



Faculty of Electrical Engineering
Department of Computer Science

Master's thesis

Optimal Scheduling in Railway Container Transport Optimization

Josef Zelinka

May 24, 2024

Supervisor: Ing. Vilém Heinz

I. Personal and study details

Student's name: **Zelinka Josef** Personal ID number: **492328**
Faculty / Institute: **Faculty of Electrical Engineering**
Department / Institute: **Department of Computer Science**
Study program: **Open Informatics**
Specialisation: **Artificial Intelligence**

II. Master's thesis details

Master's thesis title in English:

Optimal Scheduling in Railway Container Transport Optimization

Master's thesis title in Czech:

Optimální rozvrhování v rámci optimalizace vlakové kontejnerové dopravy

Guidelines:

In this diploma thesis, the student will solve a real-world problem of railway container freight transport defined in cooperation with the company Metrans. The main goal is to create an exact approach that will automatically and optimally solve the scheduling of freight trains for incoming orders while minimizing logistics costs [1] and fulfilling the defined conditions. While similar problems have been tackled in the scientific literature [2] [3], our problem-specific data and constraints are too different, requiring a completely new approach. The conditions and aspects considered, among others, will be:

- All orders must be fulfilled in a given time window.
- Available train corridors provided by railway maintainers are used.
- Price for train departures, train carriage use, and penalty for uneven utilization of trains given their departure and arrival terminals are considered.
- Price for the distance traveled is considered.
- The optimization of container swaps between trains is also considered [4], which tends to be the complicating factor often omitted from the existing scientific papers [5].

The student will:

- Formally define the problem based on the requirements obtained from the Metrans company.
- Revise current state-of-the-art literature on the topic, specifically focusing on exact approaches and formulations to extract useful information about the approach and limitations of the exact approaches.
- Clean the data provided by Metrans and create testing datasets based on them.
- Develop an exact approach based on Constraint Programming to solve the problem with a focus on the optimality of the solution.
- Evaluate the effectiveness of the proposed approach on differently sized instances, analyzing the scalability of exact approaches.

Bibliography / sources:

[1]

Xinghan Chen, Tianshuai Zuo, Maoxiang Lang, Shiqi Li, Siyu Li,
Integrated optimization of transfer station selection and train timetables for road-rail intermodal transport network,
Computers & Industrial Engineering,
Volume 165,
2022,
107929,
ISSN 0360-8352,
<https://doi.org/10.1016/j.cie.2021.107929>.
(<https://www.sciencedirect.com/science/article/pii/S0360835221008330>)

[2]

Heggen, Hilde and Molenbruch, Yves and Caris, An and Braekers, Kris,
Intermodal Container Routing: Integrating Long-Haul Routing and Local Drayage Decisions,
Sustainability,
Volume 11,
2019,
1634,
ISSN 2071-1050,

<https://doi.org/10.3390/su11061634>.
(<https://www.mdpi.com/2071-1050/11/6/1634>)

[3]

Tsung-Sheng Chang,
Best routes selection in international intermodal networks,
Computers & Operations Research,
Volume 35, Issue 9,
2008,
Pages 2877-2891,
ISSN 0305-0548,
<https://doi.org/10.1016/j.cor.2006.12.025>.
(<https://www.sciencedirect.com/science/article/pii/S0305054806003273>)

[4]

Lamiaa Dahite, Abdeslam Kadrani, Rachid Benmansour,
Optimization models for train load and transport planning problems,
MATEC Web Conf.,
Volume 200,
2018,
Pages 10,
<https://doi.org/10.1051/mateconf/201820000010>
(https://www.matec-conferences.org/articles/mateconf/abs/2018/59/mateconf_iwtsce2018_00010/mateconf_iwtsce2018_00010.html)

[5]

Claudia Archetti, Lorenzo Peirano, M. Grazia Speranza,
Optimization in multimodal freight transportation problems: A Survey,
European Journal of Operational Research,
Volume 299, Issue 1,
2022,
Pages 1-20,
ISSN 0377-2217,
<https://doi.org/10.1016/j.ejor.2021.07.031>.
(<https://www.sciencedirect.com/science/article/pii/S0377221721006263>)

Name and workplace of master's thesis supervisor:

Ing. Vilém Heinz Department of Control Engineering FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **01.02.2024** Deadline for master's thesis submission: _____

Assignment valid until: **21.09.2025**

Ing. Vilém Heinz
Supervisor's signature

Head of department's signature

prof. Mgr. Petr Páta, Ph.D.
Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

Date of assignment receipt

Student's signature



Declaration

I declare that the presented work was developed independently and that I have listed all sources of the information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 24, 2024

.....
Josef Zelinka



Acknowledgement

I would like thank my supervisors Ing. Vilém Heinz and prof. Dr. Ing. Zdeněk Hanzálek, who lead me on the path to writing this thesis. Secondly, I am really grateful for the support of my family and friends, who were always kind to encourage me throughout the harder times of my studies. Lastly, I am very thankful for the railway modeling community *Železnice Podbaba* as it provided me with useful background knowledge in railway domain and retreat in times calling for relaxation.

Abstrakt

Nyní je většina zboží na planetě přepravována pomocí intermodálních kontejnerů, které lze snadno překládat mezi různými druhy dopravy. Díky jednoduché manipulaci lze pro přepravu použít rozmanité dopravní prostředky. Nalezení cesty pro jeden kontejner není složitý problém, ale po přidání mnoha kontejnerů s různými zdroji, cíli a časovými omezením se problém stane NP složitým. Pro nalezení řešení jsou v této práci použity přístupy kombinatorické optimalizace: celočíselné lineární programování a programování za pomoci omezujících podmínek. Výsledky obou přístupů jsou porovnány na souboru dat vytvořených ze záznamu fungování nákladního železničního dopravce. Jelikož je problém NP složitý, optimální přístupy se zaměřují na menší instance, ale dokazují, že optimalizace může poskytnout lepší výsledky než lidský operátor.

Klíčová slova: operační výzkum, železnice, kombinatorická optimalizace



Abstract

Nowadays, most of the goods on the planet are transported using intermodal containers, which can be easily transferred between different modes of transport. The ease of transfer enables the composition of the journey from multiple types of transport. Routing a single container is a simple problem, but introducing many containers with different sources, destinations, and time constraints makes the problem much more complex (specifically NP-hard). Thus, this thesis studies such problems using combinatorial approaches: Integer Linear Programming and Constrained Programming. The performance of the approaches is compared on a real-world dataset created from data provided by a freight railway operator. As the problem is NP-hard, the optimal approaches tackle smaller instances but prove the optimization can provide better results than the human operators.


Keywords: operations research, railway, combinatorial optimization

Contents

1	Introduction	1
1.1	On Railway Transportation	3
1.2	METRANS	5
1.2.1	Current Operation	6
1.3	Thesis Structure	6
2	Mathematical Programming	9
2.1	Integer Linear Programming	9
2.2	Constraint Programming	11
2.3	Differences between ILP and CP	11
2.4	Solver Wars	12
3	Related Work	15
3.1	Enterprise Resource Planning Software - ERP	15
3.2	Optimization Approaches	16
3.2.1	Possibilities of Optimization on the Railway Networks	16
3.2.2	Freight Transportation Optimization	18
4	Problem Statement	21
4.1	NP Completeness	23
4.1.1	Reduction from NP-Complete Problem	23
5	Method	25
5.1	ILP Formulation	25
5.2	CP Formulation	27
5.2.1	CP Model with Integer Variables	27
5.2.2	CP Model with Interval Variables	27
5.3	Dataset Analysis	28
5.4	Dataset Preprocessing	31
5.5	Solution Quality Measures	33
6	Results	37
6.1	Minimal Instance	37
6.2	Big M Influence	38
6.3	Restriction Constraints in ILP Model	38
6.4	Capabilities of a CP Model with Integer Variables	39
6.5	Capabilities of CP Model with Interval Variables	39
6.6	Building the Models	40
7	Conclusion & Future Work	43
	References	45
A	Running the Experiments	49

List of Figures

1	Intermodal containers	1
2	Train with intermodal containers	2
3	Praha-Uhřetěves intermodal terminal.	2
4	An example of a grafikon.	3
5	Overview of a typical railway station and an intermodal terminal.	4
6	The locations of container terminals accessed by METRANS [1].	6
7	ERP software	15
8	Overview of a minimal instance of the container routing problem.	22
9	Reduction of the 3-partition problem to the container routing problem.	23
10	The number of trains operated each week and sorted according to weekdays.	29
11	Trains arriving and departing combined from all stations during an individual hour.	29
12	The number of containers handled by the most utilized terminals.	30
13	The overview of the dataset preprocessing pipeline.	30
14	Distance generation	32
15	Dataset network	33
16	The distribution of the train lengths in carriages.	34
17	The distribution of the train lengths in TEU.	34
18	The distribution of the train utilizations.	35
19	Minimal instance.	37
20	Build and solution times	41



List of Tables

1	Feature overview of the most popular solvers.	12
2	Overview of the used symbols	21
3	Used variables	25
4	Grouping of the constraints present in the model.	26
5	Grouping of the similar constraints between the two modeling paradigms.	28
6	The cost of the plan created by the operators and the ILP model.	33
7	The costs of train slots between the terminals.	37
8	The influence of manipulation on the assigned route to the container c_2	38
9	The effect of the big M constraints.	38
10	Comparison of ILP models	39
11	Performance of the small CP model.	39
12	Comparison of the CP and ILP models.	40
13	Comparison of the build times.	41

Chapter 1

Introduction

Since the beginning of the Industrial Revolution, the world has become increasingly global. At first, just precious items were transported across the oceans and continents. However, as the ships got bigger, the amount of cargo rose as well. The ever-increasing amount of goods carried in varying shapes and sizes signaled a need for standardization.

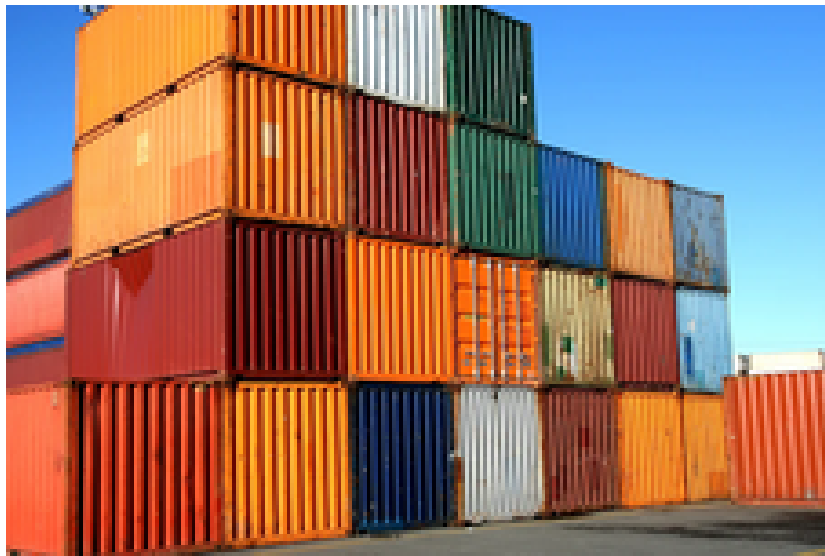


Figure 1: The intermodal containers being stacked on top of each other, removing the need for large warehouses in the ports [2].

This resulted in the creation of standardized containers shown in Figure 1 coming just in a few sizes. The containers are constructed so they can be transported using multiple modes of transport. Thus, they are often called intermodal. Their width allows them to fit on conventional trucks and even pass through railway tunnels hauled on carriages. Additionally, corrugated metal held together using multiple steel beams makes the containers ready for stacking, which can be fully utilized on ocean vessels and while storing the containers.

The containers create a unified interface for handling smaller packages. Later, all goods in the container are moved by a large crane simultaneously. The crane operators only need to pay a little attention to the type of goods inside the container as the interface is simply the ordinal metal box. Such improvement increased the volume of trade even more as the goods did not have to be handled sack by sack or box by box while changing the mode of transport in harbors or railway stations.

Railway operators followed a very similar path to their maritime counterparts as multiple modes of transport often handle the same cargo. Centuries ago, individual-covered carriages were primarily loaded manually or with the help of a small forklift. However, most internationally transported cargo is moved inside intermodal containers, which need flat railway carriages equipped with spikes to secure the container. An example of a train loaded with intermodal containers is depicted in Figure 2.

The system of most railway container operators consists of multiple terminals and trains

1. Introduction



Figure 2: METRANS train hauling a load of containers from Koper to Dunajská Streda terminal [3].

running between them. A terminal is where most container handling is done, as shown in Figure 3. The trains are coming and leaving, containers are loaded, unloaded, or stored in larger storage bays as shown in Figure 3, and finally, the customers pick up or drop the containers.



Figure 3: Praha-Uhřetěves intermodal terminal.

Thanks to the relative ease of moving a container from one train to the other, the trains do not have to run directly from the source to the final destination. Thus, the containers are often brought to a terminal where a crane switches the train hauling them. Such an option allows for multiple routes a container can take through a network, and great possibilities for optimization arise. As often is the case in an ideal scenario, one would optimize the system as a whole; however, such an approach is computationally intractable as it would require unattainable computational resources. In the optimization problem, several larger

subproblems arise, such as optimally storing the containers in the container bays so that the crane has to make a minimal number of moves in the future.

Another interesting problem, which is tackled in this thesis, is the assignment of containers to individual trains and the selection of railway routes the operator will purchase from the infrastructure operator to haul the containers. The specific goal of the thesis is to provide a precise formulation of the assigned problem and further solution using optimal methods. Naturally, *Integer Linear Programming* (ILP) and *Constraint Programming* (CP) were chosen as well-known representatives of the optimal methods, since the problem is NP-hard as demonstrated in Section 4.1. Then, the thesis aspires to compare the results of the two approaches, where the model created using CP will aim to utilize more specialized constraints provided by the CP solvers to possibly enhance the performance.

To better introduce the studied problem this, chapter first provides a brief introduction to the railway transportation 1.1. Then, the METRANS company providing the data and practical background is introduced in Section 1.2, where the Subsection 1.2.1 outlines the current operation of the company. Finally, the Section 1.3 describes the structure of the thesis.

■ 1.1 On Railway Transportation

Due to the physical features of railway tracks, which prevent trains from overtaking each other, the railway has to be operated differently than the road network. Thus, a brief overview of terms most commonly used in the railway industry is provided below.

