setminus(r^*,\varphi^*)\\ r\ni e}} \varphi\underbrace{(\underline{c}\cdot x_{r,\varphi} + y_{r,\varphi})}_{\leqslant \overline{c}}}^{\leqslant \overline{c}(\underline{lfr}(e)-\xi)}$$

$$\varphi^*(\underline{c}x_{r^*,\varphi^*} + y_{r^*,\varphi^*}) \geqslant \widetilde{ld}(e) \Leftrightarrow \overline{c}(\underline{lfr}(e) \Leftrightarrow \xi).$$

Due to $x_{r^*,\varphi^*} = 1$ and (6.21) we have $\xi \geqslant \underline{lfr}(e) \Leftrightarrow \overline{lfr}(e) + \varphi^*$ and with the assumption $\xi \geqslant 1$ we obtain

$$\varphi^*(\underline{c}x_{r^*,\varphi^*} + y_{r^*,\varphi^*}) \geqslant \widetilde{ld}(e) \Leftrightarrow \overline{c}(\underline{lfr}(e) \Leftrightarrow \max\{1, \underline{lfr}(e) \Leftrightarrow \overline{lfr}(e) + \varphi^*\})$$
$$\underline{c}x_{r^*,\varphi^*} + y_{r^*,\varphi^*} \geqslant (\widetilde{ld}(e) \Leftrightarrow \overline{c}(\underline{lfr}(e) \Leftrightarrow \max\{1, \underline{lfr}(e) \Leftrightarrow \overline{lfr}(e) + \varphi^*\}))/\varphi^*.$$

The left hand side is integer and hence we can round up the right hand side to the next integer which becomes θ afterwards. This proves the validity of (6.35) and finally the validity of (6.34). □

	$\lvert V \rvert$	$\lvert E \rvert$	$\lvert R \rvert$	F	$\underline{c}$	$\overline{c}$
sp97ic	36	52	831	$\{1\}$	3	15
sp98ic	41	46	627	$\{1,2\}$	3	15
sp98ir	44	44	420	$\{1,2\}$	3	12
sp98ar	118	134	913	$\{1,2,4\}$	2	10

Table 6.4: Problem parameters

Inequality (6.34) can be interpreted as a new bound for the number of coaches of line (r^*,φ^*). With the idea of proposition 6.9 we can easily derive a similar bound for the number of *additional* coaches of line (r^*,φ^*).

COROLLARY 6.10
Let $e \in E$ and (r^*,φ^*) be a line with $e \in r^*$. If $\widetilde{ld}(e) \Leftrightarrow \overline{c}(\underline{lfr}(e) \Leftrightarrow \max\{1, \underline{lfr}(e) \Leftrightarrow \overline{lfr}(e) + \varphi^*\} \Leftrightarrow \varphi^* \underline{c}) \geqslant 0$ then the inequality

$$\varphi^* y_{r^*,\varphi^*} \Leftrightarrow \left(\underline{lfr}(e) \Leftrightarrow \sum_{\substack{(r,\varphi)\in R\times F\setminus(r^*,\varphi^*)\\ r\ni e}} \varphi x_{r,\varphi} \right) \theta \geqslant 0 \tag{6.36}$$

is valid for (COSTILP) with $\theta = \lceil (\widetilde{ld}(e) \Leftrightarrow \overline{c}(\underline{lfr}(e) \Leftrightarrow \max\{1, \underline{lfr}(e) \Leftrightarrow \overline{lfr}(e) + \varphi^*\}) \Leftrightarrow \varphi^* \underline{c})/\varphi^* \rceil$.

PROOF The proof of proposition 6.9 directly applies to corollary 6.10. □

As mentioned above, the advanced variable elimination schemes and the bunch of cutting planes are designed for the (COSTILP) model but these techniques can be easily transferred to the (COSTBLP) model. Obviously, the route/frequency elimination scheme is completely independent of the current formulation. Furthermore, the substitutions $x_{r,\varphi} = \sum_{\gamma=\underline{c}}^{\overline{c}} z_{r,\varphi,\gamma}$ and $y_{r,\varphi} = \sum_{\gamma=\underline{c}}^{\overline{c}} (\gamma \Leftrightarrow \underline{c}) z_{r,\varphi,\gamma}$ provide a transformation for all cuts developed in section 6.5.2 to the (COSTBLP) model.

6.6 Computational investigation

In this section we discuss the performance of the models (COSTBLP) and (COSTILP). Therefore, we apply the different formulations to four real-world instances of the cost optimal line planning problem provided by NS Reizigers. The set of instances consists of the already mentioned InterCity network `sp97ic` (cf. figure 6.4) and three other supply networks of the Dutch railroad network. We have another InterCity instance `sp98ic`, a (part of) the InterRegio network `sp98ir`, and an AggloRegio instance `sp98ar` (cf. figures 6.7-6.9). The `sp98ir` supply network is not connected. Optimal lines of the small parts in the north and south can be easily computed and we can concentrate on the main part of `sp98ir`. Table 6.4 summarizes the characteristics of the instances. The data arises from a strategic planning scenario at Nederlandse Spoorwegen (NS). The problem data, especially the cost data mentioned below, does not represent a particular

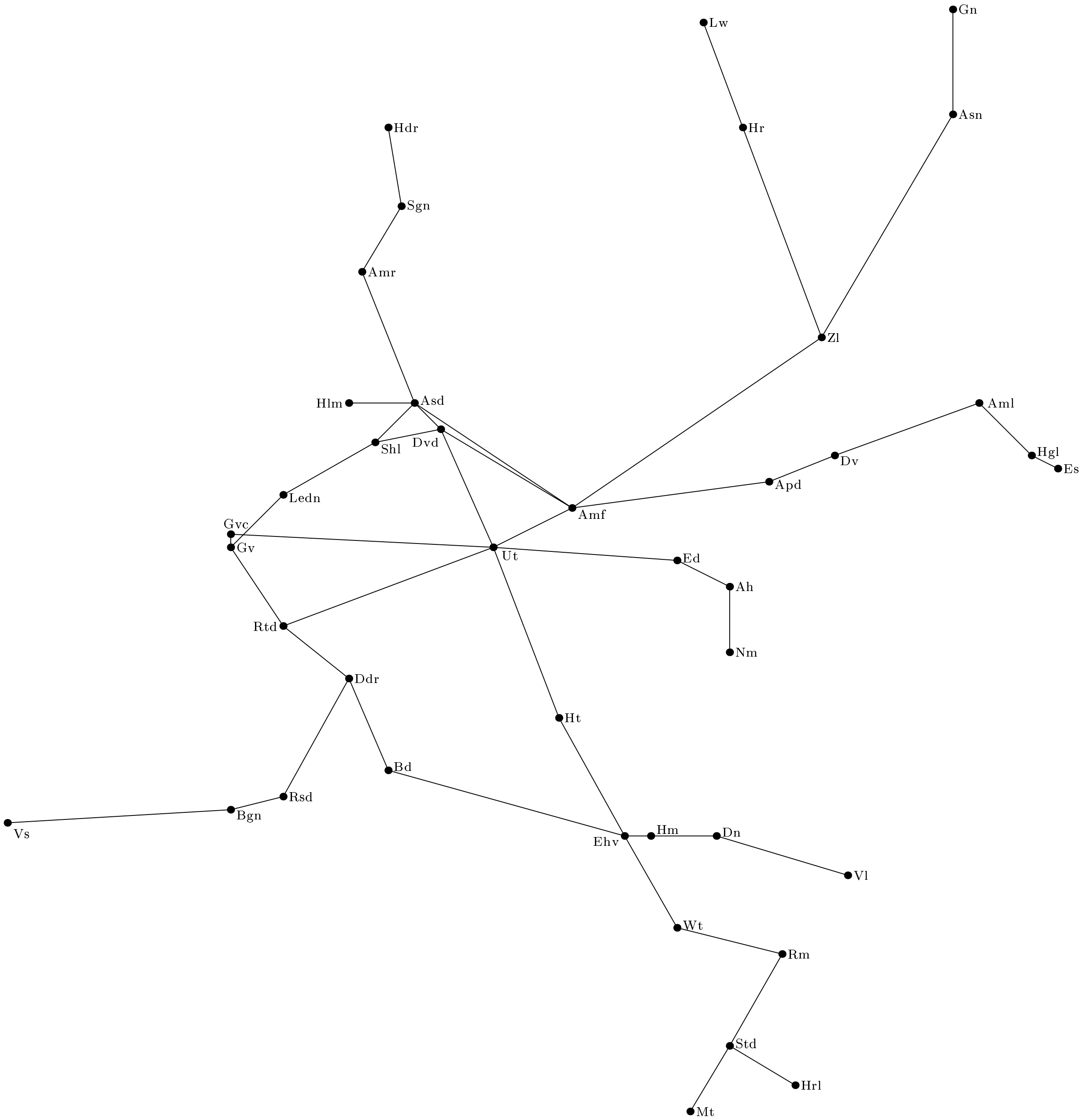


Figure 6.7: The Dutch InterCity network `sp98ic`

unit or monetary, but provides an appropriate mapping of the real-world situation. The data is stored in four GAMS files together with a model similar to (COSTBLP). This model represents some relaxation of the (COSTBLP) model, e.g. in the model the inequality (6.8) is abandoned although reasonable values of $\overline{lfr}$ are present. Furthermore, some preprocessing is done that does not strictly take care of the feasibility of some variable elimination and lower bounding. In this section we compare the model (COSTBLP) and the preprocessing suggested by CLAESSENS et. al. [23] to the (COSTILP) approach with respect to the data of the four real-world instances.

