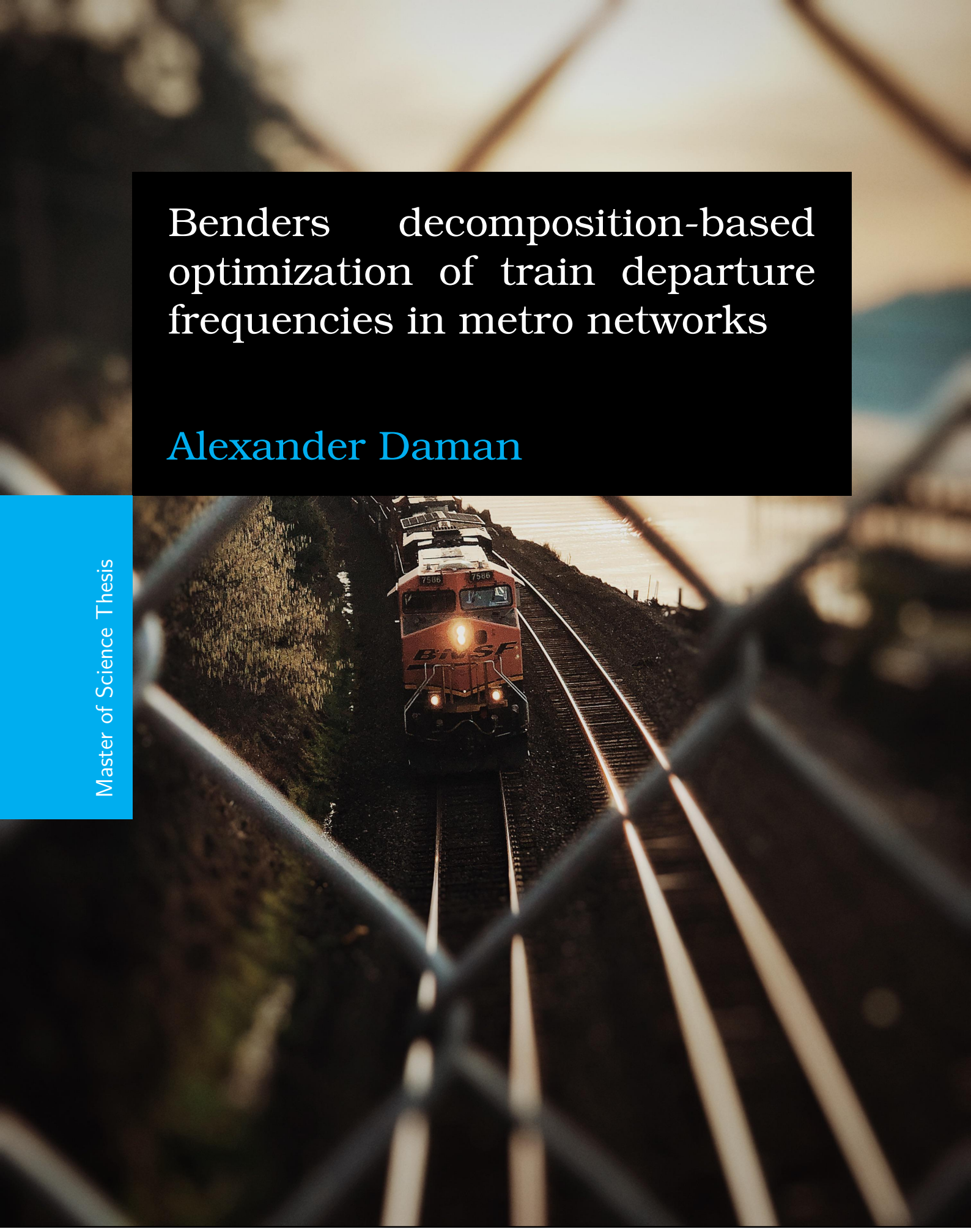
Benders decomposition-based optimization of train departure frequencies in metro networks

Alexander Daman



Master of Science Thesis

TU Delft Delft University of Technology

Delft Center for Systems and Control

Benders decomposition-based optimization of train departure frequencies in metro networks

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

Alexander Daman

June 16, 2023

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

Timetables determine the service quality for passengers and the energy consumption of trains in metro systems. In metro networks, a timetable can be made by optimizing train departure frequencies for different periods of the day. Typically, the optimization problem that arises from optimizing train departure frequencies in metro networks involves integer variables, which can cause the computational complexity of the optimization problem to be too high for real-time applications. The main objective of this thesis is to reduce the computational complexity of optimizing train departure frequencies in metro networks while maintaining a relatively accurate solution.

In this thesis, we first apply classical Benders decomposition to optimize train departure frequencies in a metro network considering time-varying origin-destination passenger demands. Subsequently, we apply ϵ-optimal Benders decomposition to reduce the computational complexity further. A simulation-based case study using a grid metro network illustrates the performance of the two Benders decomposition-based approaches.

The simulation results show that the classical Benders decomposition approach significantly reduces the computational burden of optimizing train departure frequencies in metro networks. Moreover, the ϵ-optimal Benders decomposition approach can further reduce the computation time of the optimization problem when the problem sizes increases while maintaining an acceptable level of performance.

Table of Contents

Acknowledgements

First, I would like to express my sincere gratitude to Xiaoyu Liu, who enthusiastically guided me throughout my research. I was able to complete this thesis thanks to the insights and numerous meetings I have had with Xiaoyu.