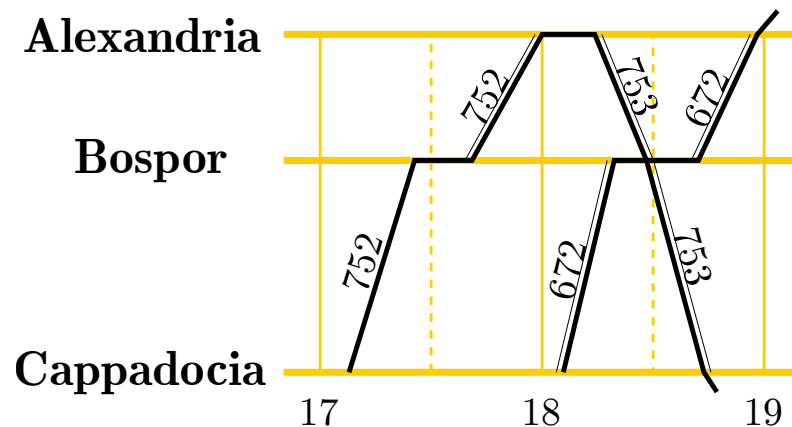


Figure 4: A grafikon is a possible display method of a timetable used on a particular railway line. The time is shown on the horizontal axis, and the distance covered is located on the vertical axis with marks depicting station names. The diagonal lines represent the trains with their numbers moving between the stations. Where the train continues outside of the area observed by the grafikon, the line representing the train continues outside of the visualization. Note that a line with a single track is modeled so that the trains can meet only at the stations, such as in Bospor at 18:30.

- **Grafikon** - due to the limitations described above, rail transportation must be carefully planned in advance. Such is in contrast to the road network, where vehicles can be ridden on the road freely except for traffic jams. That is why the railway's timetables precisely state which train operator can use the track at a given time. The timetables can be

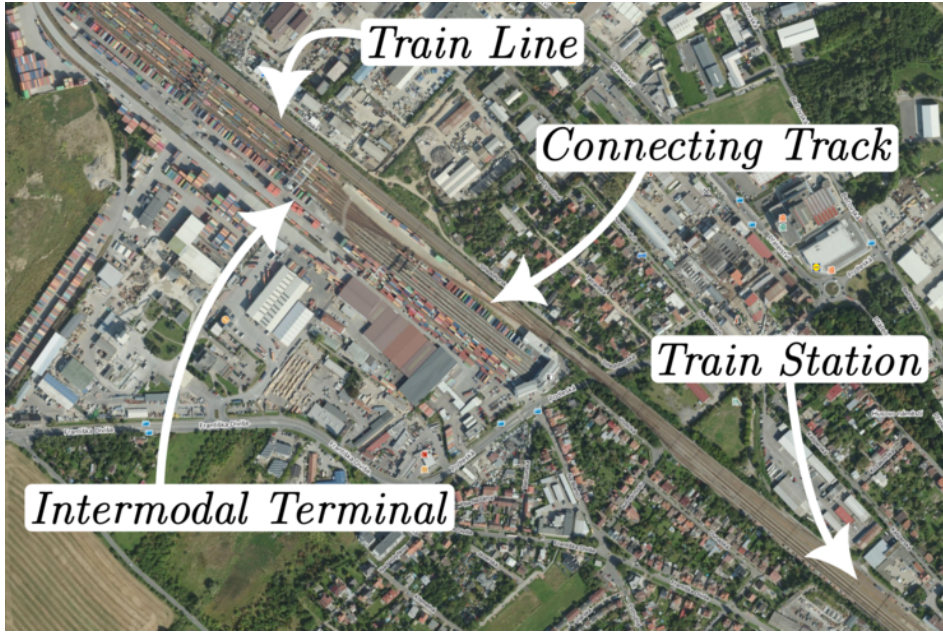


Figure 5: Overview of a typical railway station and an intermodal terminal.

expressed in a usual tabular form, similar to the one shown at the bus stops. Another option is a graphical form shown in Figure 4 called a **grafikon**. Such representation gives a better overview of the situation on the railway line as the person can observe the state at multiple stations at once. The grafikons even allow for better planning in the case of unexpected events, where the grafikon shows where the timetable provides some space to move the delayed trains closer to the destination.

- **Train slot** - a single diagonal line in the Figure 4 represents a single **train slot**. Train operators can purchase the slots from the infrastructure owner, and the slots grant permission to run a train on a route at a given time. The train slots must be purchased in advance as the creation of the timetables has to consider maintenance, upgrades, and expansion of the railway network. There is a limited supply of train slots, and to the author's best knowledge, the main lines in the Czech Republic are running to their full capacity. Thus, it is beneficial to utilize the slots to their full potential. Empty slots can be sold, which signals the need for a timely indication of such an event.

Even though the slots come with a time frame for operation, cargo trains often arrive at their destinations with delays. The higher priority of passenger trains often causes delays, as reliable arrival times are more important for the passengers. The usage of a particular rail line is conditioned by compatibility with the requirements for the achievable speed of the train and maximal load on an axle. As a result, the train slot holds such constraints in addition to its schedule.

- **Infrastructure owner** - the entity owning the railway tracks and stations. As running a railway network is not as profitable as in the past or creates unwanted monopolies, the infrastructure owner is often a state-owned company. The infrastructure owner is responsible for the maintenance of the tracks and stations and the construction of new tracks. An example of an infrastructure owner is *Správa železnic* in the Czech Republic. In most European states, the infrastructure owner is separated from the train operating

companies as the inverse would skew the competition.

- **Train operating company** - purchases train slots from the infrastructure owner to operate the leased or owned trains on the network. Most companies offering passenger service receive some form of subsidy from the state. In contrast, cargo trains can support their companies on a commercial basis. METRANS, *ČD Cargo* and *DB Schenker* are examples of cargo train operators, whereas *ZSSK* and *RegioJet* provide passenger service.
- **Rolling stock** - of a typical logistic company operating on railways simply consists of carriages presented in Figure 2 and suitable locomotives. The companies benefit from interoperable locomotives, as frequent switches would significantly slow down transportation. Naturally, the train sets can not emerge from thin air, and thus, the operation of the trains has to be planned so that the trains follow continuous paths. However, the trade is not always balanced; thus, the empty trains sometimes have to be moved to the location where the cargo is present.
- **Intermodal terminal** - is the main place for container handling. The intermodal terminals are usually connected to a railway station, as is shown in Figure 5. The terminal owner owns the tracks inside the terminal, often in contrast to the station. The trains coming into the station on the line can then access the terminal by the connecting track. However, the movement of the train between the terminal and the station operates under different rules than the operation on the main train line. Thus, the trains do not access all terminals passed along their routes due to the slowdown caused by the movement to and from the terminal, and they are most often run as a direct link between two terminals.

As the name suggests, the intermodal terminals support cargo exchange between multiple modes of transport. The exchange of cargo is achieved using cranes of varying sizes and functionalities. METRANS operates terminals predominantly in the mainland, while private subjects run the harbor terminals. The ownership then dictates procedures for handling the container trains. However, this thesis assumes complete freedom of manipulation at all terminals.

- **Container** - is a standardized unit for the transportation of goods. Some of the intermodal operators provide storage of the empty containers for the owners. Then, the container owner can request the container to be moved from the storage silo to the loading location. The denser coverage of the storage silos allows for shorter times upon requests for the empty container in comparison with the containers being stored in the ports of the vessel operators. The distribution of empty containers is another interesting problem that needs to be optimized. However, this thesis will focus mainly on the routing of the containers.

■ 1.2 METRANS

The METRANS company provided valuable insight into the problem, accompanied by real-world data. The Czech company was founded in 1948 [4], and since its creation, the main focus has been international trade and maritime transport. After the Velvet Revolution, the company mainly transitioned to container transport.

METRANS mainly operates container trains from the most critical European harbors, connecting them to the central and eastern mainland. Thus, METRANS is the middleman between the large open sea maritime operators and the end customer operating from the

1.2.1 Current Operation

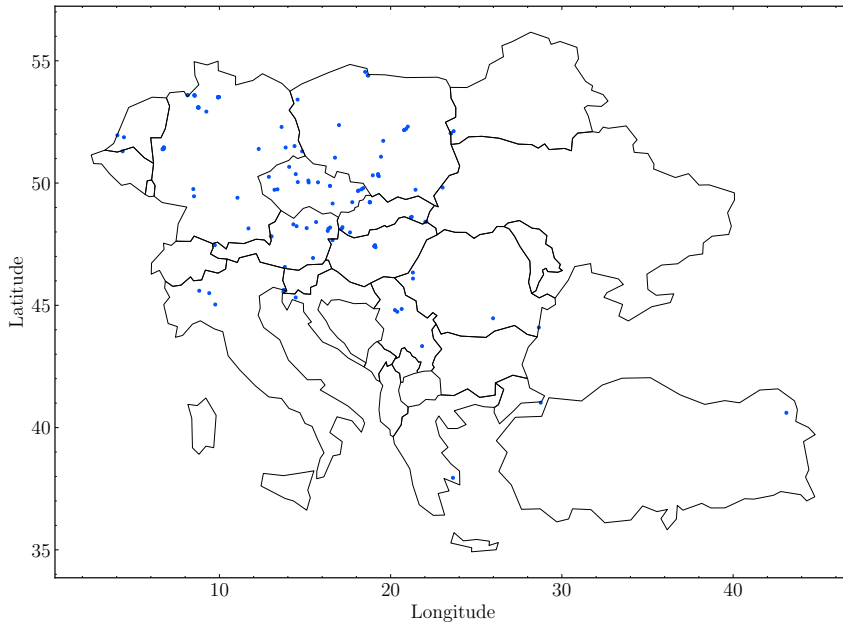


Figure 6: The locations of container terminals accessed by METRANS [1].

inland. Even the trucks can be ordered by METRANS to move the goods, but the bulk of the cargo is transported on rail. Yearly, METRANS transports around a million intermodal containers. The containers are exchanged on over 25 METRANS terminals, acting as buffers of empty containers for the maritime operators. All terminals and hubs accessed by METRANS are depicted in Figure 6. The cargo volume already allows for optimizations of container flow through the network created by the METRANS carriages. Thus, METRANS desired to investigate the possible optimization of the aforementioned container routing problem.

1.2.1 Current Operation

To understand the possible areas of improvement, it is useful first to study the current operation of the METRANS company and their solution to the container routing problem. Nowadays, each terminal with significant traffic has a dispatcher responsible for train scheduling. The dispatcher uses the METRANS's info system to plan the trains and assign the containers to them. Then, a dispatcher in the next hub terminal decides on the next steps. The process involves a lot of verbal communication between the dispatchers. Furthermore, the scope of a dispatcher is mostly limited to a single terminal, and such an approach could be prone to producing globally suboptimal decisions. Suboptimal decisions can be influenced by the inflexibility of interpersonal communication and the unwillingness to change the already prepared solution. The inflexibility is something which could be partially or completely eradicated if planned automatically.

1.3 Thesis Structure

The rest of the thesis is structured as follows: First, the foundations of exact solvers and the best representatives are introduced in Chapter 2. Chapter 3 starts with a review of commercial ERP systems that might attempt to solve a similar problem and continues with

related work covered in the academic field. The container routing problem is formally defined and accompanied by proof of its NP complexity in Chapter 4. Both the ILP and CP models are presented in Chapter 5. The achieved results are described in Chapter 6, and the thesis is concluded in Chapter 7 with a discussion of possible future work.

1.3 Thesis Structure

Chapter 2

Mathematical Programming

The term programming in this thesis is mainly used in context of formulating optimization problems where the constraints are specified to provide instructions to a specialized solver. These paradigms called mathematical programming allow for abstraction from the solving process and a standard way of describing the optimization problems are the specification of the constraints and the criterial function, where a solver is a program able to solve the given mathematical programs. Another approach for solving NP-hard optimization problems are heuristics which employ expert knowledge and simple rules to provide a solution without a guarantee of optimality. Such custom-tailored and fine-tuned heuristics to a specific domain can provide better results than mathematical programs in combination with a solver. However, developing such heuristics is time-consuming and requires a deep understanding of the problem, which is almost entirely mitigated by the utilization of the solver in combination with the mathematical program. The creation of optimization problem formulations (their variables, constraints and criterion...) and their usage are studied in the subsequent sections about Integer Linear Programming (ILP) 2.1 and Constraint Programming (CP) 2.2. Section 2.3 then compares the two paradigms and outlines their differences, while Section 2.4 explores the most popular solvers to help decide which one to use. Note that our understanding of ILP is often called *Mixed Integer Linear Programming* in the literature, and are often used interchangeably. So in this thesis, the authors decided to utilize the term ILP.

2.1 Integer Linear Programming

Following section studies the functionality of ILP pradigm, which helps to understand the capabilities of the formalism. The main progress in the field was achieved thanks to the need of solving problems in the realm of logistics, economics, and scheduling [5]. Such problems emerged during the Second World War, and the first computers such as ENIAC [6] were used to calculate the solutions soon after the end of the war. First, *Linear Programming* (LP) is defined, then LP is further restricted to integer domains of variables to form the ILP. The complexity class of the two approaches is stated, and the cores of solvers are explained.

The LP is a mathematical method used to find the maximum or minimum of a linear function, which is subject to linear constraints. The general representation of the problem (also referred to as a model) is expressed as follows:

$$\begin{aligned} & \text{maximize} && c^T x, \\ & \text{subject to} && Ax \leq b, \\ & && x \in \mathbb{R}^n. \end{aligned} \tag{1}$$

The coefficients of the vector c creates the cost function together with the vector of the variables x . The linear constraints are formed using the coefficient matrix A and the vector b on the right hand side. Thanks to the constraints being linear, a closer examination suggests that the space to search for solutions is a convex polytope. Additionally, the criteria function is linear, which results in the optimum being located at the vertex or the edge of the convex space created by the constraints. A naïve approach would check all the vertices, but the number of the vertices grows exponentially with the number of constraints. Thus, usually the

2.1 Integer Linear Programming

simplex algorithm is used to find the optimum, which often needs polynomial time. However, its worst-case performance results in exponential time complexity. Finding pivot criteria with strictly polynomial time complexity is an open problem [7]. Algorithms with polynomial time complexity exist, such as the ellipsoid method [8]. However, the primary usage of the ellipsoid method is the proof of the LP's polynomial time complexity. Although the ellipsoid method possesses polynomial time complexity in the worst case, the simplex algorithm outperforms it in practice [9].

However, formalized problems often require some variables to describe a discrete decision, e.g., no one would buy half a car. Such decision variables can be expressed as integers $x \in \mathbb{Z}^n$, which makes an LP problem ILP. Restriction of the variables to finite sets like $x \in \{0, 1\}$ is also possible, as it is a mere constrained integer space. The solution space is restricted and is no longer a convex polytope as the variables are discrete points of the previously continuous space used in the LP. As a result of the restriction, the edges and vertices of the polytope are no longer entirely present in the domain of the variables. Thus, the optimal solution may lay on the interior of the polytope used for the LP. The integral domains of the variables result in even for the decision variant of the criteria function $\min(0)$ the ILP is NP-Complete, whereas LP belongs to the P-Complete family [8].