All computational experiments are performed on an HP C180 workstation running HPUX 10.20. The models (COSTBLP) and (COSTILP) are coded with GAMS and the resulting binary respectively integer linear programs are *"solved"* with the commercial mixed integer linear programming solver CPLEX version 5.0. CPLEX permits a variety of options that dramatically influence the solution process of the implemented branch-and-bound algorithm. The most relevant options correspond to the node and variable selection scheme (cf. sections 4.7.1 and 4.7.2). We obtain the best results according to the value of the general lower bound which is almost responsible for a reasonable performance guarantee, with the strong branching variable and the best-bound node selection scheme. If we focus on (good) feasible solutions, the depth-first-search node selection is superior in some cases. Furthermore, CPLEX provides an implementation of a general preprocessing and a constraint generation procedure, which tries to find violated clique and cover cuts. Especially for the (COSTBLP) model the so called *presolve* substantially improves the formulation. Finally, the dual simplex algorithm seems to be superior for solving the initial linear programming relaxation of the (COSTILP) integer linear programs whereas the primal simplex seems to be better for (COSTBLP).

Solution of the pure formulations

First of all we can try to solve the instances without any problem specific preprocessing and lower bounding. The results are summarized in tables 6.5 and 6.6. We rely on the general preprocessing and constraint generation implemented in the mixed integer linear programming solver. The CPLEX presolve eliminates a humble number of variables of the (COSTBLP) as well as of the (COSTILP) problem, but improves the lower bound of the first branch-and-bound node of (COSTBLP) compared to the pure linear programming relaxation by 2.65%. For the (COSTILP) problems the improvement is about 2.57% on the average. The lower bounds of (COSTBLP)

	sp97ic	sp98ic	sp98ir	sp98ar
# constraints	665	765	552	1315
# variables	9973	16302	8400	24651
# non-zeros	180913	310284	160560	442044
LP relaxation	3.7932	4.2473	2.0634	5.0591
LP relaxation in the first B&B node	3.8841	4.3648	2.1202	5.1957
CPU seconds	62.14	211.07	40.20	215.25