I would also like to thank prof.dr.ir. Bart De Schutter for providing crucial feedback and asking critical questions, improving the quality of this thesis.

Next, I owe many thanks to my parents and sisters for keeping me motivated throughout my thesis and in life in general.

I want to thank Laura for supporting me no matter what, making life more pleasant, and providing me with a place to sleep after long nights spent in the library.

Last but least, I want to thank my friends and peers, who have made my life in Delft and Rotterdam intellectually stimulating and, more importantly, fun.

Delft, University of Technology
June 16, 2023

Alexander Daman

Chapter 1

Introduction

Metro systems have become essential to urban transportation, providing millions of people with fast, efficient, and sustainable travel options, especially in large cities. The metro system is particularly critical in densely populated urban areas, where an efficient and reliable timetable is paramount for passenger satisfaction and the energy efficiency of the metro system. According to the International Energy Agency (IEA), rail accounts for 9% of the world's passengers and 7% of global freight transport while only representing 3% of transport energy use [9]. Generally, rail requires 12 times less energy per passenger kilometer than cars and airplanes and is one of the most energy-efficient methods for transporting goods [9]. In order to realize the goals of the Paris Agreement to cut greenhouse gas emissions, increasing the share of rail use for passenger and freight transport will be vital.

In addition, rail is a relatively safe method of transportation. According to the Dutch Institute for Road Safety Research (SWOV), there were about 14 times as many fatal accidents involving cars compared with trains in the Netherlands between 2007 and 2016 [49].

Due to the energy efficiency and safety of rail, countries all over the world have been working on upgrading and expanding their railway systems. With the increase in the use of rail, efficient methods need to be developed to ensure passengers get to their destination as quickly as possible while limiting operational costs. Real-time timetable scheduling is a commonly used approach for creating efficient timetables. Efficient train scheduling approaches enable metro systems to minimize operational costs, reduce waiting times, and adjust transport capacity to meet passenger demands for different time periods.

A Nonlinear Programming (NLP) problem was formulated in [53] to minimize the time passengers spend in the metro network and the energy consumption of the dispatched trains, for which an iterative convex programming approach was proposed. A bi-directional train line was considered in [25], and a Lagrangian-based method was applied to solve the resulting NLP problem. An adaptive large neighborhood search algorithm was developed in [4] for the timetable scheduling problem of a rail rapid transit line to create convenient timetables for passengers considering a dynamic demand pattern. While passenger satisfaction typically only includes the time passengers spend in the rail network, the time passengers spend

waiting outside stations due to station capacity limits was included in [69]. Since there can be significant differences in passenger demands per station, the possibility of trains skipping low-demand stations in their model was considered in [54].

The train departure frequency, which refers to the number of trains departing from a line per time unit, determines the transport capacity of metro networks. To handle time-varying passenger Origin-Destination (OD) demands, effective strategies must be implemented to optimize departure frequencies in real time. Line frequencies and train capacities were optimized using both an exact algorithm and a heuristic approach in [20]. A Mixed-Integer Nonlinear Programming (MINLP) problem was formulated in [11] to optimize line frequencies and capacities in metro networks. A novel passenger absorption model was proposed in [32] to optimize the departure frequency of trains of each line in metro networks, and the resulting problem was formulated as a Mixed-Integer Linear Programming (MILP) problem.

Real-time timetable scheduling models often involve integer variables, resulting in non-convex optimization problems that can be time-consuming. Benders decomposition is an efficient methodology to reduce the computational burden in large-scale MILP problems by splitting the MILP into two small-scale problems [5, 43]. Benders decomposition has been successfully applied to railway timetable scheduling problems. For example, considering the uncertain passenger transfer time in metro networks, a generalized Benders decomposition approach was developed in [26] to efficiently solve the resulting MILP problem. A logic-based Benders decomposition approach that could reuse the precomputed logic Benders cuts to reduce the computational burden of the timetable rescheduling problem was applied in [30]. In [28], which focused on modifying train routes and schedules in the case of train delays, the solution time of the Benders decomposition algorithm was reduced by splitting the algorithm solution process into three steps to address the fact that the relation between routing and scheduling variables is absent in the master problem.

The Benders decomposition approaches applied in [26, 28, 30] were all shown to reduce the solution time significantly; however, passenger OD demands were not considered explicitly.

1-1 Problem statement

Benders decomposition-based approaches have been shown to be able to significantly reduce the solution time of timetable scheduling problems, enabling the use of these approaches for real-time implementation. In addition, the passenger absorption model introduced in [33] can explicitly include time-varying OD passenger demands and provides a balanced trade-off between model accuracy and solution time. The passenger absorption model can be used to formulate an optimization problem in which the train departure frequencies in a metro network are optimized.

This thesis aims to investigate the possibilities of Benders decomposition in real-time timetable scheduling. The main question can be summarized as follows:

Are Benders decomposition-based approaches suitable for optimizing train departure frequencies in metro networks?

Benders decomposition will be applied to a passenger-oriented timetable scheduling model

to optimize train departure frequencies in metro networks. The performance of the Benders decomposition-based approaches will be evaluated in terms of the objective function — a combination of the time spent by passengers in the rail network and the operational costs — and the computation time. The main question can be used to derive two sub-questions:

1. *Can Benders decomposition reduce the computational complexity of optimizing train departure frequencies in metro networks?*

   A state-of-the-art solver will serve as a benchmark to evaluate the performance of the Benders decomposition-based approaches.