On the inside, the ILP solvers employ many optimization techniques and heuristics to reduce the search space for the solution. In the following list the main approaches are presented:

- **Enumeration** - the most straightforward path to the solution, where the solver would check all possible points in the domain to compare their objective values.
- **LP relaxation** - as stated above, the simplex algorithm provides a fast solution to the LP problem. The natural initial idea is to relax the ILP problem to the LP by moving the domains of the variables to the real space \mathbb{R}^n . Then, the solution to the LP problem is used to gain an information about the location of the ILP's optimal value. Nevertheless, such a solution can be arbitrarily far from the optimum, thanks to the domain's discrete nature and the criteria function's angle.
- **Branch & Bound** - algorithm separates the domains of the variables into subproblems. The subproblems (branches) are then examined and solved separately. A solution to a branch then modifies the bound of the solution. A lower bound is set in the case of maximization as the best already achieved solution, which has to be improved upon by all new solutions, while the upper bound is set for minimization problems. The improvement over the pure enumeration method comes when some subproblem is not exhaustively explored due to the restriction imposed by the bounds [10]. However, the complex problems often result in exponential branching. Thus, a combination with another approach is desirable. Note, that there are extensions to the *Branch & Bound* algorithm such as *Branch & Cut* [11] or *Branch & Price* [12].
- **Cutting plane methods** - aim to utilize the LP relaxation to find the solution if the optimal solution is not integral, new constraints are added to remove such value and the process is repeated until the solution is integral [13]. The *cutting plane* is created so that it separates all domain's integer points and the non integral solution point. LP solvers then again retrieve a new solution from a relaxation. Generally, the addition of *cutting planes* is continued until the domain is constrained to a convex hull of the integer solutions. The method is frequently combined with the *Branch & bound* algorithm.
- **Conflict analysis** - is an additional approach, which borrows from the SAT solvers. The default behavior of the ILP solver would discard the infeasible branches, whereas the

conflict analysis transforms the infeasible branch into a new constraint, which is then added to the problem. Thus, the search space can be reduced and the computation time shortened [14].

Many of the presented techniques are implemented and described in the open-source solver SCIP [15].

■ 2.2 Constraint Programming

Constraint Programming (CP) is a mathematical programming paradigm, which, similarly to the ILP, uses constraints to define an optimization problem. However, there are differences between the two paradigms, which are outlined in the following section 2.3. An underlying problem of the CP is the *Constraint Satisfaction Problem* (CSP), where the most common techniques can be shown. The CSP is a problem where a finite set of constraints restricts a finite set of variables from finite domains. The objective is to find an assignment of the variables which satisfies all the constraints. The CSP can be changed to the constraint optimization problem (COP) by introducing a criterial function, which is then minimized or maximized in case of multiple possible assignments of the variables. Similarly to the ILP, the CSP is NP-hard as many NP-hard problems, such as graph coloring, can be easily expressed using the CSP.

The simplest approach to finding a solution to a CSP problem is to use some randomized method, where an initial assignment of the variables is found. Then, the assignment is improved to satisfy the constraints. The improvements often follow a scheme such as hill-climbing [16].

A systematic search for a solution utilizes backtracking. Similarly to ILP, the backtracking algorithm maintains information about bounds-domain restrictions in the case of CSP to prune some branches using back-jumping. It is beneficial to check the constraints earlier than in the leaf of the search tree as the earlier detection of the infeasibility can help to prune the search space again. Additionally, arc-consistency techniques are used to propagate the constraints through the domains of the variables. One of the most used arc-consistency techniques is the AC-3 algorithm [17], a simple yet efficient algorithm in practice. Some improvements upon the AC-3, like AC-4, achieve optimal worst case time-complexity but the average case is worse than the AC-3.

■ 2.3 Differences between ILP and CP

The constraints of the CP programs are more expressive than the ILP ones. Many binary decision variables must be employed to achieve a similar result in the case of ILP. Handling precedence of tasks could be taken as an example, most of the CP solvers offer a constraint for such occasion, but the decision variables have to be used in the ILP model. A feature missing in CP solvers is *LP relaxation*, which is employed in combination with *cutting plane methods*, whereas the CP solver relies more on inference of new rules and reduction of the variable domains. A typical example of a problem suitable for CP is sudoku, n-queens problem, or scheduling. On the other hand, ILP can more effectively solve complex criteria functions, which are influenced by many variables.

The problem of container routing includes a mix of both worlds. Both a schedule and minimization of non-trivial criteria function are required. Thus, the thesis aims to compare multiple modeling approaches.

2.4 Solver Wars

The world of ILP and CP solvers provides many user choices. Each option has strengths and weaknesses; thus, a short overview of the most popular and considered solvers is presented. Note that the commercial solvers outperform the open-source ones in most cases. However, it is hard to specify the exact factor of improvements. The problems presented to the solvers often vastly differ; one problem might be solved easily by one solver, while the same solver can struggle with another set of problems. Nevertheless, the most prominent solvers, such as **CPLEX** and **Gurobi**, are maintained by more organized teams with more time at hand. The larger knowledge pool helps to fine-tune the solver better on the varying problems, which results in better results on most issues. Furthermore, commercial solvers can utilize the advances made in the academic field presented by the open-source community. The openness is in stark contrast to the commercial solvers, where only occasionally, the maintainers switch teams due to some changes in the company’s structure and rarely publicize their best achievements. Next, a short overview of the most popular solvers is presented in Table 1.

Name	CP	ILP	Open-source
CPLEX	✓	✓	✗
Gurobi	✗	✓	✗
OR-Tools	✓	✓	✓
SCIP	✗	✓	✓

Table 1: Feature overview of the most popular solvers.

Between the commercial solvers, **Gurobi** slightly outperforms **CPLEX**. **Gurobi** is a newer solver written with multi-threading operation in mind [18]. Recently, Google started to develop its solver, **OR-Tools**, which is open-source. The **SCIP** is an older solver, which is still maintained and mainly provides a good starting point for the new researchers.

An interesting comparison can be made with the machine learning (ML) domain as ML is another domain using a description of a model which solves challenging problems. The solver’s landscape differs from the ML domain, where libraries such as **TensorFlow** and **PyTorch** are released under open-source licenses. The ML frameworks may cause the open-source availability scale to be better with the presented data, and the complexity of tackled problems does not vary that vastly. According to the author, the ML realm might slowly enter a state similar to that of optimization solvers. Currently, the models require more and more data, computational power, and raw electrical energy to train [19]. The training costs are rising, allowing just the biggest companies to train the largest models. Thus, it may be useful to try to borrow some novel ideas from the ML domain.

As there are many solvers available, a unification of the solver interfaces is provided by **GAMS** [20] and **MiniZinc** [21], which allows the programmer to utilize solver-agnostic interfaces. Thus, using a solver-agnostic interface prevents a lock-in effect when a single solver is chosen to model a problem. **GAMS** specializes on the ILP solvers, whereas **MiniZinc** is an attempt to create a common interface mostly for CP solvers. The authors of **GAMS** claim to be more performant than other solvers [22]. However, the performance of solver-agnostic methods is often hindered by the API exposed to them. The developers of the solvers choose not to provide the complete API, or the modeling features differ too vastly. The feature difference prevents the solver-agnostic approaches from utilizing the most optimal expressions. But, all in all, the solver agnostic approach allows a reasonable initial estimate of the most optimal solver for the problem.

Thanks to the commercial license available for university students, **CPLEX** was selected

as the main solver for the container routing problem. Additionally, **CPLEX** provides both ILP and CP interface, which eases the switch between them thanks to similar API and provide more fair comparison of the two approaches.

Chapter 3

Related Work

To provide the necessary background for formulating optimization approaches to the problem, a concise literature and software review is provided. A search for approach dealing with a similar problem is conducted to ensure the uniqueness of the presented work. As the whole problem of container routing concerns large businesses, an analysis of a sample of *Enterprise Resource Planning* software (ERP) is presented in Section 3.1. Then, a literature survey in Section 3.2 is created to examine the most up-to-date results from the academic field.

3.1 Enterprise Resource Planning Software - ERP

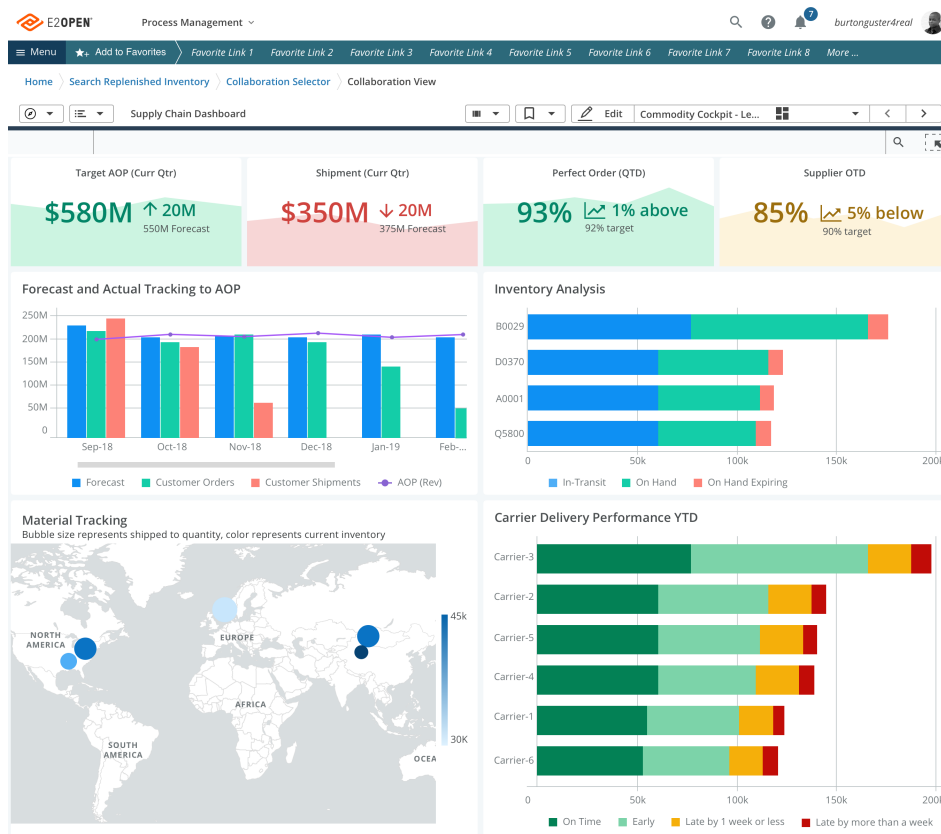


Figure 7: Illustration of e2open software, which provides an overview over the logistical chain.

As the name suggests, such software packages are mainly used by large companies. The implementation of ERP into company processes should accumulate all data collected by the company in a centralized storage. Among others, the accumulation of collected data allows for the detection of operation bottlenecks and the prediction of the estimated time of arrival. Furthermore, ERP software helps to manage inventory, receive offers, and even control the fleet of the operated vehicles. All those features make ERP software large and expensive software projects. The companies creating them guard their know-how quite well, and obtaining an

operational demo is a delicacy. Thus, mainly the presentation leaflets were reviewed, and a summary of the most interesting features is presented, while ensuring that no existing tool solves our specific problem:

Manhattan Associates [23] - provide prediction of trends in trade volumes. Thanks to the collected data, the software allows for creating and testing multiple scenarios. External weather forecasts improve the prediction of the impact of weather. Finally, the whole package includes a module tracking the overall happiness of the drivers and workers. On the other hand, the software does not help with the selection of optimization criteria, nor does it help with the optimization of movement between different legs of the transport journey. Lastly, rail transportation is not considered.

e2open (BlueJay) [24] - another software package without the support of rail transportation, enables prediction of the estimated time of arrival and improves the propagation of information through the supply chain. However, the software does not state the usage of an approach shortening the created cargo path.

INFOR [25] - includes a multi-platform, multi-leg logistics solution and allows for easy third-party logistics integration. Predicts the delivery cost and optimizes the cargo route on sea, road, and air, with rail transportation unfortunately omitted.

Overall, the reviewed software was mainly developed in the USA, and the available materials do not inform about the usage of railway transportation. The lack of railway solutions in commercial software calls for optimizations in the railway realm, such as route optimization, manipulation improvements, and load-balancing. Some of which are handled in the presented work.

■ 3.2 Optimization Approaches

The railway network is a fixed infrastructure with a limited capacity [26]. Since the end of the 19th century, the main lines in central Europe have been built, and just a small additions were created. The slowdown in railway infrastructure development can be attributed to the more concerned inhabitants as railway lines irreversibly change their surroundings. Another contributing factor is the relative denseness of the whole network, which is complete. However, the prevailing trade directions are running to their maximum capacity. Although the engineers of the past built the railways to the best of their knowledge, the current times demand for faster connections with more capacity. The speed increase is often achieved by costly infrastructure projects involving many bridges and tunnels, resulting in straighter lines required to run faster trains. The capacity of the connections can be boosted either by the increase in the number of tracks or by the rise in the number of trains running on the tracks. Lastly, optimizing the utilization of the trains inflates the capacity of the network for lower prices in comparison with newly built tracks. Although authors in [26] study mainly problems concerning passenger transportation, the freight transport issues are similar. Even slightly unrelated problems will be covered to understand the aspects of freight transportation omitted in this thesis.

■ 3.2.1 Possibilities of Optimization on the Railway Networks

As already stated before, the infrastructure manager and the train operators are separate entities. Such separation is found in most EU countries where METRANS operates. The infrastructure manager first gathers the requests for the capacity of the line. Then, the manager's internal optimization tool allocates the capacity to the train operators. In [27],

an ILP model is used to optimize the timetables in Switzerland. The test data belonged to a single Swiss canton, and the optimization tool was able to track passenger satisfaction. Additionally, authors in [28] formulate the timetabling problem as a Job-Shop Scheduling Problem and solve it using CP. They show that the CP is a viable alternative to the ILP, suggesting that even the presented problem could benefit from testing multiple approaches.

The employment of timetabling optimization is showcased in [29], which aims to create better timetables for the Berlin's underground. The main focus was laid on saving a train in operation and reducing the time needed when transferring between individual lines. This usage highlights the main problems encountered when implementing such an optimization approach. In the case of Berlin's underground, the optimization was even able to shed light on the conflicting goals of the upper management which included saving a train, while maintaining the current quality of operation. The container routing problem presented in this thesis could borrow the goals from the underground's optimization, such that the aim would be to minimize the time the containers spend in the terminals.

It should be noted that the timetabling criterion favors the passenger trains over the freight ones. The passenger service often requires the trains to run at the same minute each hour of the day to ease the navigation in the train schedules for the passengers. On the other hand the freight trains do not depend on the exact time of arrival.

Many problems in the current world, such as online advertisement [30] and spectrum allocation [31] are allocated using auctions as such system helps to extract the most value for the provider. Thus, a study in [32] shows a possibility of using auctions for timetable creation. According to the authors, the cost can be reduced with the help of train bundling concerning their speeds. Nevertheless, the auction function is quite complex, and the authors proposed more cooperation with the people in the field prior to the real world application.

Another exciting problem of railway transport is rolling stock availability, where under the rolling stock are understood locomotives and carriages. The availability together with the constant repositioning due to the railways nature creates the rolling stock circulation problem. Running a single train type on a single line is trivial, but the problem starts getting complex if the transported amount fluctuates, as in the real world. The purchase of a new locomotive or carriage is a costly, long-term investment. It is then in the interest of the operator to use the rolling stock as much as possible or optimize the operation to be utilized to the fullest. A similar problem arises in the case of the carriages, where the number of carriages should be adjusted to the demand. Moving an empty carriage results in unwanted wear and tear, which often outweighs the benefits of having fixed train lengths. In the case of passenger trains, hauling empty carriages is undesired as the number of conductors has to be increased, or the locked carriages cause confusion at the train stations, which can result in delays. The study [33] solves the problem by combining heuristics with ILP and finetuning the solution by employing the genetic algorithm. The presented solution can determine the circulation of the rolling stock in the rapid transit in Seville. On top of the rolling stock collection, the maintenance facilities' location has also been decided.