Table 6.5: Reference values of (COSTBLP) without problem specific preprocessing

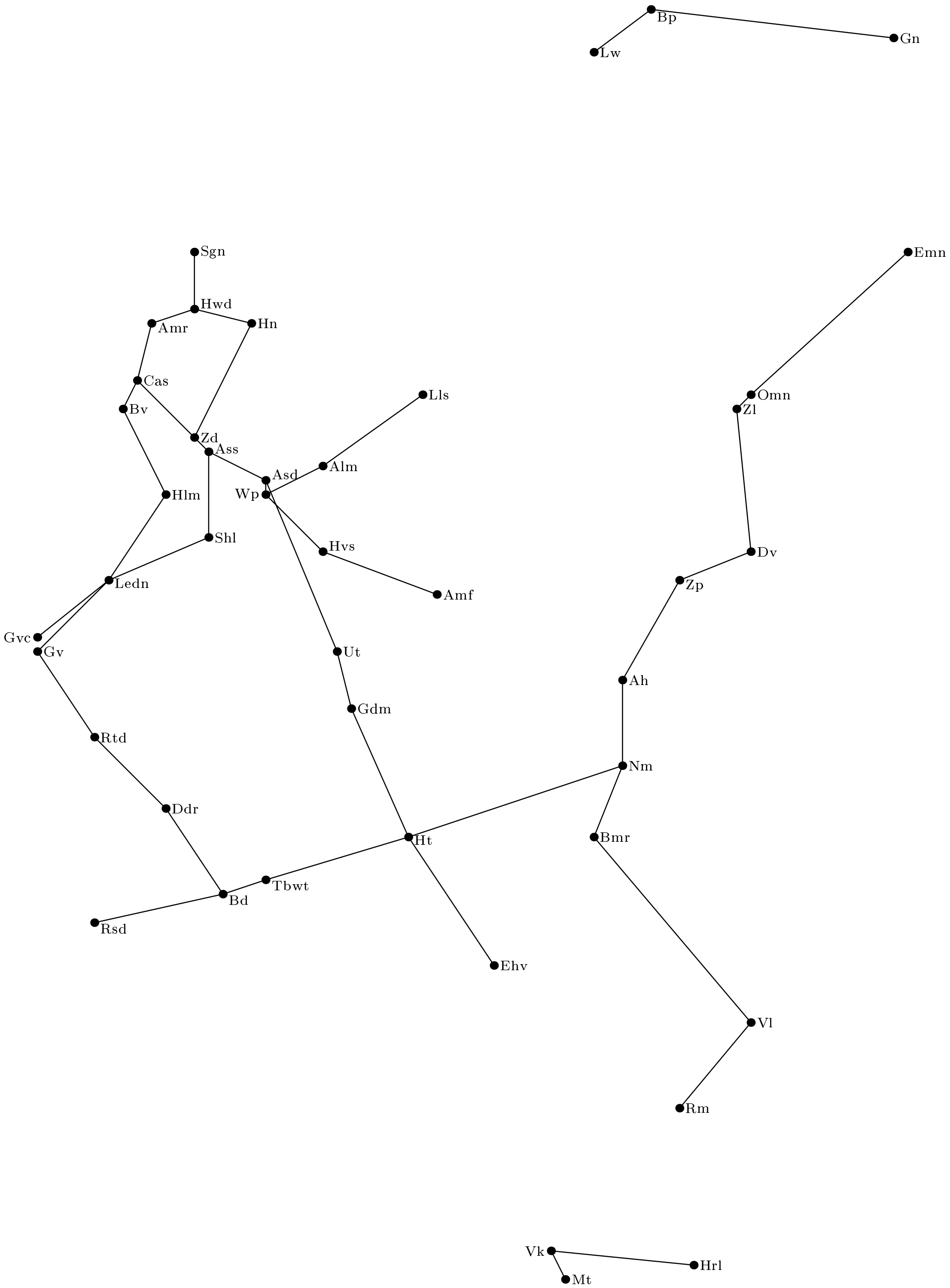


Figure 6.8: The Dutch InterRegio network `sp98ir`

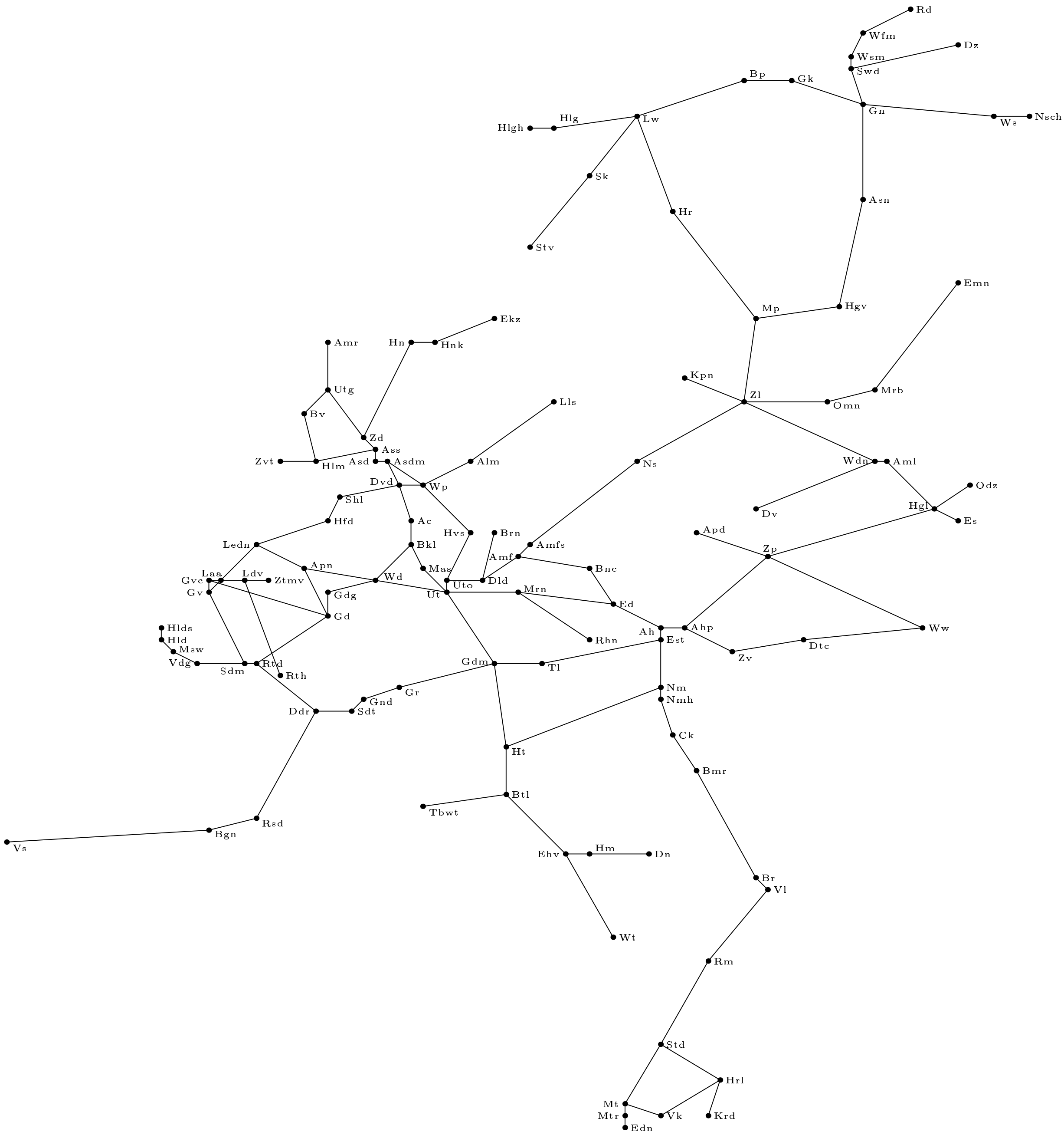


Figure 6.9: The Dutch AggloRegio network `sp98ar`

	sp97ic	sp98ic	sp98ir	sp98ar
# constraints	814	2019	1392	4054
# variables	1662	2508	1680	5478
# non-zeros	20565	33914	22808	70053
LP relaxation	3.7710	4.2422	2.0529	4.9214
LP relaxation in the first B&B node	3.8455	4.3342	2.0987	5.1145
CPU seconds	5.01	11.87	7.24	25.05

Table 6.6: Reference values of (COSTILP) without problem specific preprocessing

	sp97ic	sp98ic	sp98ir	sp98ar
# constraints	665	765	552	1315
# variables	6731	11008	3627	12583
# non-zeros	125139	218659	76107	257332
LP relaxation	3.8781	4.3392	2.1142	5.1892
LP relaxation in the first B&B node	3.8841	4.3664	2.1221	5.2107
CPU seconds	63.09	336.43	27.52	245.13

Table 6.7: Reference values of (COSTBLP) with problem specific preprocessing

compared to the bounds of (COSTILP) provide a gain of 1.08% but the solution times increase by a factor of 11 and the size of the formulation (number of non-zeros) increases by a factor of almost 8. The computation time (CPU seconds) include the time for model generation (GAMS), the problem specific and general preprocessing, and the solution of the initial linear program.

Solution with application of the preprocessing derived from (COSTBLP)

In the next step we apply the variable elimination scheme of section 6.4.1 and the tightening of constraint (6.9) respectively (6.22) to both models (cf. table 6.7 and 6.8). The elimination scheme reduces the number of variables by 42.69% and the number of non-zeros by 61.31% for the (COSTBLP) model. The number of constraints are about the same size because NS

	sp97ic	sp98ic	sp98ir	sp98ar
# constraints	814	2019	1392	4054
# variables	1662	2508	1680	5478
# non-zeros	19596	33912	22783	69791
LP relaxation	3.8477	4.3365	2.1049	5.0467
LP relaxation in the first B&B node	3.8477	4.3392	2.1142	5.1380
CPU seconds	5.40	13.39	7.42	29.88

Table 6.8: Reference values of (COSTILP) with preprocessing derived from (COSTBLP)

already processed the networks according to the edge shrinking procedure. The lower bound of the first branch-and-bound node increases due to the tightening of (6.9) by 0.10%, only. If we compare the pure linear programming relaxation values (without any general preprocessing) the improvement is about 2.36%. This indicates that the CPLEX presolve automatically identifies the possible tightening of constraint (6.9).

In section 6.5.1 we already mentioned that the variable elimination scheme for (COSTBLP) results in improved upper bounds for the y variables in the (COSTILP) formulation, hence the size of the problems does not significantly reduce, but compared with the preprocessed (COSTBLP) model still provides a reduction of factor 3 (variables), 5 (non-zeros), and 12 (CPU time). The improvement of the lower bounds is about 0.34% but compared to the lower bound of (COSTBLP) we still loose about 0.84%. The preprocessing derived from (COSTBLP) is almost covered by the general CPLEX presolve. Nevertheless, we should always modify the instances according to the preprocessing rules. On the one hand we save preprocessing time of the mixed integer linear programming solver and on the other hand we become more solver independent because there are several mixed integer linear programming solvers with flimsy presolve which do not automatically detect possibilities of elimination and tightening.

Solution with application of the new preprocessing and the valid inequalities

The computational investigations of both models indicate that (COSTBLP) provides slightly better bounds at the expense of a larger formulation and hence larger computations times. The cuts and the preprocessing described in section 6.5.2 should improve both models. In order to apply the large number of valid inequalities, we have developed a cut-and-branch algorithm on the top of CPLEX. Similar to a general cut-and-branch algorithm we iteratively add violated cuts and solve the corresponding linear program until the separation of cuts does not find another violated inequality. Afterwards, we add the cuts to the initial formulation and start the branch-and-bound procedure.

	`sp97ic`	`sp98ic`	`sp98ir`	`sp98ar`
# iterations	46	49	21	55
# cuts	253	304	131	663
# constraints	918	1069	683	1978
# variables	6731	11008	3627	12583
# non-zeros	366452	672166	203973	768964
LP relaxation	3.9167	4.4206	2.1656	5.2001
LP relaxation in the first B&B node	3.9272	4.4288	2.1716	5.2417
CPU seconds	202.57	369.24	91.69	600.41

Table 6.9: Reference values of (COSTBLP) including cuts

	sp97ic	sp98ic	sp98ir	sp98ar
# iterations	57	52	22	52
# cuts	265	301	150	596
# constraints	1079	2127	1425	4087
# variables	1356	2262	1557	4729
# non-zeros	54303	125449	69578	204666
LP relaxation	3.9188	4.4229	2.1666	5.2103
LP relaxation in the first B&B node	3.9197	4.4262	2.1668	5.2375
CPU seconds	64.21	82.31	29.78	256.63

Table 6.10: Reference values of (COSTILP) including cuts

Separation of cuts

Similar to the direct traveler approach we cannot give a general separation scheme for cuts based on some subsets of E ((6.26)-(6.31)). But with a fixed maximum cardinality of E‘ $\subset E$ the number of possibles cuts of class (6.26), (6.27), (6.29), (6.30), and (6.31) is polynomial in the size of the input. Furthermore, we have $|E|$ cuts of class (6.32) and $|E| \cdot |R|$ cuts of class (6.34) and (6.36). Hence we can separate appropriate violated inequalities (except those of class (6.33)) by checking a polynomial number of inequalities. Our investigation shows that for the class (6.30) and (6.31) inequalities corresponding to set E' with $|E'| = 2$ always dominate inequalities of the same type with larger E', hence we use cuts corresponding to $|E'| = 2$, only. Similar to the separation of inequalities of class (6.26) and (6.27) in the direct traveler approach, we only add violated constraints with $E' \subset \delta(v)$, $v \in V$ which is also polynomial for degree constraint graphs and absolutely sufficient for the real world instances with a maximal degree of 6 (Utrecht in sp98ic). For none of the instances we find violated cuts of (6.27) and (6.29) and hence we concentrate on cuts of type (6.26). Furthermore, we find out that for the particular instances the cuts of class (6.36) always dominate cuts of class (6.34). Therefore, we eliminate the cuts (6.34) from the cut-and-branch algorithm.