2. *What acceleration methods can be applied to Benders decomposition when optimizing train departure frequencies in metro networks?*

   Much research has focused on accelerating the classical Benders decomposition algorithm. Choosing a suitable acceleration method is not trivial and is often problem-specific.

1-2 Thesis outline

The remainder of this thesis is structured as follows. Chapter 2 presents relevant background information and introduces Benders decomposition. Chapter 3 discusses the classical and ϵ-optimal Benders decomposition algorithms that are used in this thesis. Chapter 4 evaluates the Benders decomposition-based approaches in a simulation-based case study. Finally, Chapter 5 concludes this thesis.

Chapter 2

Background

This chapter overviews state-of-the-art techniques and research in passenger-centric timetable scheduling. The chapter is structured as follows. In Section 2-1, different approaches to railway traffic management are discussed. Section 2-2 presents an overview of Mixed-Integer Linear Programming (MILP) and commonly used solution approaches for MILP problems. Subsequently, Benders decomposition is discussed in Section 2-3. Finally, the chapter is concluded in Section 2-4.

2-1 Railway traffic management

Railway operators generally rely on a hierarchical decision-making structure to plan and manage railway operations, breaking down the process into smaller sub-problems. The main sub-problems in the planning hierarchy are shown in Figure 2-1.

The first stage of the decision-making structure is referred to as the strategic level, which comprises two phases, i.e., network planning and line planning. Network planning focuses on the construction and maintenance of the railway infrastructure. In contrast, line planning optimizes train routes, train frequencies, and types of trains to meet passenger demands and ensure passenger satisfaction. Passenger satisfaction generally involves the time passengers spend in the metro system. Operational costs are also considered in the line planning phase.

At the tactical level, railway operators allocate the available resources to comply with the outcomes of the strategic phase through effective timetable scheduling, rolling stock circulation, and crew scheduling. The timetable scheduling phase involves determining the departure and arrival times of all the train lines. Typically, the objectives of this phase are to maximize passenger satisfaction and minimize operational costs, as evident in various literature [39, 40, 47, 50, 52, 53, 57, 62, 63]. At the same time, some studies focus solely on passenger satisfaction [3, 4, 8, 10, 16, 24, 25, 48, 51, 64, 66, 69]. In some research, train speed profiles are

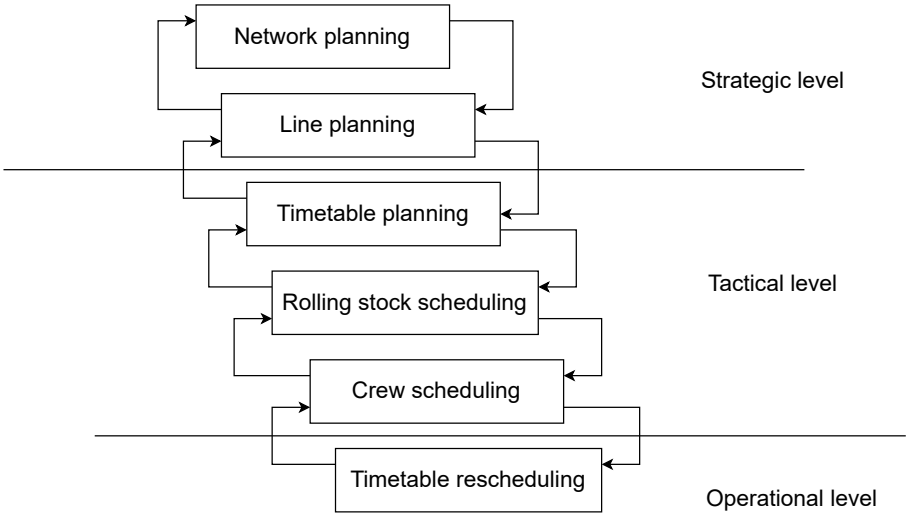


Figure 2-1: Railway operations planning. Adapted from [39].

also considered, dividing train runs into several operational phases, including the acceleration phase, cruising phase, coasting phase, and deceleration phase [54, 60, 61].

There are two main types of timetables in the daily operation of railway networks: periodic and non-periodic. In a periodic timetable, each scheduled event occurs every cycle period, e.g., one hour, whereas non-periodic timetables do not follow a fixed cycle period. Scheduling non-periodic timetables in real time can be more challenging than periodic timetables. Non-periodic timetables are advantageous when considering inhomogeneous passenger demands, often encountered in real-life scenarios, as non-periodic timetables are not restricted by cycle periods and offer more flexibility in accommodating varying passenger demands.

After the timetable scheduling phase, railway operators must consider rolling stock circulation, which involves solving the shunting problem. Trains must be moved to a shunting yard between operations for parking or maintenance purposes. In some cases, timetable and rolling stock scheduling are integrated to optimize operational costs and passenger satisfaction [8, 39, 40, 50, 64]. However, this approach may require increased computational effort. Finally, crew duties — including maintenance crews and conductors — are scheduled to complete the tactical operation phase. Relevant examples of research in this area include [12, 42, 46].

At the operational level, railway operators need to manage the railway system in real time to handle disturbances and disruptions during railway operations. In railway traffic management, disturbances refer to events that cause relatively minor changes to the timetable. At the same time, disruptions typically describe events that require rescheduling the timetable and alteration of resource allocations. Disturbance and disruption management algorithms focus on reducing delays caused by these events while ensuring passengers can reach their destinations. Fast algorithms are critical for real-time timetable adjustments to minimize delays. Examples of relevant literature on disturbance management include [35–37, 70], while examples of literature on disruption management include [6, 13, 21, 34, 67, 68].