Optimization of the circulation on railway lines is shown in [34]. The authors focus on train units, including train driver booths at both ends of the train and carriages in between. Such setup allows for faster connections of the trains, as the locomotive does not have to be moved to the other end of the train to add more carriages. The computation of the results is achieved by solving an ILP model, where the solver parameters were tweaked extensively, combined with other methods such as LP-relaxation and local search. Compared with the formerly used rolling stock circulation schedule, the new schedule saved approximately six percent of operation cost and is currently used to plan the real schedules.

Yet, most of the studies such as [35], [36], [34], and [33] focus mainly on passenger service

3.2.2 Freight Transportation Optimization

and rapid transit. To the author’s best knowledge, the focus may be caused by higher fluctuations in demand for passenger services or the higher expense of operating passenger trains. Finally, the lack of focus can originate in the less frequent possibilities for uncoupling, as it is desirable for freight trains to run longer distances. Moreover, the papers often focus on fixed train units supporting quick coupling and uncoupling. Such train units are not used to haul freight cargo. Even though the problem is out of the scope of this thesis, studying the rolling stock circulation could be interesting in the case of freight train operations.

Maintenance scheduling presented in [37] is a closely bound problem to the rolling stock circulation. According to [26], the maintenance on sparse networks should be planned as the distance to the maintenance depots is higher. Due to the closer proximity to the maintenance centers, ad-hoc planning is possible on denser networks. It is assumed that METTRANS offers maintenance in most terminals, which eases our load on the planning.

Lastly, the crew schedules are often optimized, as reducing the amount of crew decreases the operation cost. The deployment of a crew scheduling approach in [38] showcases a two percent decrease in the crew’s costs. The presented algorithm first generates a large set of possible duties and then selects the best ones using various heuristics. The possibility of swift rescheduling is highlighted as the operation has shown a less strict requirement on the buffer time to change the train. Implementing a similar crew scheduling approach into, or after, the container routing problem would be beneficial as METTRANS operates long routes. The long routes make the importance of good crew schedules more pronounced in contrast with short routes starting and ending in the same locomotive depot close to the residence of the train crew.

In reality, delays in the operations occur, which calls for frequent reschedulings, such as studied in [39]. The survey shows solutions that precompute possible reschedulings and propose an online rescheduling tool. Lastly, [40] mainly focuses on removing passengers from metro line stations after a service disruption. During the disruption, the number of passengers on the platform can rise, and then skipping a station with one train is often desirable as the operation is restored faster. A similar approach could also be employed for freight trains, as the containers might pile up at the terminals, and restoring regular service is desirable after a delay.

■ 3.2.2 Freight Transportation Optimization

Recent survey [41] concerning multi-modal freight transportation states that there are just a few research papers concerning freight rail transportation. It also identifies the main possibilities for improvement in the form of container manipulation and their assignment to the trains, which is also essential part of our optimization. In [42], the authors show valuable information for selecting a suitable criterion to model the intermodal routing problem. The authors show which aspect to consider such as the penalty for idle time, distance and time traveled. The transport of fresh fish from Norway to continental Europe is an example; thus, some of the results might be slightly problem-specific.

Authors in [43] optimize the layout of the cargo loaded on the intermodal trains and the loading process by using the Hungarian algorithm or successive shortest paths. They conclude that the train’s brakes benefit from the heavier cargo being closer to the locomotive, resulting in lesser wear and tear. The results of the study point to the fact that the loading criteria could take into account many requirements, and thus, it is necessary to select them carefully.

The problem of finding a new route for cargo in the US is presented in [44], where the authors formulate the task as flow maximization, which can be solved using the Ford-Fulkerson algorithm. Their result is a proposal for adding a new railway track to increase the capacity of the whole network through the US. Such a result shows that relaxing the problem of individual

container routing to the max flow problem would yield a solution in a polynomial time. Thus, as the problem of adding a track could be similar to solving a container assignment, the Ford-Fulkerson algorithm could be used as the initial heuristic solution.

Intermodal container routing problem is studied in [45]. The authors describe the problem using many constraints to express all intermodal transportation issues. The main goal is to minimize the cost of the transportation plan so that the approach can create plans for both long-range and short deliveries. Large neighborhood search is utilized as the initial heuristic, but the authors do not attempt to solve the problem optimally. A similar approach could be used in the future work to accelerate the exact models.

The solution to the best route selection, using Lagrange relaxation and Multi-modal multicommodity flow problem with time windows, is created in [46]. As a heuristic, the shortest path of a package was utilized. When a route reaches its maximum capacity, a re-optimizer is called to find a feasible solution. The approach does not try to optimize on a per-container basis, thus enabling this thesis to achieve better results.

Authors in [47] mainly focus on a similar problem as this study, with road and rail transportation. In the article, more focus is put on the uncertainties of transporting and handling hazardous material. Similarly, to [46], shortest paths are used as the heuristic to provide an initial guess at a solution. The study also shows the added work when considering the hazardous material, which is not considered by this thesis. The solution is then provided using tabu search, which selects the best routes, but the authors do not attempt to solve the problem using optimal methods. Another aspect of the problem not considered in the model are the container sizes, which are introduced in this thesis.

In [48], the influence of requirements for green logistics is added. The problem is modeled in fuzzy set theory, which helps create the uncertainties arising in real-world scenarios. The solution is then obtained using a LINGO solver. The authors conclude that the requirements for green logistics and maximization of transported freight are often contradictory. Nevertheless, optimizing the container routing is desirable as it aids in extracting the most out of the trains that are already operated, helping to achieve our generation's green goals.

The literature overview shows that the presented approaches mainly use heuristics with initial solutions provided by the shortest paths. The attempt to solve the problem using optimal methods is often omitted. Hence, the presented approach aims to fill this void by formulating the models and utilizing the optimal methods and to the author's best knowledge, no such approach has been presented in the literature.

3.2 Freight Transportation Optimization

Chapter 4

Problem Statement

This chapter first introduces the reader to the list of used symbols in Table 2. Then, the requirements for the solutions are outlined to lay the foundation for the method and lastly the NP-completeness of the problem is proven in Section 4.1.

Symbol	Explanation	
G	all possible train slots	} Train slots
g	a particular train slot	
ψ_g	time of departure of a train slot	
ϕ_g	time of arrival of a train slot	
l_g^{max}	maximal load of a train in TEU	
l	the distance between the stations of a train slot	
α_g	first station of the train slot	
ω_g	last station of the train slot	} Stations
S	all stations used by the trains	
s	station from the set of all stations	
C	all containers	} Containers
c	container	
τ	size of the container in TEU	
ψ_c	release time of a container	
ϕ_c	deadline of a container	
α_c	the station where the container is released	
ω_c	the destination of a container	} Penalties
p_t	fixed price for owning a train and the carriages	
p_d	price per km covered	
p_m	price of moving a container between trains	
p_i	penalty for having more trains in one direction than in the other	

Table 2: Overview of the used symbols for better orientation in the proposed models.

The datasets and requirements of the METRANS company leads to the following problem formulation. The $s \in S$ are all the intermodal terminals accessible by the METRANS trains. The network to plan on is then created by the train slots $g \in G$. Each train slot g states that the railway transport operator is granted permission by the infrastructure owner to operate a train between the stations α_g and ω_g . The departure and arrival times ψ_g and ϕ_g are specified for each train slot. In reality, delays often occur, and thus, the actual departure and arrival times can differ from the planned ones. However, this thesis chose to ignore the delays and assume the trains are always on time, as modeling the delays would require a more complex model. The train slot g specifies maximal load l_g^{max} , which is expressed in TEU. The unit TEU stands for *twenty-foot equivalent unit*, a usual container size. The studied dataset involves containers of 1 and 2 TEU sizes. Additionally, the train slot could be described by the minimal speed the train has to be able to achieve in order not to slow down the other trains in the network and maximal allowed weight of the whole train. It is assumed all trains

4. Problem Statement

can achieve the minimal speed and no combination of containers will overweight the trains. Finding the optimal assignment of locomotives and carriages would further complicate the problem, and this thesis will focus mainly on the assignment of the containers to the train slots.

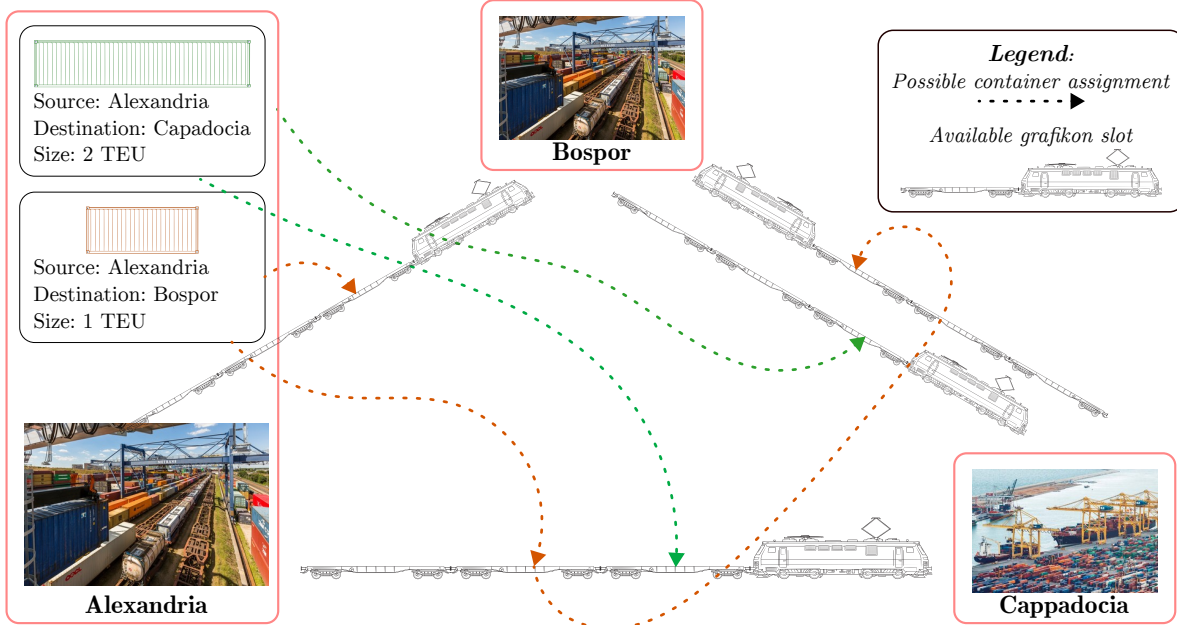


Figure 8: An overview of a simple instance of the container routing problem. There are two containers to be transported and four train slots to be chosen from. The possible assignments of the containers to the trains representing the train slots are depicted using the dashed lines. The optimal solution will be based on the prices of individual train slots and the price of container manipulation.

The terminals could be constrained using parameters describing the number of tracks and the storage space for the containers. However, in the first iteration of presented problem, a relaxed version is solved, assuming all terminals can accommodate all incoming trains and containers.

The containers C are the entity that has to flow through the network and generates the profit for the company. Each container $c \in C$ comes with its size τ_c , release time ψ_c with starting location α_c and deadline ϕ_c with final destination ω_c . The container must be delivered to the final destination before the deadline. Some other properties could, among others, include the weight and presence of hazardous cargo. But upon agreement with METTRANS, those properties and constraints are omitted from the herein presented problem formulation.

For the creation of optimization criteria, the following penalties are specified: cost of running the trains with the carriages p_t , cost of moving the container in the terminal from one train to the other p_m , and cost of the distance covered by in a train slot p_d and penalty for imbalanced train usage p_i . In reality the price of manipulation varies between the terminals operated by METTRANS and the ones controlled by a different company. Thus, the manipulation in own terminals should be preferred, but the initial solution assumes all terminals price the manipulation similarly. The penalty for imbalanced train usage is employed when a train, meaning a locomotive, is not present or not used further after its required utilization by a train slot, which penalizes inconsistencies in the transport network. The usage of such a penalty will be determined in the future work. The problem statement presents a relaxed but

still quite descriptive model of the real-world scenario and was agreed upon by METTRANS.

4.1 NP Completeness

A proof of NP-completeness for the task is presented to ensure there is no reason to try to look for polynomial algorithms to solve the problem because finding a polynomial algorithm would suggest the groups of P and NP tasks are the same. A polynomial reduction from a known NP-hard problem to the herein-presented container routing problem is presented to prove the studied problem is NP-hard. Thus, the thesis will focus on solvers and methods primarily suited for NP-hard problems.

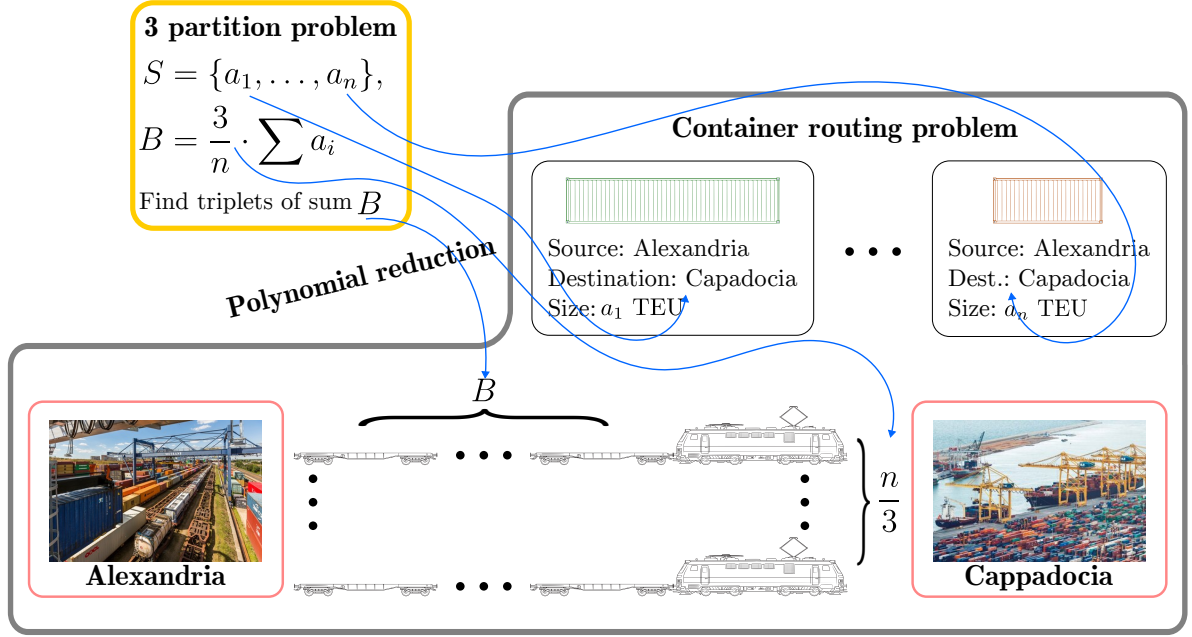


Figure 9: Reduction of the 3-partition problem to the container routing problem.