With the configuration $R^* = \{r_0\}$ we easily find candidates of inequalities of class (6.33) which can be separated as well by checking a polynomial number of route-edge combinations. The problem of finding larger sets R^* which leads to violated cuts of type (6.33) is related to the clique generation in the implication graph (cf. section 4.9.2). The poor general preprocessing indicates that hardly nothing is known about the implication graph and finding appropriate sets R^* might be difficult. Therefore, in the cut-and-branch algorithm we use violated cuts corresponding to $R^* = \{r_0\}$ only.

Tables 6.9 and 6.10 summarize the result of the cut-and-branch procedure and tables 6.11 and 6.12 give the number and class of the generated cuts. The size of the problems moderately grows (40.39% constraints, 297% non-zeros for (COSTBLP) and 19.35% constraints, 255% non-zeros for (COSTILP)) but also the lower bound increases by 1.19% for (COSTBLP) and 2.07% for (COSTILP). The lower bound provided by the (COSTBLP) model is still slightly superior to the lower bound of (COSTILP).

	(6.36)	(6.30)	(6.33)	(6.26)	(6.32)	(6.31)
sp97ic	105	25	96	4	3	20
sp98ic	37	47	180	2	6	32
sp98ir	-	46	45	9	1	30
sp98ar	52	197	211	-	19	184

Table 6.11: Cut statistic for the (COSTBLP) model

	(6.36)	(6.30)	(6.33)	(6.26)	(6.32)	(6.31)
sp97ic	113	23	97	4	5	23
sp98ic	35	44	189	2	4	27
sp98ir	-	48	61	9	3	29
sp98ar	42	161	189	-	15	189

Table 6.12: Cut statistic for the (COSTILP) model

Tables 6.13 and 6.14 summarize the results of the branch-and-bound run with a time limit of 10 CPU hours. The global lower bound of the (COSTILP) provides a gain of 0.36% compared to the lower bounds of (COSTBLP). For (COSTILP) the branch-and-bound algorithm produces better feasible solution than for (COSTBLP) which are already satisfactory, with the exception of `sp98ar`. With a good upper bound the pruning criterion can be frequently applied for the (COSTILP) model and lead to an overall improved performance.

In section 4.7 we mentioned that the branch-and-bound method systematically increases the lower bound but generally does not produce a feasible solution until the branch-and-bound procedure regularly terminates. Within a time limit of 10 CPU hours only for `sp98ir` the branch-and-bound algorithm terminates regularly. For the instances `sp97ic` and `sp98ic` the branch-and-bound method, applied to the (COSTILP) formulation, generates good feasible solutions, but for the hardest instance `sp98ar`, the feasible solution produced within 10 CPU hours seems to be far away from the optimal solution. Even after successive application of some rounding heuristics,

	sp97ic	sp98ic	sp98ir	sp98ar
best solution	4.2660	*5.8956	**2.1968	*5.9599
best lower bound	3.9412	4.4514	**2.1968	5.2601
gap	8.24%	32.44%	0.00%	13.30%
# B&B nodes	5248	3881	15527	3820

Table 6.13: Branch-and-bound for improved (COSTBLP) with time limit of 10 hours

* No solution was found after 10 hours branch-and-bound with best-bound node selection. The solution was generated with a depth-first-search node selection.
** Branch-and-bound algorithm terminates after 23072.12 seconds with the optimal solution

	sp97ic	sp98ic	sp98ir	sp98ar
best solution	4.0454	4.5179	*2.1968	5.8199
best lower bound	3.9688	4.4680	*2.1968	5.2730
gap	1.93%	1.12%	0.00%	10.04%
# B&B nodes	7790	5764	1178	3501

Table 6.14: Branch-and-bound for improved (COSTILP) with time limit of 10 hours

* Branch-and-bound algorithm terminates after 2486.78 seconds with the optimal solution of `sp98ir`.

substantially reduction of the set of lines by deleting expensive lines, and various experiments with CPLEX parameter settings we do not obtain better feasible solutions.

With the following decomposition approach we succeed in generating convenient line plans for `sp98ar`. We decompose the problem by splitting the network in several parts. First, we split the network at the nodes Ah, Amf, Ddr, and Gdm. Network `sp98arSE` contains the south-eastern part of the network and consists of 71 nodes, 74 edges, and is equipped with 180 routes which completely run inside this part of the network. The north-western part of the network, named `sp98arNW`, consists of 51 nodes, 60 edges, and 228 routes. We can solve the model (COSTILP) of instance `sp98arSE` within several minutes. The optimal solution provides a line plan of cost 2.5337. The instance `sp98arNW` is much harder and cannot be solved within 10 hours computation time, but the branch-and-bound algorithms provides a feasible solution with objective value 2.8436. This yields a feasible solution of value 5.3823 for `sp98ar`. We can improve this solution by repeal some variable fixings. We apply the branch-and-bound algorithm to the (COSTILP) formulation of `sp98ar` and fix most of the variables corresponding to the optimal solution of `sp98arSE` and the good solution of `sp98arNW`. We repeal the fixing of variables corresponding to routes terminating at the split nodes Ah, Amf, Ddr, and Gdm. Hence we can use lines running across the border of the subnetworks and save some cost. Most of the variables are fixed and the branch-and-bound procedure applied to the corresponding (COSTILP) formulation terminates within a few minutes and provides a solution with objective value 5.3781.

Although, the decomposition provides a good method for large scale networks, the general use of this approach is limited. We also apply the decomposition to `sp98arNW` with split nodes Ac, Hfd, and Hrs (cf. figure 6.10). The resulting subproblems can be solved within some minutes but the combined optimal solutions provide a solution with value 2.9341 which is substantially larger than the best known solution of `sp98arNW`.

In this section we represent computational investigations for the models (COSTBLP) and (COSTILP). The lower bounding provided by the (COSTBLP) model seems to be superior to the lower bounding of (COSTILP). Nevertheless, the branch-and-bound algorithm generated much better feasible solutions for (COSTILP). Improved upper bounds are essential ingredients for a branch-and-bound algorithm in order to efficiently apply the pruning criterion. If we focus on a good feasible solution we should apply the (COSTILP) model. The (COSTBLP) model is superior for *proving* optimality of a promising feasible solution. Due to the reduced size of the (COSTILP) model the core memory represents a minor resource limit. Indeed, we can run