Smart card technologies are widely used in railway systems, providing detailed time-varying origin-destination (OD) passenger travel data that can be used to predict passenger demands. Passenger demands are typically measured during specific time intervals. There are two main approaches in the literature for describing passenger demand: OD-independent and OD-dependent. OD-independent passenger demand models only consider passenger demands between stations, while OD-dependent models consider the origin and destination of passengers when scheduling the timetable. Robust methods have been developed in [10,31,55,62,63] to deal with the uncertainty of passenger demands.

2-2 Mixed-Integer Linear Programming

The optimization problems encountered in passenger-centric timetable scheduling often involve MILP problems, as they typically contain a combination of continuous and integer variables. The model used in this thesis can also be used to derive a MILP problem. MILP problems are typically classified as NP-hard problems, which means that the search space, and therefore the computation time, increases exponentially with the size of the problem. It is essential to choose a suitable solution approach for the optimization problem. In this section, the possible solution approaches for MILP problems are discussed.

2-2-1 Solution approaches

Branch-and-Bound

Branch-and-Bound (BB) algorithms are commonly used for MILP problems and have been successfully applied in railway traffic management problems, as demonstrated in prior studies such as [10, 36, 37]. BB algorithms partition the solution space into smaller subsets, corresponding to different values of the integer variables in the case of MILP problems, and eliminate subsets that do not contain the optimal solution based on a bound, which allows BB algorithms to significantly reduce computation time compared to evaluating every possible solution.

Commercial solvers such as `cplex` and `gurobi`, which are commonly used in railway traffic management problems, often employ BB algorithms. Both solvers incorporate enhancements to the standard BB algorithm, such as strong branching and feasibility heuristics. For example, `gurobi` combines BB with a barrier method [1], which is a type of interior-point method that is particularly effective at solving Linear Programming (LP) problems with many constraints [7]. The barrier method is used to solve the LP relaxations within each branch of the BB tree, which can improve the lower bounds and pruning of infeasible branches. On the other hand, `cplex` combines BB with a simplex method [2], which is a classical LP-solving technique that iteratively improves a feasible solution until it reaches an optimal solution or proves optimality. Both `cplex` and `gurobi` are often used for optimization problems in real-time timetable scheduling, leading to satisfactory results.

Heuristics

Heuristic search algorithms are commonly used to find approximate solutions for large-scale MILP problems. A drawback of heuristic algorithms is that they are not guaranteed to find the optimal or even a feasible solution [44]. Examples of heuristic algorithms used in railway traffic management include:

- Neighborhood search algorithms: Neighborhood search algorithms find a solution to an optimization problem by exploring solutions in the vicinity of the current best solution. Neighborhood search algorithms have been used in studies concerning railway traffic management such as [17, 24, 67].
- Genetic algorithms: Genetic algorithms simulate natural selection and evolution to find a solution. Genetic algorithms start with a population of potential solutions, evaluate their fitness based on an objective function, and then apply genetic operators to create new offspring until a satisfactory solution is found. Genetic algorithms have been applied in studies such as [47, 52, 57, 60] in the context of railway traffic management.

Dantzig-Wolfe decomposition

Dantzig-Wolfe decomposition is a powerful mathematical technique developed in [19] for solving LP problems, and has been widely used in various fields, including railway traffic management. The method involves breaking down a complex LP problem into smaller, more manageable sub-problems, offering an efficient approach for solving large-scale optimization problems.

In the Dantzig-Wolfe decomposition approach, the original problem is formulated as a master problem, which consists of a set of constraints and decision variables that describe the overall problem. The master problem is then decomposed into a set of sub-problems, each associated with a subset of the original decision variables. These sub-problems can be solved independently, subject to constraints that ensure consistency among their solutions.

The solutions obtained from solving the sub-problems are combined to obtain the solution to the original problem. This approach allows for efficient computation of solutions by exploiting the structure of the problem and leveraging the interactions among the sub-problems. Examples of the successful application of Dantzig-Wolfe decomposition in railway traffic management can be found in studies such as [12, 14, 42, 46]

Benders decomposition

Benders decomposition, introduced in [5], is a widely used method for solving large-scale optimization problems involving complex variables. Benders decomposition is closely related to Dantzig-Wolfe decomposition and is, in fact, equivalent to Dantzig-Wolfe decomposition applied to the dual for LP problems [18].

The core idea of Benders decomposition is to divide the original problem into a master problem and a sub-problem. The master problem is solved first, and the solution is then used as input for solving the sub-problem. If the sub-problem finds a feasible solution, it is added to the master problem, and the process is iteratively repeated until a satisfactory solution is

obtained. If the sub-problem is infeasible, a constraint is added to the master problem to eliminate the infeasible solution.

Benders decomposition has found successful applications in various areas of railway traffic management, as evidenced by studies such as [26, 28–30, 45].

2-2-2 Conclusions

Heuristic algorithms can be useful in specific scenarios to quickly obtain a feasible or approximate solution for MILP problems. However, they are not guaranteed to find the optimal solution, and the solution quality may vary depending on the specific problem instance. On the other hand, exact optimization methods such as Dantzig-Wolfe decomposition and Benders decomposition are guaranteed to find the optimal solution given enough computational resources.