4.1.1 Reduction from NP-Complete Problem

The 3-partition problem is a well-known NP-complete problem, which is defined as follows:

Definition 1. (3-partition problem) Given a set of integers $S = \{a_1, a_2, \dots, a_n\}$ and a number B , the task is to determine if S can be partitioned into $n/3$ subsets, each with a sum of B .

The polynomial reduction is visualized in Figure 9. An instance of a container routing problem with $|C| = n$ containers and $|G| = n/3$ train slots is created. The reduction is achieved by mapping the bins of size B in the 3-partition problem to train slots between the same city, with the capacity of each slot being B TEU. The integers a_i represent the size of the containers in TEU to be assigned to the train slots. If a solution to such an instance of this container routing problem is found, it solves the underlying NP-hard problem of 3-partition, which suggests the container routing problem is at least as hard as the 3-partition problem. The reduction is polynomial; thus, the container routing problem is NP-complete. While the presented datasets involve containers of only 1 or 2 TEU long, the two possible values are already enough to make the original 3-partition NP-Hard.

4.1 Reduction from NP-Complete Problem

Chapter 5

Method

This chapter presents an approach for solving the container routing problem presented in Chapter 4 as ILP in Section 5.1 and CP in Section 5.2. Furthermore, to prepare for a solution using more capable methods, an analysis of the real-world dataset provided by METRANS is presented in Section 5.3 to gain more intuition about the underlying problem. Section 5.4 describes the preprocessing pipeline to prepare smaller datasets for testing the presented methods. Lastly, Section 5.5 presents the quality measures used to evaluate the solution's performance, as it was determined that simply comparing the costs of the plans is not a sufficient quality measure.

5.1 ILP Formulation

The ILP model uses variables described in Table 3.

Symbol	Description
$x_{c,g} \in \{0, 1\}$	container c is transported using a train slot g
$y_g \in \{0, 1\}$	train slot g is used

Table 3: List of used variables. Note, that there is $|C| \cdot |G|$ variables $x_{c,g}$ and $|G|$ variables y_g .

Then the aforementioned problem specification in Chapter 4 allows to create the following ILP, where the criteria function is defined as follows:

$$\min \sum_{c \in C} \sum_{g \in G} x_{c,g} \cdot p_m - |C| \cdot p_m + \sum_{g \in G} y_g \cdot (p_t + p_d \cdot l_g). \quad (2)$$

The criteria function penalizes each manipulation with every container using the first term, where each used train slot results in an additional manipulation. Naturally, a usage of a single train slot should not result in an manipulation cost and hence those costs are offset by the second term. The costs of train slot g in the third term are broken down into fixed costs p_t , representing the cost of owning or sourcing the locomotive and variable costs p_d , based on the distance covered. Note, that the penalty for imbalanced train usage p_i is ignored as the inclusion would result in enlargement of the model.

5.1 ILP Formulation

The ILP formulation is as follows:

$$\text{subject to } \sum_{g \in F} x_{c,g} = 1 \quad \forall c \in C, F = \{g \in G : \alpha_g = \alpha_c\}, \quad (3)$$

$$\sum_{g \in F} x_{c,g} = 0 \quad \forall c \in C, F = \{g \in G : \omega_g = \alpha_c\}, \quad (4)$$

$$\sum_{g \in H} x_{c,g} = 1 \quad \forall c \in C, H = \{g \in G : \omega_g = \omega_c\}, \quad (5)$$

$$\sum_{g \in H} x_{c,g} = 0 \quad \forall c \in C, H = \{g \in G : \alpha_g = \omega_c\}, \quad (6)$$

$$\sum_{f \in F} x_{c,f} = \sum_{h \in H} x_{c,h} \quad \forall c \in C, \forall s \in S \setminus \{\alpha_c, \omega_c\},$$

$$F = \{g : \omega_g = s\}, H = \{g : \alpha_g = s\}, \quad (7)$$

$$\sum_{c \in C} x_{c,g} \cdot \tau_c \leq l_g^{max} \quad \forall g \in G, \quad (8)$$

$$\sum_{c \in C} x_{c,g} \leq y_g \cdot M \quad \forall g \in G, \quad (9)$$

$$\sum_{f \in F} x_{c,f} \cdot \phi_f(s) \leq \sum_{h \in H} x_{c,h} \cdot \psi_h(s) \quad \forall c \in C, \forall s \in S \setminus \{\alpha_c, \omega_c\},$$

$$F = \{g : \omega_g = s\}, H = \{g : \alpha_g = s\}, \quad (10)$$

$$\psi_c \leq \sum_{g \in G} x_{c,g} \cdot \psi_g(\alpha_c) \quad \forall c \in C, \quad (11)$$

$$\phi_c \geq \sum_{g \in G} x_{c,g} \cdot \phi_g(\omega_c) \quad \forall c \in C. \quad (12)$$

Constraint/s	Group
3, 4, 6, 5, 7	Flow constraints
10, 11, 12	Time constraints
8	Capacity constraint
9	Slot selection

Table 4: Grouping of the constraints present in the model.

Table 4 is presented for a better orientation in the various constraints in the model. The Equation 3 ensures there is a start to a path of the container through the network by selecting a train slot g beginning at the starting location of the container denoted α_c . Creation of loops in the final solution is prevented by Equation 4, which prohibits the container from entering its starting location α_c . Solutions with loops are undesirable despite their lower objective value, as the cyclic routing does not reflect the real problem. The loops would be achieved around the origin and destination without a connection between the two loops, which does not create a continuous path the container could move along from origin α_c to the destination ω_c . Similar actions are achieved by the Equations 5 and 6 for the destination ω_c of the container.

The continuous path through the transport network is created using Equation 7, where the entrance of a container into a terminal has to result in a departure from said terminal. Note that the Equation 7 does not hold for the starting α_c and final location ω_c of the container.

Additionally, a maximal number of containers on each train is restricted by Equation 8. Equation 9 determines for each train slot g if it was used in the optimal plan or not for the

computation of the criterion function. The constant M in Equation 9 helps avoid creating constraints for each container. Big M is chosen to be greater than the length in TEU of the longest train in the dataset.

The time continuity of train slots along the container’s path is ensured by Equation 10, which operates in a similar fashion as 7. Finally, the Equations 11 and 12 achieve the transport of the container within the time constraints in the form of release time and deadline. The constraint for the case of release time is as follows: just one train slot leaving the initial station should be selected. Thus, the sum over all train slots should result just in the departure time of the selected train slot.

■ 5.2 CP Formulation

Two approaches for creating a CP model are presented. Both models share some similarities with ILP model, yet the first model uses integer variables, while the second model uses interval variables.

■ 5.2.1 CP Model with Integer Variables

The ILP model is transferred to the CP while using the integer variables in CP. It is expected that the behavior of such a model will be the slowest of all created models, as the integer variables do not allow for the fast propagation of the constraints through the model.

■ 5.2.2 CP Model with Interval Variables

The second modeling approach aims to use the CP paradigm more by exploiting the **interval** variables. Each decision variable of the ILP model representing if a container c is assigned to a train slot g is replaced by an interval variable $x_{c,g}$. The **interval** variable is defined as follows:

$$x_{c,g} = \mathbf{interval}(\psi_g, \phi_g, \mathbf{optional}). \quad (13)$$

The length of the interval is forced to be equal to the duration of the train slot $\phi_g - \psi_g$. The optionality allows for the nonpresence, and the **interval** variable is bounded fixedly using the departure ψ and arrival ϕ of the train in a given train slot.

The flow constraints are shown for a single container c , but in the final model, similar constraints are created for all containers. A master **interval** variable $O_{c,s}$ selects an alternative between all the slots leaving the station s . Similarly, a master **interval** variable $I_{c,s}$ combines trains incoming to the station s

$$O_{c,s} = \mathbf{alternative}(\forall x_{c,g} : \{g \in G : \alpha_g = s\}, \mathbf{optional}), \quad (14)$$

$$I_{c,s} = \mathbf{alternative}(\forall x_{c,g} : \{g \in G : \omega_g = s\}, \mathbf{optional}). \quad (15)$$

The start of the container journey removes the **optional** argument from the master variable for alternative selection m_{c,α_c} as the container has to be forced into the network. Next, the compliance of the train slot with the release time ψ_c is ensured by the following constraints

$$\mathbf{start_of}(O_{c,\alpha_c}) \geq \psi_c, \quad (16)$$

$$\mathbf{presence_of}(I_{c,\alpha_c}) = 0. \quad (17)$$

The **presence_of** constraint prevents the creation of loops in the solution. Analogous constraints are created for the deadline of the container ϕ_c in the destination station ω_c :

$$\mathbf{end_of}(I_{c,\omega_c}) \leq \phi_c, \quad (18)$$

$$\mathbf{presence_of}(O_{c,\omega_c}) = 0. \quad (19)$$

The flow of the container and the subsequent timing is ensured by:

$$\text{presence_of}(I_{c,s}) = \text{presence_of}(O_{c,s}), \quad (20)$$

$$\text{end_before_start}(I_{c,s}, O_{c,s}). \quad (21)$$

Lastly, the constraints on the capacity of the trains have to be created. In contrast to the previous constraints, the capacity constraints are created for each single train slot g using a `cumul` variable, which the `pulse` function returns. The `cumul` variables provided by **CPLEX** allow us to model the capacity of a resource over time. The `interval` variables occupy parts of the capacity by using the

$$\text{train_length} = \sum_{c \in C} \text{pulse}(x_{c,g}, \tau_c), \quad (22)$$

where the τ is the size of the container c and the `train_length` has to be bounded by the capacity of the train l_g^{max}

$$l_g^{max} \geq \text{train_length} \cdot y_g, \quad (23)$$

where an integer variable y_g is used to determine if the train slot is used or not for the criteria computation.

The criteria function is then defined as:

$$\min \sum_{c \in C} \sum_{g \in G} \text{presence_of}(x_{c,g}) \cdot p_m - |C| \cdot p_m + \sum_{g \in G} y_g \cdot (p_t + p_d \cdot l_g). \quad (24)$$

to achieve the same results as the ILP model. The constraints are grouped in Table 5 to make comparing the two models easier.

ILP	CP	Description
3, 4, 6, 5	17, 6	Handling of origin and destination
7	20	Flow constraints
11, 12	16, 18	Releases and deadlines
10	21	Time continuity
8, 9	8	Capacity constraint and slot select

Table 5: Grouping of the similar constraints between the two modeling paradigms.

5.3 Dataset Analysis

For the creation of additional datasets or new batches of data, the provided dataset is observed for patterns and usual operation. The operation record is from January of 2023, presented in Figure 10b, where the first day of 2023 saw far fewer utilized trains than in the following weeks. During the month, 64 542 containers were moved by 1955 individual train connections. Figure 10a shows that the number of trains remains relatively constant during the weeks apart from the first month, hindered by the ending winter holiday season. However, the number of trains during each weekday is the highest on Thursdays before the weekdays and usually reaches the low point on Mondays as displayed in Figure 10b. The amount of trains arriving and departing into all terminals is shown in Figure 11. Most trains depart from the terminals during the late afternoon and evening, probably to arrive at their destinations in the early morning. Interestingly, the operation on the terminals reaches a low point around 7 in the

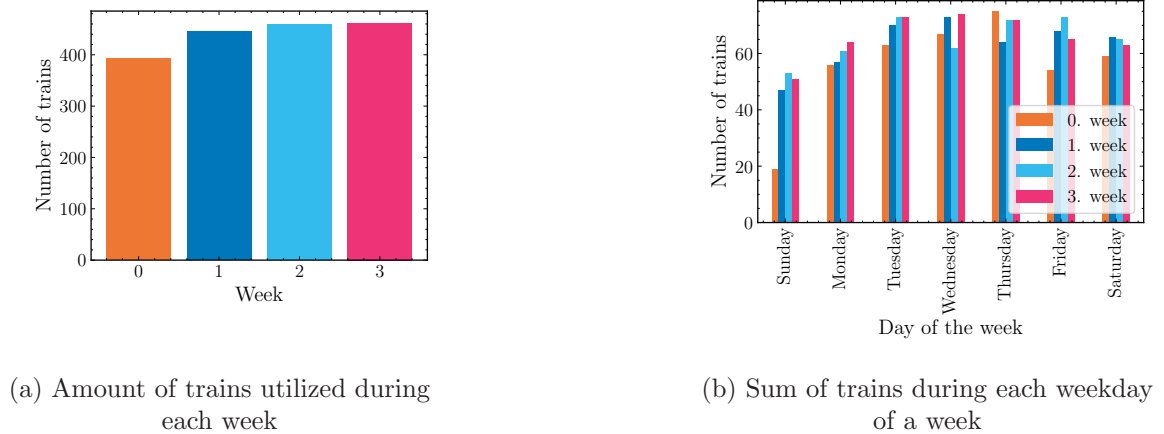


Figure 10: The number of trains operated each week and sorted according to weekdays.

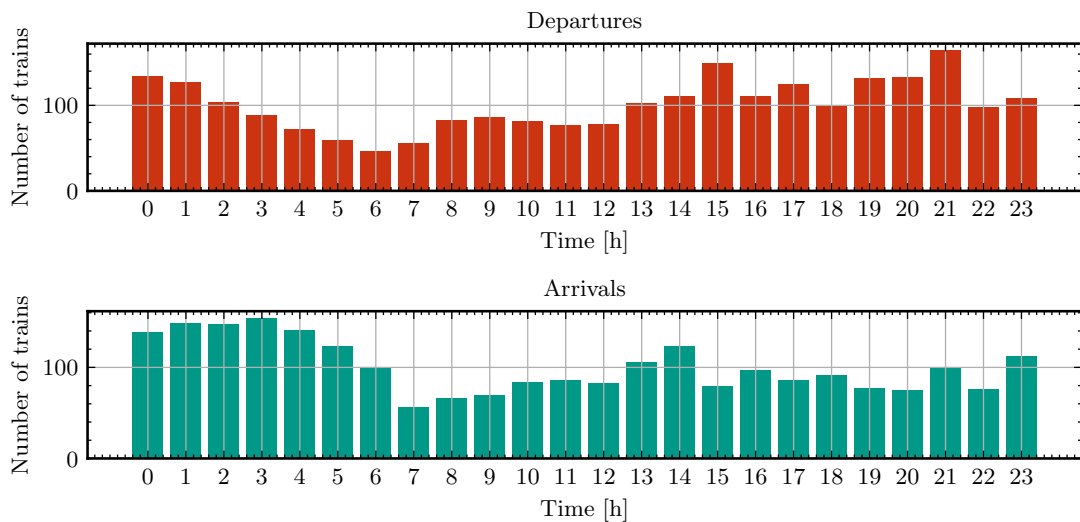


Figure 11: Trains arriving and departing combined from all stations during an individual hour.

morning. Such is most certainly caused by the morning rush hour connections occupying the railway capacity.