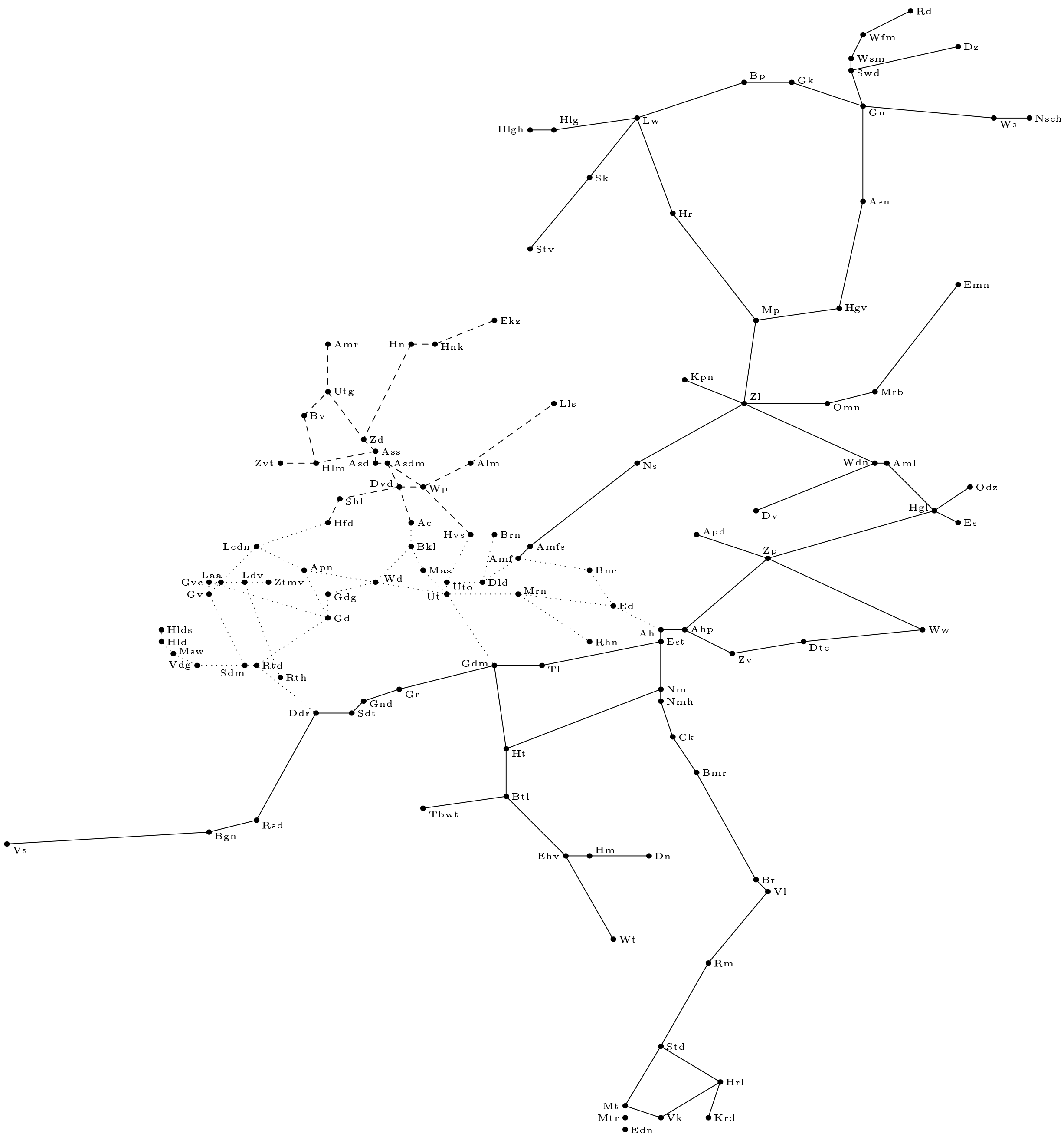


Figure 6.10: The subnetworks of the Dutch AggloRegio network `sp98ar`

	sp97ic	sp98ic	sp98ir	sp98ar
best solution	4.0377	4.5066	2.1968	5.3781
best lower bound	4.0000	4.4886	2.1968	5.2974
gap	0.94%	0.40%	0.00%	1.52%

Table 6.15: Branch-and-bound for (COSTILP) with no time limit

the branch-and-bound procedure several days without memory problems and increase the lower bounds. Table 6.15 presents the best known lower bounds and the resulting gaps according to this time exhaustive procedure. Although, we cannot solve the problems, except `sp98ir`, the (COSTBLP) and the (COSTILP) model seems to be a promising approach for the cost optimal line planning problem that yields a reasonable and practically sufficient performance guarantee.

6.7 Extension of the models

The models for the cost optimal approach described and analyzed above are applied to individual supply networks. With an extension of the models we can simultaneously solve the cost optimal line planning problem for a set of different supply networks. Therefore, let S be the set of the different systems, e.g. $S = \{\text{IC}, \text{AR}\}$. For reasons of simplicity assume that the edges of a system $\sigma \in S$ correspond to chains of edges of a fundamental supply network $\sigma_0 \in S$, e.g. $\sigma_0 = \text{AR}$. For example, the edge Amf-Zl of the InterCity network `sp98ic` consists of the chain Amf-Amfs-Ns-Zl of edges in the AggloRegio network `sp98ar`. We build a new network on top of the network corresponding to system σ_0 by adding the edges of the remaining supply networks. If we create parallel edges we sum up the relevant parameters ld, $\underline{lfr}$, and $\overline{lfr}$. Hence the line plans of the individual supply networks lead to a feasible solution of the combined problem but the exchange of expensive InterCity or InterRegio lines by cheaper AggloRegio lines may provide additional cost savings. Let ξ_σ with $\xi \in \{\mathfrak{c}^{t_{\text{fix}}}, \mathfrak{c}^{c_{\text{fix}}}, \mathfrak{c}^{t_{\text{var}}}, \mathfrak{c}^{c_{\text{var}}}, c_{\text{cap}}, \underline{c}, \overline{c}\}$ denote the parameters of the corresponding supply network, R_σ the set of possible routes and F_σ the corresponding frequencies. The following model gives an appropriate formulation of the combined cost optimal line planning problem.

$$\min \sum_{\sigma\in S}\sum_{r\in R_\sigma}\sum_{\varphi\in F_\sigma} \lceil \varphi\cdot\Gamma_r \rceil \left(x_{r,\varphi}\mathfrak{c}_\sigma^{\mathrm{t_{fix}}} + (\underline{c}^\sigma x_{r,\varphi} + y_{r,\varphi})\cdot \mathfrak{c}_\sigma^{\mathrm{c_{fix}}}\right) + d_r\cdot\varphi\cdot\left(x_{r,\varphi}\mathfrak{c}_\sigma^{\mathrm{t_{var}}} + (\underline{c}^\sigma x_{r,\varphi} + y_{r,\varphi})\cdot \mathfrak{c}_\sigma^{\mathrm{c_{var}}}\right)$$

s.t.

$$\sum_{\sigma\in S}\sum_{r\in R_\sigma,\, r\ni e}\sum_{\varphi\in F_\sigma} \varphi x_{r,\varphi} \geqslant \underline{lfr}(e) \qquad \forall\, e\in E \tag{6.37}$$

$$\sum_{r\in R_\sigma,\, r\ni e}\sum_{\varphi\in F_\sigma} \varphi x_{r,\varphi} \leqslant \overline{lfr}(e) \qquad \forall\, e\in E \tag{6.38}$$

$$\sum_{\sigma\in S}\sum_{r\in R_\sigma,\, r\ni e}\sum_{\varphi\in F_\sigma} c^\sigma_{\mathrm{cap}}\cdot\varphi(\underline{c}^\sigma x_{r,\varphi} + y_{r,\varphi}) \geqslant ld(e) \qquad \forall\, e\in E \tag{6.39}$$

$$y_{r,\varphi} \Leftrightarrow (\overline{c}^\sigma \Leftrightarrow \underline{c}^\sigma)\cdot x_{r,\varphi} \leqslant 0 \qquad \forall\,\sigma\in S \quad \forall\, r\in R_\sigma, \quad \forall\,\varphi\in F_\sigma \tag{6.40}$$

$$\sum_{\varphi\in F_\sigma} x_{r,\varphi} \leqslant 1 \qquad \forall\,\sigma\in S \quad \forall\, r\in R_\sigma \tag{6.41}$$

$$x\in\{0,1\}^{\sum_{\sigma\in S}|R_\sigma|\times|F_\sigma|},\ \ y\in\mathbb{Z}_+^{\sum_{\sigma\in S}|R_\sigma|\times|F_\sigma|} \tag{6.42}$$

Note that if the $\overline{lfr}$ values represent safety rules for the physical tracks we easily may change the summation index of (6.38) to include all lines with trains running via a particular track.

We would like to mention some comparisons of line plans designed with respect to the direct travelers as well as to the cost aspect. We only refer to the analysis of CLAESSENS et. al. because we cannot replicate these investigations due to the absence of reliable origin destination data for the networks. CLAESSENS et. al. [23] compare line plans for the small network depicted in figure 6.1. The cost optimal line plan L_{cost} was produced by solving the (COSTBLP) formulation of the combined problem with $S = \{\mathrm{IR},\mathrm{AR}\}$. The line plan L_{trav} with respect to the number of direct travelers was produced by the heuristic by DIENST (cf. section 5.3) and hence represents not necessarily a line plan that is optimal with respect to the number of direct travelers. Both line plans are compared on their operating cost and their number of direct travelers. Table 6.16 summarizes the relevant figures. As one may expect the cost optimal line plan is superior for cost aspects whereas the direct travel optimal line system is superior for service aspects. Nevertheless,

	L_{trav}	L_{cost}	relative difference
cost	9473	7845	-17%
unused seats	10335	5900	-42%
empty seat kilometers	66391	33606	-49%
coach kilometers	1970	1573	-20%
train kilometers	619	670	+8%
coaches needed per day	100	77	+8%
average trains length (in coaches)	3.2	2.7	-15%
average route length (in kilometers)	51.6	37.2	-27%
direct travelers	65996	62051	-6%

Table 6.16: Comparison of the cost and the direct traveler approach

the cost savings of L_{cost} get a better percentage than the increase of service of L_{trav}. In the following discussion we give some arguments that extenuate the results of CLAESSENS et. al. at least for larger and more general networks.