While Benders decomposition and Dantzig-Wolfe decomposition are similar for LP problems, they differ when integer variables are present. In Dantzig-Wolfe decomposition, the relaxed problem is an LP relaxation of the original MILP problem, where the integer constraints are removed, allowing the problem to be solved as an LP problem. The solution to this relaxed problem provides a lower bound on the optimal solution of the original MILP problem. By iteratively improving the relaxed solution, the Dantzig-Wolfe decomposition can eventually converge to the optimal solution of the original MILP problem.

On the other hand, the Benders decomposition algorithm directly converges to an optimal solution without embedding in a BB framework, making Benders decomposition more efficient than Dantzig-Wolfe decomposition for solving MILP problems with integer variables [43]. Hence, for our optimization problem, we decide to use Benders decomposition.

2-3 Benders decomposition

This section introduces the classical Benders decomposition algorithm [5] and discusses potential enhancement strategies.

2-3-1 Classical Benders decomposition

Suppose an MILP problem is considered, consisting of continuous variables, discrete variables, equality constraints, and inequality constraints, in the following form:

$$\begin{aligned} \min_{x,y} \quad & c^{\mathrm{T}}x + f^{\mathrm{T}}y && \text{(2-1)} \\ \text{subject to} \quad & Ax + By = b && \text{(2-2)} \\ & Dx + Ey \leq d && \text{(2-3)} \\ & Gy = g && \text{(2-4)} \\ & Fy \leq e && \text{(2-5)} \\ & x \in \mathbb{R}^{n_1} && \text{(2-6)} \\ & y \in \mathbb{Z}^{n_2} && \text{(2-7)} \end{aligned}$$

Here, $y \in \mathbb{Z}^{n_2}$ are the so-called 'complicating' variables as they are integers and satisfy constraints $Gy = g$ and $Fy \leq e$, with $G \in \mathbb{R}^{g_1}$; $F \in \mathbb{R}^{g_2}$; $g \in \mathbb{R}^{g_1}$; and $e \in \mathbb{R}^{g_2}$. The continuous variables are represented by $x \in \mathbb{R}^{n_1}$ and, together with y, satisfy the constraints $Ax+By = b$ and $Dx + Ey \leq d$, with $A \in \mathbb{R}^{m_1 \times n_1}$; $B \in \mathbb{R}^{m_1 \times n_2}$; $D \in \mathbb{R}^{m_2 \times n_1}$; $E \in \mathbb{R}^{m_2 \times n_2}$; $b \in \mathbb{R}^{m_1}$; and $d \in \mathbb{R}^{m_2}$.

Complicating variables y are fixed as $\bar{y}$, after which 2-1 becomes:

$$\begin{aligned} \min_{x} \quad & c^{\mathrm{T}}x + f^{\mathrm{T}}\bar{y} && \text{(2-8)} \\ \text{subject to} \quad & Ax + B\bar{y} = b && \text{(2-9)} \\ & Dx + Ey \leq d && \text{(2-10)} \\ & x \in \mathbb{R}^{n_1} && \text{(2-11)} \end{aligned}$$

The original MILP problem is re-expressed as:

$$\min_{\bar{y} \in \mathbb{Z}^{n_2}} \left[f^{\mathrm{T}}\bar{y} + \min_{x \in \mathbb{R}^{n_1}} \left\{ c^{\mathrm{T}}x \mid Ax = b - B\bar{y}, Dx \leq d - E\bar{y} \right\} \right] \tag{2-12}$$

Dual variables $u_1 \in \mathbb{R}^{m_1}$ and $u_2 \in \mathbb{R}^{m_2}$ are introduced, that satisfy constraint $u_1^{\mathrm{T}}A + u_2^{\mathrm{T}}D = c^{\mathrm{T}}$. Then, $c^{\mathrm{T}}x$ is rewritten as $u_1^{\mathrm{T}}Ax + u_2^{\mathrm{T}}Dx$. Since $Ax = b - B\bar{y}$, $u_1^{\mathrm{T}}Ax$ can be re-expressed as $u_1^{\mathrm{T}}(b - B\bar{y})$, and since $Dx \leq d - E\bar{y}$ it follows that $u_2^{\mathrm{T}}Dx \leq d - E\bar{y}, \quad \forall u_2 \geq 0$. In other words, the following inequality holds:

$$-\left(u_1^{\mathrm{T}}(b - B\bar{y}) + u_2^{\mathrm{T}}(d - E\bar{y})\right) \leq u_1^{\mathrm{T}}Ax + u_2^{\mathrm{T}}Dx \qquad \forall u_2 \geq 0 \tag{2-13}$$

Therefore, with a fixed $\bar{y}$, the best lower bound of $c^{\mathrm{T}}x$ can be found by solving the following problem (referred to as the dual sub-problem):

$$\begin{aligned} \max_{u_1,u_2} \quad & -\left(u_1^{\mathrm{T}}(b - B\bar{y}) + u_2^{\mathrm{T}}(d - E\bar{y})\right) && \text{(2-14)} \\ \text{subject to} \quad & u_1^{\mathrm{T}}A + u_2^{\mathrm{T}}D = c^{\mathrm{T}} && \text{(2-15)} \\ & u_1 \in \mathbb{R}^{m_1} && \text{(2-16)} \\ & u_2 \in \mathbb{R}^{m_2} && \text{(2-17)} \\ & u_2 \geq 0 && \text{(2-18)} \end{aligned}$$