The Figure 12 shows the twenty most used terminals. Unsurprisingly, harbors hold many positions in the chart as METRANS specializes in container transport to and from terminals. The transportation network created by the trains operated by METRANS follows the main transportation corridors and connects the harbors with the inland terminals. A selection of the utilized train lines is visualized in Figure 15.

The dataset analysis was requested by METRANS to gain a closer understanding of the operation and can be further used to adjust the processes at the terminals.

5.4 Dataset Preprocessing

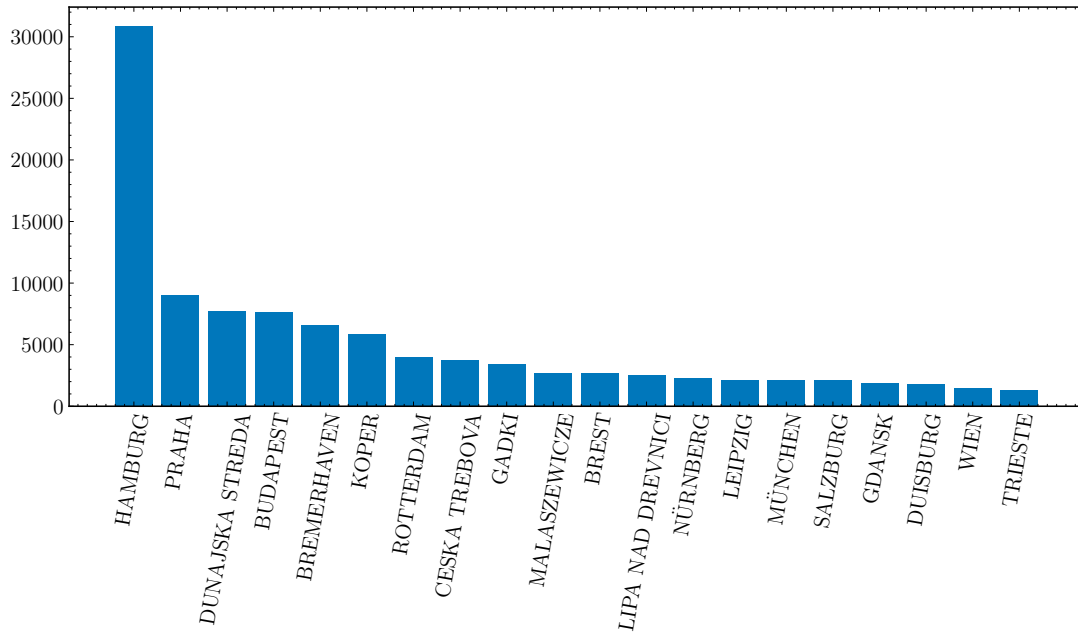


Figure 12: The number of containers handled by the most utilized terminals.

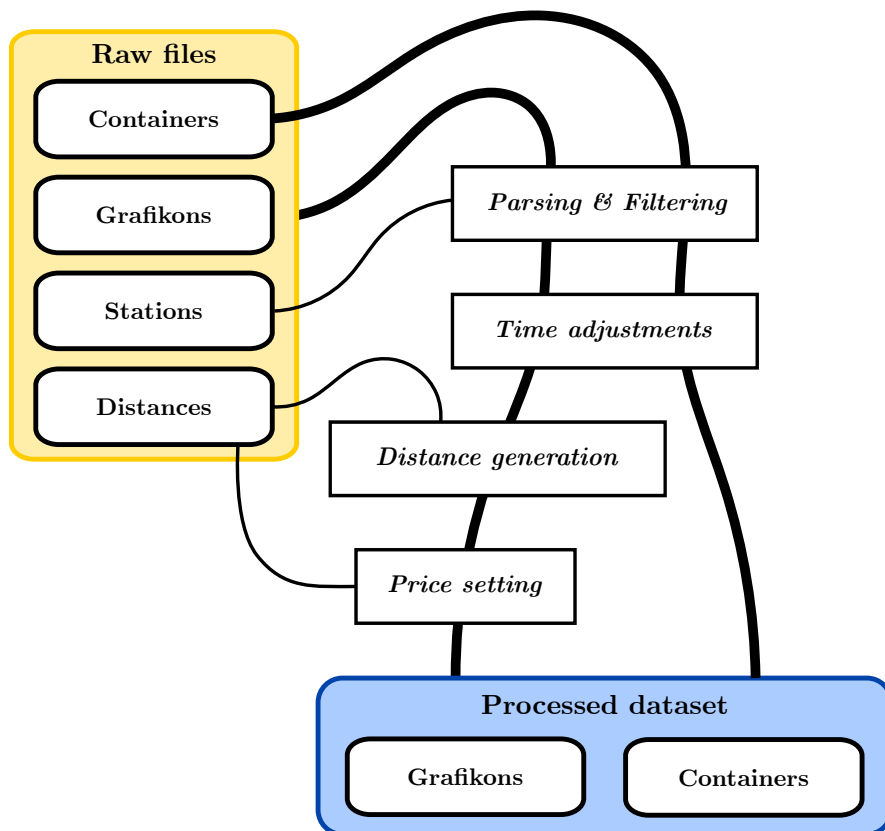


Figure 13: The overview of the dataset preprocessing pipeline.

■ 5.4 Dataset Preprocessing

The dataset provided by METRANS came in the form of `.csv` files, created as a database dump. The database is utilized as storage, and the data is not used to optimize processes directly. Thus, the raw files must be made consistent and transformed into a format suitable for optimization. Besides the consistency, creating smaller datasets for incremental testing of the methods is beneficial. The outline of the transformation needed for the dataset creation is depicted in Figure 13, and the steps achieve the following:

- **Parsing & Filtering** - the data from raw files has to be parsed. The grafikons come with a slightly more complicated way of describing their operation days, and thus, a *Finite state Machine* (FSM) is employed to retrieve data from the files. To successfully add a new train slot to the list of all slots, the FSM has to switch through states where it reads the parameters of the slot, the route, and the schedule of reaching the individual stations. Additionally, the routes' stations are compared with the list of terminals holding information about the terminal's adjacent stations. As the train slots do not have to cover all terminals, the containers provide connection to, the stations are filtered only to include the ones present in the grafikons. Moreover, an adjacency matrix is created using the train slots. The created matrix allows for checking if the containers can reach their supposed destination using a depth-first search. When a need arises for a smaller dataset, the containers are limited to a specific time range by keeping their release dates and deadlines constrained inside the time frame.

In reality, the operators are often presented with a backlog of containers as they need to adjust the containers' assignments frequently. Such is not true when restricting the dataset to a fixed time window. Initially, the amount released from the containers slowly rises before reaching a steady state. A buffering date can be set to mimic the actual behavior, which will move the release dates of containers to a future date, creating a backlog released at the start of planning horizons. In the process, the deadline is also modified to ensure the container can be transported.

- **Time adjustment** - in scheduling problems, the time is discretized to facilitate allocating tasks to resources. The presented approaches do not require the time to be discretized, but the date-time objects are transformed to integral values. The preprocessing steps set the earliest release time ψ_c of a container c as a zero time. Then, all-time data is adjusted to an integer value holding the number of minutes from the earliest release time.
- **Distance generation** - the distance matrix provided by the METRANS company is incomplete. Thus, it is necessary to generate the missing distances. To the author's best knowledge, there is no simple API or library to retrieve the railway distances using publicly available maps.

A manual option would be to use *bikerouter.de* [49], which allows to switch to railway planning. However, the routes created often do not correspond to the company's real-world routes as they follow the shortest paths. The shortest paths sometimes can not support the cargo trains' weight or lower speed. The variance in the trains and available lines results in more challenging planning than road networks or hiking trips.

Finally, the relation of the actual distance l between terminals and the shortest possible path l_s calculated using GPS coordinates is studied in Figure 14. A linear regression approximates the actual distance from the distance as the crow flies. The coefficients of

5.4 Dataset Preprocessing

the linear function were set to:

$$l = 1.29 \cdot l_s + 37.44. \quad (25)$$

Additionally, some noise could be added to the linearly approximated samples. Nevertheless, adding the noise is omitted as it would have little impact on the final criteria function.

Experimental evaluation was carried out using the *bikerouter.de* routing engine, where the distance between the terminal located in *Vienna* and *Pilsen* was compared. The shortest distance is 292 km, while the linear regression estimates the distance to be 416 km. The resulting trip planned using *bikerouter.de* takes 372 km, but it utilizes smaller railway lines, which would probably not be capable of running heavier freight trains.

- **Price setting** - for the criteria function, the price of each train slot is calculated using the following formula:

$$p_g = p_t + p_d \cdot l_g, \quad (26)$$

where the p_t is the cost of owning or sourcing the locomotive and p_d is the cost of covering the distance. The l_g describes the distance between the stations of the train slot.

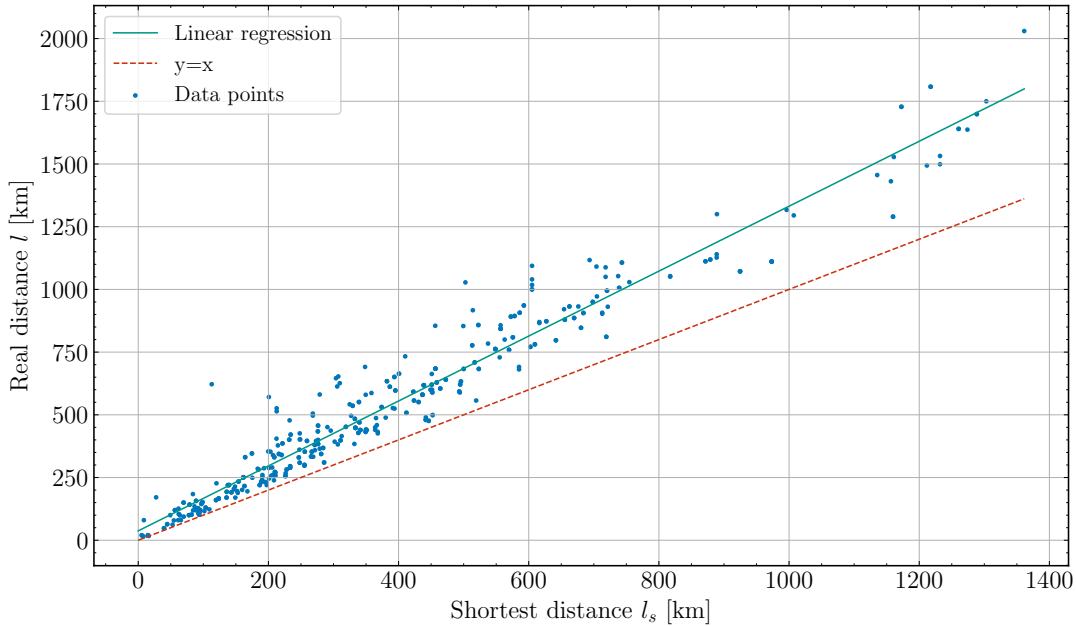


Figure 14: Relation between the shortest route and real-world distance between stations and the proposed linear regression.

An example of the created dataset using the described pipeline is presented in Figure 15. The dataset spans 14 days, contains 406 containers, and allows for selection from 543 train slots. Note that a subset of all METRANS's terminals shown in Figure 6 is taken to create an instance feasible for optimization by the ILP and CP models.

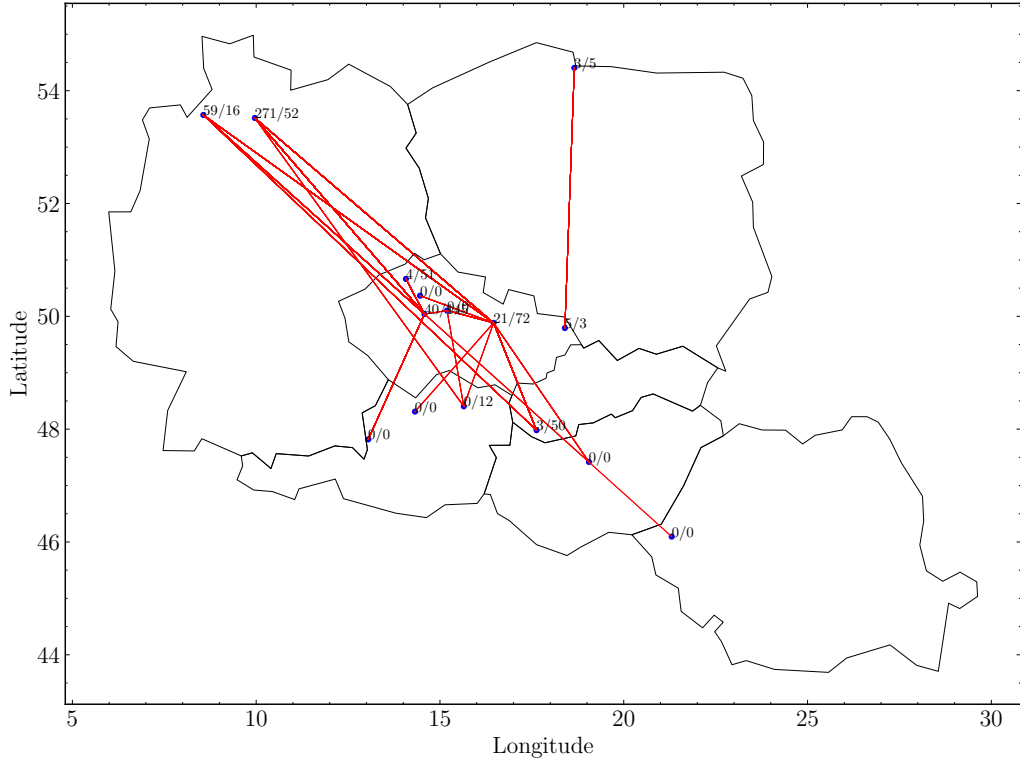


Figure 15: Illustration of the network created by the available train slots. The cities are marked by the number of containers to transport to and from the terminal.

■ 5.5 Solution Quality Measures

After creating the initial models, the achieved criteria value was requested to be compared with the cost of the real-world plan created by the operators. The results are presented in Table 6. Interestingly, the ILP model outperforms the operators as the $\Delta_{\text{METRANS-ILP}}$ is positive in all cases. Note that the instances where the ILP model could provide some solution were selected. The ILP model far superior results even when it does not reach a fully optimal solution, as marked by the ticks and crosses in the table.

$ C $	METRANS	ILP	Optimal	$\Delta_{\text{METRANS-ILP}}$
21	96 575	35 151	✓	61 424
83	264 300	61 258	✗	203 042
195	447 575	119 710	✓	327 865
259	414 975	139 088	✗	275 887
406	582 335	203 734	✗	378 601
580	736 085	270 674	✗	465 411
836	1 026 325	383 524	✗	642 801
1032	1 170 545	465 095	✗	705 450

Table 6: The cost of the plan created by the operators and the ILP model.