First of all note that the network under consideration is rather small, hence the travel routes are quite short and *every* line plan will provide a large number of direct travelers. With a total volume of traffic of 68200 passengers in this network almost all travelers are provided with a direct connection in L_{trav} as well as in L_{cost}. The ratio of direct travels and total number of travels points out the particularity of the network. For larger networks the number of direct travelers in cost optimal line plans will significantly reduce. Nevertheless, for the Dutch railroad system this might be a justifiable approach because the average travel distance in the Netherlands is only about 44 kilometers and the average edge length in the InterCity network is approximately 48 kilometers (for the InterRegio it is 19 kilometers) [43]. Hence, in any line plan, most travelers have a direct connection. These results for the Netherlands cannot be generalized for other railroad networks, e.g. the average travel distance in the German railroad network is about 285 kilometers with average edge length of 60 (38) kilometers for the InterCity (InterRegio) network.

Finally, the cost calculation requires a detailed review. The real cost arising for the production of a line plan and the cost computed by the cost optimal approach might substantially differ because the operational cost used in the models gives only a rough estimation. The circulation of rolling stock and personnel finally is responsible for the cost of the railroad system. Furthermore, one should carefully discuss the cost decrease, which is due to replacing expensive, fast lines, say InterCity lines or InterRegio lines, by cheaper but slower and less attractive AggloRegio lines. Corresponding cuts in cost may reduce the attraction of the railroad system. In particular, CLAESSENS et. al. yield cost saving of 5.2% for the simultaneous determination of line plans compared to individual cost optimal line plans. We combined the supply networks `sp98ic` and `sp98ir`, but within 10 CPU hours time we did not find any feasible solution.

We conclude this chapter with a small example which clearly shows the limits of a pure cost optimal approach. Consider the small network presented in figure 6.11. Suppose the volume of traffic of the origin-destination pair a,c consists of 100 passengers and the capacity of the lines $a \Leftrightarrow b \Leftrightarrow c$, $a \Leftrightarrow b \Leftrightarrow d$, and $c \Leftrightarrow b \Leftrightarrow d$ provides a capacity of 50 (seats). The capacity demand of the edges ab and bc is obviously fulfilled by the lines but half of the passengers of origin-destination pair a,c must change the line at b. If we apply this line plan in practice, all passengers of origin-destination pair a,c will use the direct connection and produce an overcrowded line $a \Leftrightarrow b \Leftrightarrow c$ which leads to reduction in the attraction of the system and finally results in the loss of customers. Nowadays, practitioners at NS overcome this general difficulty in the following way. A convenient way of verifying the results of an optimization is to simulate the line plan in a more realistic environment (cf. section 3.4). The line

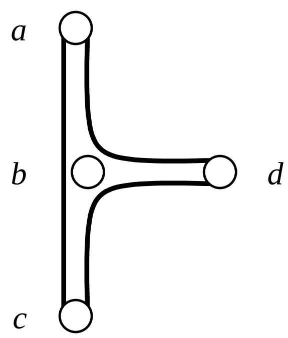


Figure 6.11: Small example

plan generated by the cost approach is passed to a simulation tool named PROLOP[3]. PROLOP simulates the behavior of the travelers based on the routes and the frequencies of the line plan but omits the capacity (number of coaches). The simulation assigns the passengers to convenient lines which results in a particular load of each line. For our small example, PROLOP would assign most passengers of origin-destination pair a, c to the line $a \Leftrightarrow b \Leftrightarrow c$. The capacity of a line is determined with respect to the load of a line and will in general destroy the cost savings provided by the cost optimal line planning approach. This justifies neglecting the consideration of holes in the domain of possible coach numbers in the (COSTILP) model. Furthermore, we can relax the integrality requirement of the y variables and hence (COSTILP) becomes a binary mixed integer linear program with a substantially reduced number of integer variables.

[3]PROLOP is a commercial simulation tool of the Adtranz Signal GmbH.

Chapter 7

Conclusions and suggestions for further research

In this thesis we present a mathematical programming approach to one of the fundamental planning tasks in public rail transport, namely the line planning problem. We investigate both, the mathematical and the practically aspects of the line optimization problem. From the theoretical point of view we are faced with an intractable optimization problem. A bunch of heuristic procedures has been introduced since the first paper concerning the line optimization problem has been published in 1925. This monograph adds an approach which is based on integer linear programming. Integer linear programming provides a powerful tool for modeling various aspects of the line optimization problem. Moreover, with relaxation algorithms for integer linear programs, like linear programming based branch-and-bound, we obtain feasible solutions as well as bounds for an optimal solution which results in a performance guarantee. Nevertheless, creative mathematical work is essential in order to achieve an acceptable performance guarantee in reasonable computation times for instances arising from large scale, real-world data. A practical improvement of an algorithm almost precedes a "*roundabout way*" via theoretical results concerning structural properties of mathematical objects. In this thesis we have the ability to discover and prove (more or less) elegant theorems dealing with such properties and to apply these results in order to solve a practically relevant problem.

Beyond this general aspects concerning the persuasive power of a mathematical programming approach there are several concrete prospects of further research.

First of all a combination of the service- and the cost-oriented approach should be discussed. The cost-oriented models introduced in chapter 6 can be easily extended to represent direct travelers in the transportation network. For example, let $z_{r,a,b} \in \mathbb{Z}_+$ denote the number of direct travelers commuting between a and b in a line on route r. Similar to the (LOP) model the values of this kind of variables are subject to

$$\begin{aligned}
\sum_{r \in R_{a,b}} z_{r,a,b} &\leqslant T^{a,b} && \forall\, a,b \in V_T^2 \\
\sum_{\substack{a,b \in V_T^2 \\ r \in R_{a,b}, e \in r_{a,b}}} z_{r,a,b} &\leqslant c_{\text{cap}} \cdot \varphi(\underline{c} \cdot x_{r,\varphi} + y_{r,\varphi}) && \forall\, \varphi \in F\,,\, r \in R\,,\, e \in r
\end{aligned}$$

in the (COSTILP) model. The notation can be looked up in chapter 5 and 6. Due to the contrary objectives we cannot simply add the sum of the z variables in the objective. On the one hand, if the planners can determine a trade-off between cost and service, we could add a negative weight to the z variables in the objective. On the other hand, we could concentrate on *pareto-optimal* solutions. A solution (x,y,z) is called pareto-optimal if there is no other solution (x',y',z') that provides more direct travelers at lower cost. Furthermore, we could add a constraint

$$\sum_{a,b\in V_T^2}\sum_{r\in R_{a,b}} z_{r,a,b} \geqslant t \cdot \sum_{a,b\in V_T^2} T^{a,b}$$

with a parameter $t \in [0,1]$ that eliminates line plans with a small number of direct travelers.

This is a straight forward idea of combining the service and the cost approach and we have to set up the whole machinery of model relaxation and improvement in order to obtain practically relevant results from such a model.

All models presented in this thesis rely on the fact that the traffic load ld for all edges is known in advance. This is a justifiable assumption if we completely believe in the system split procedure introduced in chapter 3. The system split distributes the passengers along the network without having a line plan in mind. A concrete line plan significantly changes the behavior of the travelers and hence the ld values. An approach that simultaneously models the finding of a line plan and the behavior of all travelers (not only the direct travelers) is of substantial interest. Nowadays, practitioners overcome this problem by iteratively solving the line optimization problem, simulating the results, and adjusting the input parameters (including the ld values) of the optimization run.

If this method converges, we are aware of a *traffic-flow stable line plan*. Obviously, this methods does not necessarily converge and if it does, the resulting line plan represents a feasible solution of a simultaneous model, only.

For the Netherlands there are some interesting ideas of relaxing the rigid separation in IC/ICE, IR, and AR supply networks. Currently, a line of system $X \in \{\mathrm{IC/ICE}, \mathrm{IR}, \mathrm{AR}\}$ must stop at all stations belonging to system X on its route. In the new model the line may skip a stop of system X or may include a stop at a station of an inferior system. For example, at the line's end it may look like a local train and in the intermediate part it may act like a fast train. A line in this model consists of a route, a frequency, a capacity, and the intermediate stops. For this approach, a well modeled and dynamic behavior of the passengers represents a substantial part.