The feasible space of the dual sub-problem is denoted as: $\Omega = \left\{u_1, u_2 \mid u_1^{\mathrm{T}} A + u_2^{\mathrm{T}} D = c^{\mathrm{T}}, u_2 \geq 0\right\}$. If Ω is not empty, the dual sub-problem can be either unbounded or feasible for any arbitrary choice of $\bar{y}$. If the dual sub-problem is unbounded, given the set of extreme rays $\mathbb{Q}$ of Ω, there exists a direction of unboundedness $\{\bar{q}_1, \bar{q}_2\} \in \mathbb{Q}$ for which $\bar{q}_1^{\mathrm{T}}(b - B\bar{y}) + \bar{q}_2^{\mathrm{T}}(d - E\bar{y}) < 0$. To restrict movement in this direction, the following feasibility cut is added:

$$\bar{q}_1^{\mathrm{T}}(b - By) + \bar{q}_2^{\mathrm{T}}(d - Ey) \geq 0 \tag{2-19}$$

The extreme rays $\bar{q}_1$ and $\bar{q}_2$ can be computed by finding a feasible solution for the following problem:

$$q_1^{\mathrm{T}}(b - B\bar{y}) + q_2^{\mathrm{T}}(d - E\bar{y}) < 0 \tag{2-20}$$
$$q_1^{\mathrm{T}} A + q_2^{\mathrm{T}} D = 0 \tag{2-21}$$
$$q_2 \geq 0 \tag{2-22}$$

If, on the other hand, a feasible solution $\{\bar{u}_1, \bar{u}_2\} \in \mathbb{E}$ is found to the sub-problem, with $\mathbb{E}$ being the set of extreme points of Ω, the upper bound U_{ub} is updated such that it is always equal to the best current solution of the sub-problem:
$U_{\mathrm{ub}} = \min\left(U_{\mathrm{ub}}, f^{\mathrm{T}}\bar{y} + \bar{u}_1^{\mathrm{T}}(b - B\bar{y}) + \bar{u}_2^{\mathrm{T}}(d - E\bar{y})\right)$. In addition, the following optimality cut is generated:

$$\bar{u}_1^{\mathrm{T}}(b - By) + \bar{u}_2^{\mathrm{T}}(d - Ey) \leq \eta \tag{2-23}$$

Finally, the master problem is formulated as follows:

$$\min_{y,\eta} \quad f^{\mathrm{T}} y + \eta \tag{2-24}$$
$$\text{subject to} \quad \bar{q}_1^{\mathrm{T}}(b - By) + \bar{q}_2^{\mathrm{T}}(d - Ey) \geq 0 \tag{2-25}$$
$$\bar{u}_1^{\mathrm{T}}(b - By) + \bar{u}_2^{\mathrm{T}}(d - Ey) \leq \eta \tag{2-26}$$
$$Gy = g \tag{2-27}$$
$$Fy \leq e \tag{2-28}$$
$$y \in \mathbb{Y}^{n_2} \tag{2-29}$$
$$\eta \in \mathbb{R} \tag{2-30}$$

The solution $\bar{y}$ to the master problem is used to update the lower bound U_{lb}:
$U_{\mathrm{lb}} = \max\left(U_{\mathrm{lb}}, f^{\mathrm{T}}\bar{y} + \eta\right)$, and to solve the dual sub-problem of the next iteration of the algorithm.

The Benders decomposition algorithm iterates between solving the dual sub-problem and the master problem, using the solution $\bar{y}$ for the dual sub-problem and the feasibility cuts and optimality cuts generated by the solution of the dual sub-problem as constraints for the master problem. The algorithm terminates when the optimality gap falls below a specified threshold, i.e., $U_{\mathrm{ub}} - U_{\mathrm{lb}} \leq \alpha$. The threshold value α should be a small positive number to obtain an accurate solution. The solution is globally optimal if the difference between the upper and lower bound is zero.

2-3-2 Enhancement strategies

The classical Benders decomposition approach may pose challenges regarding computing time and memory requirements [38,41]. The main potential drawbacks of classical Benders decomposition include erratic behavior of primal solutions, slow convergence towards the end of the algorithm, and the presence of equivalent solutions resulting in unchanging upper bounds [43]. To address these challenges, extensive research has been conducted to reduce the solution time of the Benders decomposition algorithm, typically focused on two main areas: enhancing the quality of generated solutions and cuts to minimize the number of iterations needed, or optimizing the solution procedure for both the master problem and sub-problem in each iteration, reducing the time required for each iteration. A four-dimensional taxonomy capturing these factors was identified in [43].

The *decomposition strategy* refers to the approach used to partition the problem into the master problem and the sub-problem. In the classical decomposition, the master problem does not consider the linking constraints or the non-complicating variables. In a modified decomposition, these constraints and variables are partially projected to retain an approximation of the projected terms in the master problem.

The *solution procedure* refers to the algorithms utilized for solving the master problem and the sub-problem. Common techniques include the BB algorithm for the master problem and the simplex method for the sub-problem. However, the master problem can pose computational challenges, as it involves a non-convex problem that grows in size with each iteration. As a result, alternative strategies can be employed to exploit the structure of the master problem or the sub-problem to improve computational efficiency.

The *solution generation* refers to the approach used to set trial values for the complicating variables. In classical Benders decomposition, a typical strategy is to solve the master problem without modification to obtain trial values for the complicating variables. The quality of these trial values directly impacts the number of iterations required, as they are used to generate cuts and bounds. To improve the quality of the solutions or expedite their generation, several methods have been suggested, including (1) utilizing alternative formulations, (2) enhancing the master problem formulation, and (3) employing heuristics to generate solutions or enhance the ones obtained autonomously.