5.5 Solution Quality Measures

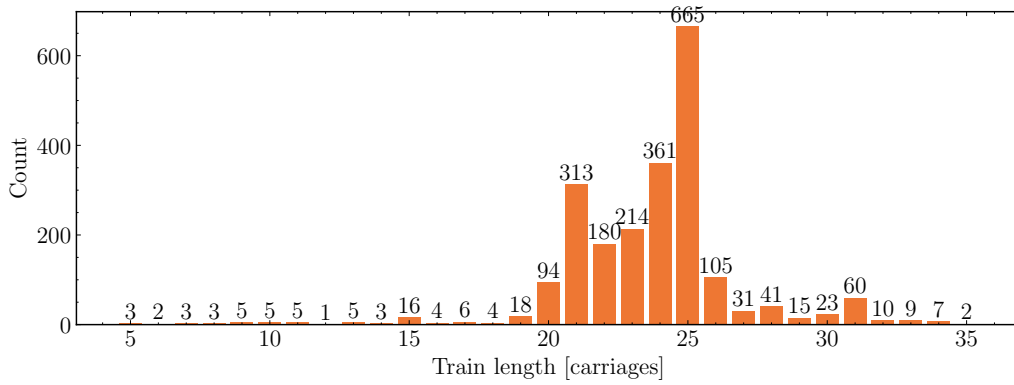


Figure 16: The distribution of the train lengths in carriages.

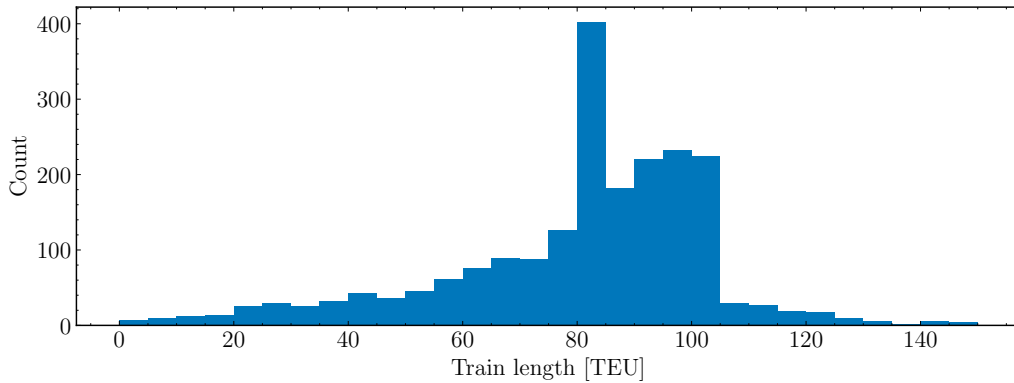


Figure 17: The distribution of the train lengths in TEU.

The difference between the results is caused by the restrictions of the planning horizon and the number of available train slots. Our instances are restricted to fewer days, and containers are transported within such a window. As the operators had to transport containers released before our restriction, they utilized more train slots, and thus, the trains traveled empty when using the same plan for such smaller instances.

Thus, there has to be another measure that could be employed to measure the quality of the solutions. Such a measure could be the utilization of the trains. The good plans would likely request the manipulation of full trains with the containers closely following the shortest paths with minimal train changes. That is why the analysis of utilization of the METRANS plan is created. The number of carriages in trains is shown in Figure 16 as METRANS is not running trains with fixed length. However, in created datasets, the equal length was assumed as modeling the train carriage resources would make the models more complex. The figure shows that METRANS company mainly uses trains with 20 to 26 carriages. The dataset contained some outliers, where the trains included more than 35 carriages. Such trains were filtered out as the European infrastructure would not be able to support such long trains.

Figure 17 shows the distribution of the train lengths in TEU, where the lengths of containers assigned to individual containers were summed together. The number of TEU in the train rises steadily with an outlier around 80 TEU and drops after reaching more than 100 TEU. The spike is probably caused by fully loaded trains of similar length, while the maximal allowed size of the trains does not allow for more than 100 TEU.

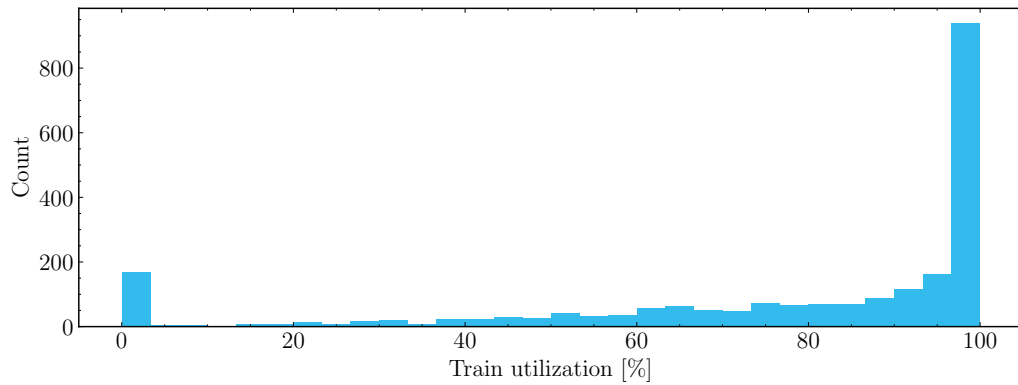


Figure 18: The distribution of the train utilizations.

The dataset did not provide information about the train carriage length in TEU, but a simple manual heuristic search determined that the ideal carriage length is 4 TEU. Figure 18 supports such assumption as the dataset contained just four trains with the utilization over 100 %, which were filtered. METRANS aims to run the trains at maximum capacity, which is suggested by the high count of trains with appropriate utilization. Nevertheless, the imbalanced demand can lead to a need for carriage relocation as the bin around the 0 % is populated as well. Overall, the METRANS operates the trains with the utilization of 77 % on average. Thus, the created models should aim to achieve similar utilization.

The need for such measure engineering could be mitigated by providing a more detailed dataset. The ideal dataset would entail snapshots of the requested container movement inside smaller planning horizons with the plan suggested by the operators. However, METRANS does not store such history about its operation.

5.5 Solution Quality Measures

Chapter 6

Results

This chapter first introduces the minimal instance, which is used to demonstrate the influence of the cost of manipulation on selecting the routes in Section 6.1. Then, the constraints for the most performant ILP model are found in Section 6.2 and Section 6.3. The results of the CP model with integral variables are presented in Section 6.4. The comparison of the ILP and CP models is presented in Section 6.5, where CP is represented by the model with interval variables. Finally, Section 6.6 compares the time needed to build the models, which is an exciting insight into the operation of the models.

6.1 Minimal Instance

The minimal instance is artificially designed to suit the example, of how the selection of cost influences the routes assigned to the containers. An overview of the instance consisting of three terminals, three train slots, and two containers is presented in Figure 19, which depicts the position of the terminals. The costs of the train slots are presented in Table 7, and all slots can transport all containers simultaneously.

$p_{A \rightarrow C}$	$p_{A \rightarrow B}$	$p_{C \rightarrow B}$
40	30	10

Table 7: The costs of train slots between the terminals.

The two containers are in terminal A ; the first container c_1 will be transported to terminal C and the second container c_2 to terminal B . As there does not exist a train slot between in the direction from $B \rightarrow C$, the container c_1 guarantees the selection of the route $A \rightarrow C$.

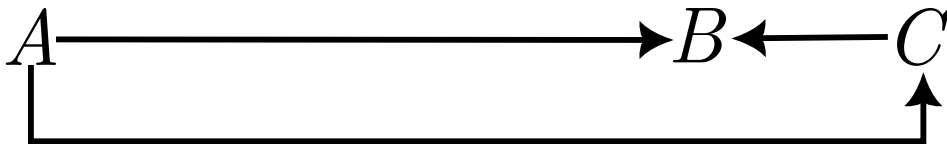


Figure 19: Visual overview of the utilized minimal instance. Note, the instance is similar to the one presented in Figure 8.

The assignment of container c_2 can then be chosen between the routes $A \rightarrow B$ and $A \rightarrow C \rightarrow B$. The first option allows the container to follow a direct path, while the second enables the two containers to be loaded on a single train. As Table 8 shows, the selection of the route is influenced by the cost of the manipulation, which penalizes more fragmented routes. However, the manipulation cost has to be high to outweigh the benefit of loading the two containers on a single train and force the selection of the direct route. The manipulation cost is usually lower than the cost of running a train over longer distances.

The presented experiments were conducted using the cost of manipulation $p_m = 5$, the cost of a train kilometer $p_d = 10$, and the cost of a train per hour being $p_t = 10$. The values of the costs were suggested by METTRANS to match the real-world scenarios roughly. In the final plans, the cost of a train kilometer p_d was usually the most expensive.

p_m	Assigned route	Criterion
5	$A \rightarrow C \rightarrow B$	55
21	$A \rightarrow B$	71

Table 8: The influence of manipulation on the assigned route to the container c_2 .

6.2 Big M Influence

The constraint 9, which sets the y_g variable, can be expressed as:

$$x_{c,g} \leq y_g \quad \forall g \in G, \forall c \in C. \quad (27)$$

For the model with 27 constraints, the number of constraints grows. Each big M constraint 9 is replaced by $|C|$ constraints 27. Thus, this section studies the influence of the increased number of constraints on the model's performance in the Table 9.

$ C $	Objective value		Utilization	
	M	\mathcal{M}	M	\mathcal{M}
2	15 693	15 693	10.00 %	10.00 %
33	49 350	49 350	37.78 %	37.78 %
92	88 147	88 147	63.06 %	63.06 %
182	<u>159 245</u>	167 818	77.12 %	71.90 %
359	<u>288 622</u>	353 735	81.63 %	63.12 %
593	<u>450 879</u>	615 957	87.50 %	63.16 %
732	X	X	X	X

Table 9: Comparison of the models with the big M constraint and with the expanded constraint denoted as \mathcal{M} .

The two approaches were compared on instances with a solver's time limit of 1000s and the results can be found in Table 9. While the two approaches achieve similar results on the minor instances, the reduction in the number of constraints helps to reach lower criteria values for the instances involving more containers. As a result, the model with the big M constraint was chosen for further experiments.

6.3 Restriction Constraints in ILP Model

The ILP model can be further improved by restricting the possible train slots for each container to those falling inside an interval constrained by the release and deadline of the container. The constraints would be formalized as:

$$\sum_{h \in H} x_{c,h} = 0 \quad H = \{\forall g \in G : \psi_c \leq \psi_g \wedge \phi_c \geq \phi_g\}, \forall c \in C, \quad (28)$$

and is named $M + R$.

The model $M + R$ was improved by completely removing variables for the train slots, which can not be selected for the container due to the release and deadline. Such approach was then called $M + \mathcal{R}$.

Table 10 compares the results of the ILP models. The models were given 1000s to find the best solutions on given instances. Sadly, no model could find the optimum for more than

359 containers. Then, the $M + R$ and $M + \mathcal{R}$ outperform the M model and can provide some solutions for bigger instances. Interestingly, omitting variables in $M + \mathcal{R}$ model resulted in a slight improvement in the objective value and the $M + \mathcal{R}$ model is used in further comparisons. All three models can achieve better utilization than METRANS on larger instances, which suggests the created plans would behave well in real-world scenarios.

$ C $	Objective value			Utilization		
	M	$M + R$	$M + \mathcal{R}$	M	$M + R$	$M + \mathcal{R}$
2	15 693	15 693	15 693	10.00 %	10.00 %	10.00 %
33	49 350	49 350	49 350	37.78 %	37.78 %	37.78 %
92	88 147	88 147	88 147	63.06 %	63.06 %	63.06 %
182	159 245	159 245	159 245	77.12 %	77.12 %	77.12 %
359	288 622	288 622	288 622	81.63 %	81.63 %	81.63 %
593	450 879	429 269	<u>429 254</u>	87.50 %	90.37 %	89.78 %
732	x	<u>540 313</u>	543 198	x	89.34 %	88.27 %
768	x	563 147	<u>554 564</u>	x	89.43 %	91.63 %
1194	x	x	x	x	x	x

Table 10: Comparison of ILP with the big M constraint and restricted variants of ILP $M + R$, or $M + \mathcal{R}$.

6.4 Capabilities of a CP Model with Integer Variables

As depicted in Table 11, the performance of the CP approach with integral variables denoted as CP is inferior to the model constructed using interval variables called CP₂. In the given computational budget of 1000s, the simple model solved just very simple instances of the generated problems. Such behavior was expected, as the **CPLEX** solver is more suitable to work with interval variables.

$ C $	$ G $	Objective value		Utilization	
		CP	CP ₂	CP	CP ₂
2	345	15 693	15 693	10.00 %	10.00 %
33	426	49 350	49 350	37.78 %	37.78 %
92	465	x	<u>109 086</u>	x	53.42 %

Table 11: Performance of the small CP model.

6.5 Capabilities of CP Model with Interval Variables

The CP model with interval variables (CP₂) is directly compared with the $M + \mathcal{R}$ ILP model in Table 12. The $M + \mathcal{R}$ model can achieve better objective values thanks to the better utilization of the train slots. The CP₂ can solve approximately instances of double the size. Furthermore, the $M + \mathcal{R}$ model is able to determine the optimality of its solution on more instances than the CP₂ model.

$ C $	$ G $	Objective value		Optimality		Utilization	
		CP ₂	$M + \mathcal{R}$	CP ₂	$M + \mathcal{R}$	CP ₂	$M + \mathcal{R}$
2	345	15 693	15 693	✓	✓	10.00 %	10.00 %
33	426	49 350	49 350	✗	✓	37.78 %	37.78 %
92	465	109 086	<u>88 147</u>	✗	✓	53.42 %	63.06 %
182	499	226 125	<u>159 245</u>	✗	✓	67.79 %	77.12 %
359	545	466 127	<u>288 622</u>	✗	✗	75.36 %	81.63 %
593	581	746 776	<u>429 254</u>	✗	✗	77.12 %	89.78 %
732	625	1 102 860	<u>543 198</u>	✗	✗	78.78 %	88.27 %
768	661	1 168 723	<u>554 564</u>	✗	✗	61.48 %	91.63 %
1194	706	<u>1 914 866</u>	✗	✗	✗	63.63 %	✗
1602	745	<u>3 470 466</u>	✗	✗	✗	43.30 %	✗
1996	779	<u>3 989 626</u>	✗	✗	✗	44.02 %	✗
2369	825	✗	✗	✗	✗	✗	✗

Table 12: Comparison of the CP and ILP models.

6.6 Building the Models

For a better understanding of the models' capabilities, Table 13 compares the build times of each model for the instances the model solved. The summing constraints of big M halved the time needed to create the model compared to \mathcal{M} . The addition of restricting constraints introduced in model $M + \mathcal{R}$ led to a slight increase in the build times and improved the number of solved instances. Ommiting variables in the $M + \mathcal{R}$ model helped to reduce the build times of the $M + \mathcal{R}$ model close to the M model. Despite the atrocious performance, the models for the simple CP approach took little time to build. The CP₂ column presented perhaps the most exciting results as the build times are significantly larger than the ones of the ILP models. The CP₂ models for the biggest tackled instances took longer to build than the computational budget of 1000s provided to the solver. Note, that the build times are not added to the total time available to the solver. Nevertheless, in the real-world deployment the addition of the durations should be considered. Finally, Figure 20 visualizes the build times added to time needed to provide the solution for the $M + \mathcal{R}$ and CP₂ models. The build time is insignificant for the smaller instances, where both models are able to provide results. The shorter solution time for $M + \mathcal{R}$ model on 182 containers was achieved by finding the optimal solution quickly. However, for the larger instances, solved by CP₂ model, the build time takes up significant portion of the total time needed to provide a solution.