In the cost-oriented approach described in chapter 6 the lines are planned with respect to a particular circulation of rolling material. An integration of planning tasks with contrary objectives is important for generating a satisfactory solution for the complete planning process. Currently, a combination of the most important planning tasks line planning, train scheduling, and circulation of rolling stock and personnel seems to be impossible with today's models, methods, and computer power. Therefore, peculiarities of subsequent planning tasks should be taken into account whenever possible.

Research in the field of discrete optimization for railroad planning problems and particularly for the line planning problem combines the fascinating world of applied or practical mathematics with the chance of succeeding in solving real-world problems.

List of Figures

List of Tables

Bibliography

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows — Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] J. Allan, C.A. Brebbia, R.J. Hill, G. Sciutto, and S. Sone, editors. *Railway Systems and Management*, volume 1 of *Computers in Railways V*. Computational Mechanics Publications, 1996.

[3] R.N. Anthony. Planning and control systems: A framework for analysis, 1965. Harvard, Bosten.

[4] J. Arós, W.H. Cunningham, J. Edmonds, and J. Green-Krótki. Reductions to 1-matching polyhedra. *Networks*, 13:455–473, 1983.

[5] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discr. Comput. Geom*, 8:295–313, 1992.

[6] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj. Gomory cuts revisited. *Oper. Res. Lett.*, 19:1–9, 1996.

[7] R. Barbour and J.D. Fricker. Estimating an origin-destination table using a method based on shortest augmenting paths. *Transportation Res. Part B*, 28(2):77–89, 1994.

[8] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for huge integer programs. *Oper. Res.*, 46(3):316–329, 1998.

[9] D. Bienstock and O. Guenluek. Capacitated network design – polyhedral structure and computation. *INFORMS J. Comput.*, 8(3):243–259, 1996.

[10] R.E. Bixby. private communication, 1996.

[11] A. Brook, D. Kendrick, and A. Meeraus. *GAMS: A User's Guide, Release 2.25*. Scientific Press, 1992.

[12] M.R. Bussieck, P. Kreuzer, and U.T. Zimmermann. Optimal lines for railway systems. *European J. Oper. Res.*, 96:54–63, 1996.

[13] M.R. Bussieck, T. Winter, and U.T. Zimmermann. Discrete optimization in public rail transport. *Math. Programming*, 79(1–3):415–444, 1997.

[14] A. Caprara, M. Fischetti, P. Toth, D. Vigo, and P.L. Guida. Algorithms for railway crew management. *Math. Programming*, 79(1–3):125–141, 1997.

[15] A. Car and A. Frank. General principles of hierarchical spatial reasoning — The case of wayfinding. In *Proceedings of the 6th International Symposium on Spatial Data Handling*, pages 646–654, Edinburgh, Scotland, 1994.

[16] M. Carey. A model and strategy for train pathing with choice of lines, platforms, and routes. *Transportation Res. Part B*, 28(5):333–353, 1994.

[17] E. Cascetta and S. Nguyen. A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportation Res. Part B*, 21:437–455, 1987.

[18] Y.L. Chen. Finding the *k* quickest simple paths in a network. *Inform. Process. Lett.*, 50:89–92, 1994.

[19] T. Christof. Ein Verfahren zur Transformation zwischen Polyederdarstellungen. Master's thesis, Universität Augsburg, May 1991.

[20] T. Christof. `PORTA` – POlyhedron Representation Transformation Algorithm. Sourcecode available via `ftp://ftp.zib-berlin.de/pub/mathprog/polyth/porta`, 1994.

[21] V. Chvátal. *Linear programming*. Freemann & Co., 1983.

[22] M.T. Claessens. De kost-lijnvoering. Master's thesis, University of Amsterdam, 1994.

[23] M.T. Claessens, N.M. van Dijk, and P.J. Zwaneveld. Cost optimal allocation of rail passenger lines. *European J. Oper. Res.*, 110(3):474–4489, 1998.

[24] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley & Sons, 1998.

[25] CPLEX Optimization. *Using the CPLEX callable library*.

[26] H. Crowder, E.L. Johnson, and M.W. Padberg. Solving large-scale zero-one linear programming problems. *Oper. Res.*, 31:803–834, 1983.

[27] H. Dienst. *Linienplanung im spurgeführten Personenfernverkehr mit Hilfe eines heuristischen Verfahrens*. PhD thesis, TU Braunschweig, 1978.

[28] B.L. Dietrich and L.F. Escudero. Coefficient reduction for knapsack-like constraints in 0-1 programs with upper bounds. *Oper. Res. Lett.*, 9(1):9–14, 1990.

[29] G. Dijkhuizen and U. Faigle. A cutting-plane approach to the edge-weighted maximal clique problem. *European J. Oper. Res.*, 69:121–130, 1993.

[30] J. Dreier and U. Brockmann. Integrated system planning for railways. In Allan et al. [2], pages 155–168.

[31] G. Ertl. Shortest path calculation in large road networks. *OR Spektrum*, 20(1):15–20, 1998.

[32] S. Even. *Graph algorithms*. Computer Science Press, 1979.

[33] R.W. Floyd. Algorithm 97, shortest path. *Comm. ACM*, 23(6):350–351, 1962.

[34] GAMS Development Corp. GAMS/CPLEX 4.0 User Notes. URL http://www.gams.com/solvers/cplex/cplexman.htm, 1997.

[35] M.R. Garey and D.S. Johnson. *Computers and Intractability*. Freeman & Co., 1979.

[36] R.E. Gomory. Outline of an algorithm for integer solutions of linear programs. *Bull. Amer. Math. Soc. (N.S.)*, 64:275–278, 1958.

[37] M. Grötschel, A. Löbel, and M. Völker. Optimierung des Fahrzeugumlaufs im öffentlichen Nahverkehr. In Hoffmann et al. [39], pages 609–624.

[38] M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, 1988.

[39] K.-H. Hoffmann, W. Jäger, Th. Lohmann, and H. Schunck, editors. *Mathematik – Schlüsseltechnologie für die Zukunft*. Springer, 1997.

[40] K. Jörnsten and S.W. Wallace. Overcoming the (apparent) problem of inconsistency in origin-destination matrix estimations. *Transportation Sci.*, 27(4):374–380, 1993.

[41] M. Jünger, G. Reinelt, and S. Thienel. Provabily good solutions for the traveling salesman problem. *Z. Oper. Res. Ser. A-B*, 40:183–217, 1994.

[42] P. Kreuzer. Linienoptimierung im schienengebundenen Personenverkehr. Master's thesis, TU Braunschweig, 1994.

[43] M. Krista. *Verfahren zur Fahrplanoptimierung dargestellt am Beispiel der Synchronzeiten*. PhD thesis, TU Braunschweig, 1996.

[44] M. Laurent and S. Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra Appl.*, 223–224:439–461, 1995.

[45] E.L. Lawler, J.K. Lenstra, A.H.G Rinnooy Kan, and D.B. Shmoys, editors. *The Travelling Salesman Problem*. John Wiley & Sons, 1985.

[46] T. Lengauer. *Combinatorial Algortihms for Integrated Circuit Layout*. John Wiley & Sons, 1990.

[47] A. Löbel. *Optimal vehicle scheduling in public transit*. PhD thesis, Technical University Berlin, 1997.

[48] P. Modesti and A. Sciomachen. A utility for finding multiobjective shortest paths in urban multimodal transportation networks. Technical report, University Genova, Italy, 1996.

[49] K. Nachtigall. Cutting planes for a polyhedron associated with a periodic network. Technical Report IB 112-96/17, Inst. f. Flugführung Braunschweig, July 1996.

[50] K. Nachtigall. Periodic network optimization with different arc frequencies. *Discrete Appl. Math.*, 69:1–17, 1996.

[51] F. Nelissen. private communication, 1997.

[52] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[53] M.A. Odijk. A constraint generation algorithm for the construction of periodic railway timetable. *Transportation Res. Part B*, 30(6):455–464, 1996.

[54] M.A. Odijk, M.S.P.A.M. Maarschalkerweerd, and E.A.G. Weits. Timetable-independent design and analysis of railway infrastructure. In Allan et al. [2], pages 3–12.

[55] C. Oltrogge. *Linienplanung für mehrstufige Bedienungssyteme im öffentlichen Personenverkehr*. PhD thesis, TU Braunschweig, 1994.

[56] C.H. Papadimitriou. Polytopes and complexity. In W.R. Pulleyblank, editor, *Progress in Combinatorial Optimization*, pages 295–305. Academic Press, 1984.