$ C $	$ G $	Build time [s]					
		\mathcal{M}	M	$M + R$	$M + \mathcal{R}$	CP	CP ₂
2	345	0.11	0.08	0.09	0.09	0.04	0.03
33	426	0.99	0.62	0.56	0.70	0.64	0.73
92	465	3.04	1.78	1.63	2.10	\times	2.62
182	499	6.48	3.21	3.93	4.87	\times	6.81
359	545	13.63	6.81	8.55	7.17	\times	20.87
593	581	23.94	12.04	15.29	13.95	\times	69.28
732	625	\times	\times	20.87	17.11	\times	115.61
768	661	\times	\times	23.83	18.56	\times	138.14
1194	706	\times	\times	\times	\times	\times	354.95
1602	745	\times	\times	\times	\times	\times	477.88
1996	779	\times	\times	\times	\times	\times	1008.34
2369	825	\times	\times	\times	\times	\times	1617.84

Table 13: Comparison of the build times.

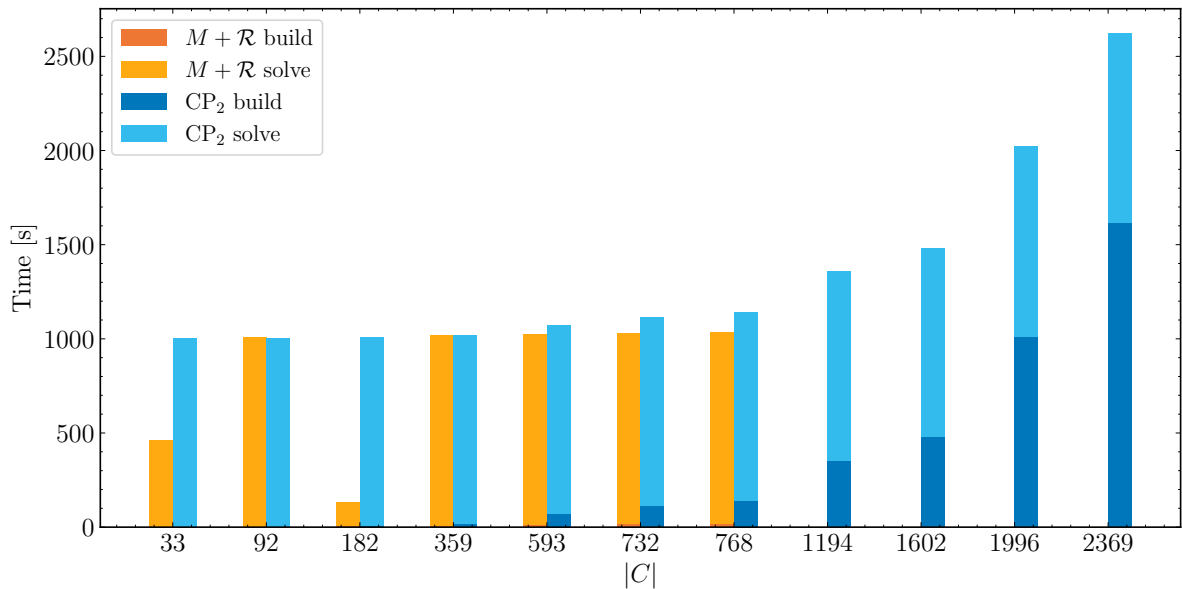


Figure 20: Visualization of the build times. If there is no bar for a model at a given instance, the model did not solve the instance.

6.6 *Building the Models*

Chapter 7

Conclusion & Future Work

This thesis presents a solution to the container routing problem. Such a problem arises in the case of intermodal transportation networks where goods inside containers are transported using multiple modes of transport, as the goal is to exploit the capacity of each transportation resource to the fullest. As the number of transfer points grows, so does the number of available paths of a single package through the network, which further increases the complexity of routing multiple such packages. The need for optimization has led freight transportation company METRANS to request an examination of possible approaches.

The presented solution involves first studying the exact optimization paradigms ILP and CP. Then, the related literature is reviewed, mainly focusing on freight transportation on railways due to its unique structure. The review is followed by a precise formulation of the container routing problem and proof of its NP-complexity. Models in both CP and ILP are presented and then executed using the **CPLEX** solver.

The METRANS company provided a dataset to allow for the testing of proposed models. A tool for creating smaller instances of the provided dataset is introduced as the exact approaches cannot handle the whole dataset. Additionally, as the recorded data is not entirely consistent, the data is filtered, and missing values are extrapolated when viable.

The smaller dataset instances are used to test the models. The results show that the ILP model benefits from cleverly engineered constraints. On the other hand the CP approach profits from the usage of interval variables compared to integral ones. The ILP model achieves lower optimal values on instances where it finds some initialization, and the CP model can solve larger instances. The results show that the models can achieve similar utilization of the network's resources to the plan employed by METRANS in the provided dataset. Thus, the performance of the models is satisfactory.

A study of cooperation with a constructive heuristic would be a promising path to examine in future work. The heuristic approach could provide an initial solution to the problem, and the exact approach could guide the search closer to the optimum. Conversely, the solutions of the exact models could be used to test the heuristic's performance. Finally, the exact approaches can be applied to solve smaller real-world instances to aid the operators. Naturally, such a step would facilitate closer cooperation with the METRANS company and could force the addition of more constraints. Such constraints could help to handle the circulation of a rolling stock and crew scheduling.

7. Conclusion & Future Work

References

- [1] M. facebook page. Metrans terminal. <https://www.facebook.com/photo/?fbid=2881325428827363&set=a.1692588784367706>. [Online; accessed 15-April-2024].
- [2] R. LaGore. Intermodal containers. <https://blog.intekfreight-logistics.com/iso-intermodal-container-defined-and-facts>. [Online; accessed 15-April-2024].
- [3] O. Janiš. Metrans train. https://spz.logout.cz/fotoalbum/fotoalbum_zobrazeni.php?stat=cz&rada=9154_7383_40&vlastnik=cz_096. [Online; accessed 15-April-2024].
- [4] METRANS — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Estimation_lemma&oldid=375747928. [Online; accessed 20-January-2024].
- [5] G. B. Dantzig. Linear programming. *Operations research*, 50(1):42–47, 2002.
- [6] S. I. Gass. The first linear-programming shoppe. *Operations Research*, 50(1):61–68, 2002.
- [7] J. Fearnley and R. Savani. The complexity of the simplex method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 201–208, 2015.
- [8] N. Megiddo et al. *On the complexity of linear programming*. Citeseer, 1986.
- [9] R. Freund and C. Roos. The ellipsoid method. *Optimization*, 1:26, 2007.
- [10] S. Boyd and J. Mattingley. Branch and bound methods. *Notes for EE364b, Stanford University*, 2006:07, 2007.
- [11] L. Brunetta, M. Conforti, and G. Rinaldi. A branch-and-cut algorithm for the equicut problem. *Mathematical Programming*, 78:243–263, 1997.
- [12] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.
- [13] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.
- [14] T. Achterberg. Conflict analysis in mixed integer programming. *Discrete Optimization*, 4(1):4–20, 2007.
- [15] T. Achterberg. Scip: solving constraint integer programs. *Mathematical Programming Computation*, 1:1–41, 2009.
- [16] M. Yokoo. Why adding more constraints makes a problem easier for hill-climbing algorithms: Analyzing landscapes of csps. In *Principles and Practice of Constraint Programming-CP97: Third International Conference, CP97 Linz, Austria, October 29–November 1, 1997 Proceedings 3*, pages 356–370. Springer, 1997.
- [17] Y. Zhang and R. H. Yap. Making ac-3 an optimal algorithm. In *IJCAI*, volume 1, pages 316–321, 2001.

- [18] R. Anand, D. Aggarwal, and V. Kumar. A comparative analysis of optimization solvers. *Journal of Statistics and Management Systems*, 20(4):623–635, 2017.
- [19] A. de Vries. The growing energy footprint of artificial intelligence. *Joule*, 7(10):2191–2194, 2023.
- [20] Gams. <https://www.gams.com/>. [Online; accessed 29-April-2024].
- [21] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, and G. Tack. Minizinc: Towards a standard cp modelling language. In *International Conference on Principles and Practice of Constraint Programming*, pages 529–543. Springer, 2007.
- [22] Gams performance analysis. <https://www.gams.com/blog/2023/07/performance-in-optimization-models-a-comparative-analysis-of-gams-pyomo-gurobipy-and-ju#fnref:2>. [Online; accessed 29-April-2024].
- [23] Manhattan Associates. <https://www.manh.com/>. [Online; accessed 21-January-2024].
- [24] e2open. <https://www.e2open.com/>. [Online; accessed 21-January-2024].
- [25] INFOR. <https://www.infor.com>. [Online; accessed 21-January-2024].
- [26] A. Caprara, L. Kroon, M. Monaci, M. Peeters, and P. Toth. Passenger railway optimization. *Handbooks in operations research and management science*, 14:129–187, 2007.
- [27] T. Robenek, Y. Maknoon, S. S. Azadeh, J. Chen, and M. Bierlaire. Passenger centric train timetabling problem. *Transportation Research Part B: Methodological*, 89:107–126, 2016.
- [28] E. Oliveira and B. M. Smith. A job-shop scheduling model for the single-track railway scheduling problem. 2000.
- [29] C. Liebchen. The first optimized railway timetable in practice. *Transportation Science*, 42(4):420–435, 2008.
- [30] H. R. Varian. Online ad auctions. *American Economic Review*, 99(2):430–434, 2009.
- [31] P. Klemperer. How (not) to run auctions: The european 3g telecom auctions. *European economic review*, 46(4-5):829–845, 2002.
- [32] R. Borndörfer, M. Grötschel, S. Lukac, K. Mitusch, T. Schlechte, S. Schultz, and A. Tanner. An auctioning approach to railway slot allocation. *Competition and Regulation in Network Industries*, 1(2):163–196, 2006.
- [33] D. Canca and E. Barrena. The integrated rolling stock circulation and depot location problem in railway rapid transit systems. *Transportation Research Part E: Logistics and Transportation Review*, 109:115–138, 2018.
- [34] P.-J. Fioole, L. Kroon, G. Maróti, and A. Schrijver. A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, 174(2):1281–1297, 2006.
- [35] Y. Wang, A. D’Ariano, J. Yin, L. Meng, T. Tang, and B. Ning. Passenger demand oriented train scheduling and rolling stock circulation planning for an urban rail transit line. *Transportation Research Part B: Methodological*, 118:193–227, 2018.

- [36] S. Zhao, H. Yang, and Y. Wu. An integrated approach of train scheduling and rolling stock circulation with skip-stopping pattern for urban rail transit lines. *Transportation Research Part C: Emerging Technologies*, 128:103170, 2021.
- [37] Q. Zhong, R. M. Lusby, J. Larsen, Y. Zhang, and Q. Peng. Rolling stock scheduling with maintenance requirements at the chinese high-speed railway. *Transportation Research Part B: Methodological*, 126:24–44, 2019.
- [38] E. Abbink, M. Fischetti, L. Kroon, G. Timmer, and M. Vromans. Reinventing crew scheduling at netherlands railways. *Interfaces*, 35(5):393–401, 2005.
- [39] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37, 2014.
- [40] J. Yin, L. Yang, T. Tang, Z. Gao, and B. Ran. Dynamic passenger demand oriented metro train scheduling with energy-efficiency and waiting time minimization: Mixed-integer linear programming approaches. *Transportation Research Part B: Methodological*, 97:182–213, 2017.
- [41] C. Archetti, L. Peirano, and M. G. Speranza. Optimization in multimodal freight transportation problems: A survey. *European Journal of Operational Research*, 299(1):1–20, 2022.
- [42] T.-E. S. Hanssen, T. A. Mathisen, and F. Jørgensen. Generalized transport costs in intermodal freight transport. *Procedia - Social and Behavioral Sciences*, 54:189–200, 2012. Proceedings of EWGT2012 - 15th Meeting of the EURO Working Group on Transportation, September 2012, Paris.
- [43] P. Corry and E. Kozan. An assignment model for dynamic load planning of intermodal trains. *Computers and Operations Research*, 33(1):1–17, 2006.
- [44] L. Fan, W. W. Wilson, and D. Tolliver. Optimal network flows for containerized imports to the united states. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):735–749, 2010.
- [45] H. Heggen, Y. Molenbruch, A. Caris, and K. Braekers. Intermodal container routing: integrating long-haul routing and local drayage decisions. *Sustainability*, 11(6):1634, 2019.
- [46] T.-S. Chang. Best routes selection in international intermodal networks. *Computers & operations research*, 35(9):2877–2891, 2008.
- [47] M. Verma, V. Verter, and N. Zufferey. A bi-objective model for planning and managing rail-truck intermodal transportation of hazardous materials. *Transportation research part E: logistics and transportation review*, 48(1):132–149, 2012.
- [48] Y. Sun. Green and reliable freight routing problem in the road-rail intermodal transportation network with uncertain parameters: a fuzzy goal programming approach. *Journal of Advanced Transportation*, 2020:1–21, 2020.
- [49] bikerouter.de. <https://bikerouter.de/#map=8/50.637/15.485/standard&profile=rail>. [Online; accessed 18-April-2024].

Appendix A

Running the Experiments

Running the experiments requires the METRANS dataset to be present as the `METRANS_data` folder in the root of the codebase. The `requirements.txt` file describes the utilized packages that can be downloaded via `pip`, a package manager. Additionally, smaller instances are provided to test the basic functionality of the models, and the structure of the codebase is as follows:

- `dataset_creator.py` - allows for the creation of smaller dataset instances. The raw dataset is filtered to contain consistent data about the containers and train slots and preprocessed into a format suitable for the models. The script mainly enables the user to set the time window for the transportation of the containers, which controls the size of the dataset.
- `ilp/cp/cp2.py` - include the formalizations of the models in the Python language. The models can accept instances the `dataset_creator.py` provides. Usually, the user provides a time limit to constrain the execution of the model.
- `visualization_creator.py` - creates all Figures and Tables seen in this thesis.
- `app.py` - hosts solution visualizations in an interactive app. The `dash` library of `plotly` package is used to create the app. The app helps display the exact routing of container batches and information about the dataset. Note that the creation of a user interface compatible with today's requirements was out of the scope of this thesis.
- `batch_experiments.py` - was used to create batches of experiments for the computational server.
- `src` - includes the source code for the parsing of files and creation of visualizations and tables needed for the composition of this thesis.
- `generated_datasets/dataset_showcase` - contains a small instance for showcase purposes.

Most of the utilized scripts can be commanded using a command line, and the precise information about the arguments can be invoked using the `--help` flag. A `Dockerfile` is provided with an image with the **CPLEX** solver as the experiments were run on a server to offload the computation burden of the personal laptop. After cloning the repository, the following commands can be used to run the container routing on a simple instance:

```
pip install -r requirements.txt
python ilp.py -t 100 dataset_showcase
python app.py
```