[57] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization — Algorithms and Complexity*. Prentice Hall, 1982.

[58] U. Pape, Y.-S. Reinecke, and E. Reinecke. Line network planning. In J.R. Daduna, I. Branco, and J.M.P. Paixao, editors, *Computer-Aided Transit Scheduling*, volume 430 of *Lecture Notes in Economic and Mathematical Systems*, pages 1 – 7. Springer, 1995.

[59] K. Park, K. Lee, and S. Park. An extended formulation approach to the edge-weighted maximal clique problem. *European J. Oper. Res.*, 95(3):671–682, 1996.

[60] C. Pascolo, P. Serafini, and W. Ukovich. A hierarchical approach to periodic scheduling of large scale traffic light systems. In *Methodology, implementation and applications of decision support systems*, volume 320 of *Proc. Sch., Udine/Italy*, pages 199–209. CISM Courses Lect, 1990.

[61] A. Patz. Die richtige Auswahl von Verkehrslinien bei großen Straßenbahnnetzen. *Verkehrstechnik*, 50/51, 1925.

[62] K. Pierick and K.-D. Wiegand. Methodische Ansätze für eine Optimierung des schienengebundenen Personenfernverkehrs. *Schienen der Welt*, 46:197–203, 1976.

[63] D.J. Rosenkrantz, R.E. Stearns, and P.M. Lewis. An analysis of serveral heuristics for the traveling salesman problem. *SIAM J. Comput.*, 6:563–581, 1977.

[64] K.N.A. Safwat and T.L. Magnanti. A combined trip generation, trip distribution, modal split, and traffic assignment model. *Transportation Sci.*, 22(1):14–30, 1988.

[65] M.W.P. Savelsbergh. Preprocessing and probing techniques for mixed integer programming problems. *ORSA J. Comput.*, 6(4):445–454, 1994.

[66] A. Schrijver. Minimum circulation of railway stock. *CWI Q.*, 3:205–217, 1993.

[67] P. Serafini and W. Ukovich. A mathematical model for periodic scheduling problems. *SIAM J. Disc. Math.*, 2(4):550–581, 1989.

[68] H.D. Sherali, R. Sivanandan, and A.G. Hobeika. A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes. *Transportation Res. Part B*, 28(3):213–233, 1994.

[69] C. Simonis. Optimierung von Omnibuslinien. Berichte Stadt – Region – Land, Institut für Stadtbauwesen, RWTH Aachen, 1981.

[70] H. Sonntag. *Linienplanung im öffentlichen Personennahverkehr*. PhD thesis, TU Berlin, 1977.

[71] H. Sonntag. Ein heuristisches Verfahren zum Entwurf nachfrageorientierter Linienführung im öffentlichen Personennahverkehr. *Z. Oper. Res. Ser. A-B*, 23:B15–B31, 1979.

[72] V. Stausberg. Ein Dekompositionsverfahren zur Berechung kürzester Wege auf fast planaren Graphen. Master's thesis, University of Cologne, 1995.

[73] S.Warshall. A theorem on boolean matrices. *Comm. ACM*, 9(1):11–12, 1962.

[74] S. Thienel. *ABACUS – A Branch-And-CUt System*. PhD thesis, University of Cologne, 1995.

[75] J.H.A. van den Berg and M.A. Odijk. DONS: Computer aided design of regular service timetables. Technical report, 1996.

[76] T.J. van Roy and L.A. Wolsey. Solving mixed integer programming problems using automatic reformulation. *Oper. Res.*, 33(1):45–57, 1987.

[77] P.H. Vance, C. Barnhart, E.L. Johnson, and G.L. Nemhauser. Solving binary cutting stock problems by column generation and branch-and-bound. *Comput. Optim. and Applics.*, 3:111–130, 1994.

[78] F. Vanderbeck. *Decomposition and Column Generation for Integer Programs*. PhD thesis, Universität Louvain, 1994.

[79] H. Wegel. *Fahrplangestaltung für taktbetriebene Nahverkehrsnetze*. PhD thesis, TU Braunschweig, 1974.

[80] R. Weismantel. On the $0/1$ knapsack polytope. *Math. Programming*, 77A(1):49–68, 1997.

[81] H.P. Williams. *Model building in mathematical programming*. John Wiley & Sons, 1990.

[82] P.J. Zwaneveld, M.T. Claessens, and N.M. van Dijk. A new method to determine the cost optimal allocation of passenger lines. In *Defence or Attack: Proceedings 2nd TRAIL PhD Congress 1996, part 2*, Delft/Rotterdam, 1996. TRAIL Research School.

Index

Deutsche Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit dem Thema der optimalen Linienführung für getaktete spurgeführte Verkehrsysteme, wie der Eisenbahn. Der Linienführung zugrundeliegende Linienplan besteht aus einer Menge von Wegen im Schienennetz, denen Takte bzw. Frequenzen zugeordnet sind. Ziel der Linienoptimierung ist es, einen Linienplan zu finden, der unter Berücksichtigung des Verkehrsaufkommens eine Zielfunktion optimiert.

Die Linienplanung ist in den Prozess der hierarchischen Verkehrsplanung eingebettet. Relevante Ansätze der mathematischen Optimierung aus der Literatur für die einzelnen Aufgaben dieses hierarchischen Prozesses stehen in Kapitel 2 im Mittelpunkt.

Die Linienplanung läßt sich erneut in vier hierarchisch angeortnete Aufgaben zerteilen, unter denen die Linienoptimierung den zentralen Aspekt darstellt. Kapitel 3 enthält eine ausführliche Beschreibung der Linienplanungsdekomposition sowie einen Überblick über Ansätze zur Linienoptimierung aus der verkehrstechnischen Literatur. Die vorgeschlagenen Verfahren beruhen ausschließlichen auf Heuristiken, die keine Gütegarantien für generierte Lösungen liefern können.

Die in dieser Arbeit präsentierten Modelle zur Linienoptimierung basieren auf ganzzahligen linearen Programmen. Kapitel 4 stellt die notwendigen theoretischen Konzepte zur Modellierung, Klassifizierung und Lösung dieser Programme bereit und präsentiert fundamentale Modelle und Ergebnisse zur Komplexität verschiedener Varianten des Linienoptimierungsproblems.

Es zeigt sich schnell, daß zur realistischen Bewertung von Linienplänen einfache lineare Zielfunktionen, die einer Linie in (linearer) Abhängikeit von der Frequenz einen Wert bzw. Kosten zuordnen, nicht ausreichen. In Kapitel 5 und 6 werden zwei praktisch relevante Bewertungen von Linienplänen präsentiert.

Die Modelle aus Kapitel 5 bewerten die Linienpläne aus Kundensicht und nutzen dazu das Konzept der Direktfahrer. Als Direktfahrer werden Reisende bezeichnet, die ohne Umsteigen ihr Reiseziel erreichen. Für reale Daten läßt sich ein ganzzahliges lineares Programm zur Bestimmung eines Linienplans mit maximaler Anzahl von Direktfahrern mit heute verfügbaren Techniken und Computern nicht lösen. Mittels einer Relaxation von Kapazitätsrestriktionen läßt sich ein extrem verkleinertes Modell ableiten, das einen Linienplan und eine obere Schranke für die Optimallösung des ursprünglichen Programms liefert. Die Anzahl der Direktfahrer im generierten Linienplan stellt eine untere Schranke für die gesuchte Optimallösung dar, so daß für eine konkrete Instanz des Linienoptimierungsproblems eine Lösung und eine Gütegarantie angegeben werden kann. Das Kapitel wird abgerundet mit potentiellen Erweiterungen der Modelle und einer polyedrischen Analyse des zugrundeliegenden Lösungsraums.

Kapitel 6 präsentiert ein Modell zur kostenorientierten Bewertung von Linienplänen. Bei diesem Vorgehen sind neben den Wegen und Frequenzen zusätzlich die Kapazitäten der Züge festzulegen. Ausgehend von einem ganzzahligen quadratischen Programm werden zwei Linearisierungen des Modells zur kostenoptimalen Linienplanung vorgestellt und verglichen. Der Einsatz von Methoden aus der polyedrischen Optimierung führt zur Verbesserung beider linearer Modelle, so daß nun auch Instanzen von praktisch relevanten Größe mit ausreichender Genauigkeit in akzeptabler Zeit gelöst werden können.

Die Arbeit schließt mit einem Ausblick für die zukünftige Entwicklung mathematischer Ansätze zum Linienoptimierungsproblem.