The *cut generation* refers to the approach used for generating optimality and feasibility cuts. In classical Benders decomposition, the regular sub-problem obtained from the decomposition is solved to generate cuts. However, alternative methods can be employed — such as reformulating the sub-problem or solving auxiliary sub-problems — to strengthen traditional feasibility and optimality cuts or generate additional cuts to reduce the number of iterations required.

ϵ-optimal approach

To reduce the computation time of the master problem, [23] proposed a variant of Benders decomposition where the master problem stops as soon as a feasible instead of an optimal solution is found. The requirement for this feasible solution is that the objective function

value of the master problem must be below $U_{\text{ub}} - \epsilon$, where ϵ is a number between one and zero, and U_{ub} denotes the upper bound. Since there is a finite number of dual solutions for the sub-problem, and each dual solution must be improved by at least ϵ in each iteration, the ϵ-optimal approach is guaranteed to converge to an optimum in a finite number of steps, as the optimal value is bounded below.

The motivation behind this algorithm is that optimizing the master problem each iteration, especially in the early iterations, may not be efficient, as the master problem lacks information about the optimization problem in the beginning and requires multiple Benders cuts before this information is effectively incorporated. In addition, finding the optimal solution to the master problem becomes more complex with each iteration, as a Benders cut is added with each iteration. By focusing on finding a feasible solution instead of an optimal solution, the master problem turns into a feasibility problem, which is often easier to solve than an optimization problem. A potential drawback of this algorithm is that it may require more iterations than the classical Benders decomposition approach, as the non-optimal results of the master problem are likely to lead to non-optimal Benders cuts.

2-4 Conclusions

This chapter has provided a comprehensive overview of various aspects of railway traffic management, including the various aspects of railway operations planning, the types of models commonly used, and the mathematical programming algorithms used in railway traffic management problems.

The model that is used in this thesis can be used to derive a MILP problem. MILP problems are typically classified as NP-hard problems, which means advanced mathematical programming techniques might be required to overcome computational challenges. After a few potential solution approaches were discussed, Benders decomposition was chosen as the optimization technique for this thesis, highlighting its ability to find high-quality solutions with relative ease for large-scale MILP problems.

Benders decomposition was introduced and discussed in detail, in addition to possible enhancement strategies to the classical Benders decomposition approach. One of these methods, the ϵ-optimal Benders decomposition, was introduced and will be applied in this thesis.

Chapter 3

Benders decomposition for train departure frequency optimization

This chapter presents two Benders decomposition-based optimization approaches that will be applied to the passenger absorption model introduced in [32]. Section 3-1 presents the classical Benders decomposition approach, while Section 3-2 presents the ϵ-optimal Benders decomposition approach. Section 3-3 concludes this chapter.

3-1 Classical Benders decomposition

Benders decomposition [5] is a widely used method for solving large-scale optimization problems involving continuous and discrete variables. The main idea of Benders decomposition is to divide an optimization problem into a master problem and a sub-problem, each of which can be solved independently. The master problem is formulated as a Mixed-Integer Linear Programming (MILP) which is used to determine the integer variables, while the sub-problem is formulated as a Linear Programming (LP) problem for which the integer variables are fixed. The solution of the master problem is used as input for the sub-problem. The sub-problem is formulated as a dual problem using duality theory. The dual sub-problem can be feasible and bounded, feasible but unbounded, or infeasible. Depending on the feasibility and boundedness of the dual sub-problem, so-called Benders cuts are added to the master problem.

Suppose the dual sub-problem is feasible and bounded. In that case, the resulting extreme points (the optimal dual solutions) are used to generate a Benders cut — called an optimality cut — which is added to the master problem. If the dual sub-problem is unbounded, a set of extreme rays (the dual solutions leading to unboundedness) is used to generate a feasibility cut — the other possible Benders cut — which is added to the master problem. Together, the optimality and feasibility cuts define the feasible space and the projected costs of the optimization problem. The process of solving the dual sub-problem, generating Benders cuts, and solving the master problem is repeated iteratively until stopping criteria apply. If the

solution of the dual sub-problem is equal to the solution of the master problem, this solution is globally optimal.

This section applies the classical Benders decomposition approach to the optimization problem derived from the passenger absorption model introduced in [32]. The passenger absorption model can explicitly include time-varying Origin-Destination (OD) passenger demands. Furthermore, the model provides a good balance between solution quality and computational complexity. We briefly introduce the model and the corresponding optimization problem below, and for more details on the model, we refer to [32]. In the passenger absorption model, the planning time window is divided into several periods, and passenger OD demands are assumed constant in each period. The total time of passengers within a given planning time window is estimated as follows:

$$\begin{aligned} J_{\text{time}} = & \sum_{k=k_0}^{k_0+N-1} \sum_{p \in P} \Big(n_p(k)T + n_p^{\text{depart}}(k)\bar{r}_p + n_p^{\text{arr,tra}}(k)\theta_p^{\text{trans}} \Big) \\ & + \sum_{p \in P} n_p(k_0+N)T, \end{aligned} \tag{3-1}$$

where N denotes the number of periods in the planning time window; P is the set of all platforms in the metro network; T is the length of a period; $n_p(k)$ denotes the number of passenger waiting at platform p at the start of period k; $n_p^{\text{depart}}(k)$ represents the number of passenger departing from platform p during period k; $n_p^{\text{arr,tra}}(k)$ denotes the number of passengers arriving at platform p with the intention of transferring to another platform during period k; and θ_p^{trans} is the average travel time for passengers transferring from platform p. In the metro network, trains travel a predetermined route, stopping at every platform. The average travel time for a train departing from platform p to the next platform on its route is denoted as $\bar{r}_p$. The operational costs of trains in the planning time window are estimated as follows:

$$J_{\text{cost}} = \sum_{k=k_0}^{k_0+N-1} \sum_{p \in P} f_p(k)\bar{E}_p, \tag{3-2}$$

where $f_p(k)$ is the train departure frequency at platform p during period k and $\bar{E}_p$ denotes the average operational costs associated with dispatching a train from platform p towards the

next platform on its route. The optimization problem is given as follows:

$$\min \quad J = J_{\text{time}} + \zeta J_{\text{cost}}, \tag{3-3a}$$

subject to

$$f_p(k) = \frac{T-\gamma_p}{T} l_p\left(k-\delta_p\right) + \frac{\gamma_p}{T} l_p\left(k-\delta_p-1\right), \tag{3-3b}$$

$$f_p(k) \leq f_p^{\max}, \tag{3-3c}$$

$$C_p(k) = f_p(k) C_{\max} - \sum_{m \in S} n_{p,m}^{\text{train}}(k), \tag{3-3d}$$

$$n_{p,m}(k+1) = n_{p,m}(k) + \lambda_{p,m}(k) T + n_{p,m}^{\text{arr,tra}}(k) - n_{p,m}^{\text{absorb}}(k), \tag{3-3e}$$

$$n_p^{\text{wait}}(k) = n_p(k) + \lambda_p(k) T + n_p^{\text{arr,tra}}(k), \tag{3-3f}$$

$$n_p^{\text{absorb}}(k) = \min(C_p(k), n_p^{\text{wait}}(k)), \tag{3-3g}$$

$$n_{p,m}^{\text{absorb}}(k) = \alpha_{p,m}(k) n_p^{\text{absorb}}(k), \tag{3-3h}$$

$$n_{p,m}^{\text{train}}(k) = \frac{T - \bar{r}_p^{\text{pla}}}{T} n_{\text{p}^{\text{pla}}(p,m)}^{\text{depart}}(k) + \frac{\bar{r}_p^{\text{pla}}}{T} n_{\text{p}^{\text{pla}}(p,m)}^{\text{depart}}(k-1), \tag{3-3i}$$

$$n_{p,\text{sta}(p)}^{\text{alight}}(k) = n_{p,m}^{\text{train}}(k), \tag{3-3j}$$

$$n_{p,m \in S/\{\text{sta}(p)\}}^{\text{alight}}(k) = n_{p,q,m}^{\text{trans}}(k), \tag{3-3k}$$

$$n_{p,m}^{\text{depart}}(k) = n_{p,m}^{\text{train}}(k) - n_{p,m}^{\text{alight}}(k) + n_{p,m}^{\text{absorb}}(k), \tag{3-3l}$$

$$n_{q,p,m}^{\text{trans}}(k) = \chi_{q,p,m}(k) n_{q,m}^{\text{train}}(k), \tag{3-3m}$$

$$n_{p,m}^{\text{arr,tra}}(k) = \sum_{q \in \text{sta}(p)} \Big(\frac{T-\theta_{q,p}^{\text{trans}}}{T} n_{q,p,m}^{\text{trans}}(k) + \frac{\theta_{q,p}^{\text{trans}}}{T} n_{q,p,m}^{\text{trans}}(k-1) \Big), \tag{3-3n}$$

$$k = k_0, k_0+1, ..., k_0+N-1,$$

where ζ is a weight used to balance both objectives; $l_p(k)$ denotes the train departure frequency of the starting platform of the line on which platform p lies; $\delta_p = \text{floor}\{\frac{\psi_p}{T}\}$ and $\gamma_p = \psi_p - \delta_p T$, with ψ_p denoting the average time for a train between departing from the starting platform of a line and departing from another platform p of that same line; $f_p^{\max}$ denotes the maximum train departure frequency of platform p; $C_p(k)$ represents the remaining capacity on a train at platform p during period k with $C_{\max}$ being the maximum capacity of a train; $n_{p,m}^{\text{train}}(k)$ is the number of passengers on board of trains at platform p with destination m during period k; $n_{p,m}(k)$ denotes the number of passenger waiting at platform p with destination m during period k; $\lambda_{p,m}(k)$ is the passenger arrival rate at platform p with destination m during period k; $n_{q,p,m}^{\text{trans}}(k)$ denotes the number of transferring passengers arriving at platform q to transfer to platform p with destination m during period k; $n_{p,m}^{\text{absorb}}(k)$ represents the number of passengers who board a train at platform p with destination m during period k; $n_{p,m}^{\text{wait}}(k)$ denotes the number of passengers waiting for a train at platform p with destination m during period k; and $n_{p,m}^{\text{alight}}(k)$ denotes the number of passengers alighting a train at platform p with destination m during period k. Parameter $\alpha_{p,m}(k)$ is the relative fraction of passengers that board a train at platform p whose destination is station m and parameter $\chi_{q,p,m}(k)$ is the relative fraction of passengers arriving at platform q with destination m, who will transfer from platform q to platform p. Both variables can be estimated using historical data. The set of platforms belong to the same station as platform p is denoted as $\text{sta}(p)$.

Eq. (3-3g) is a nonlinear constraint, which can be transformed into linear inequalities using the method in [56]. The transformation is described in Appendix A. For compactness, the linear inequalities are expressed as:

$$n_p^{\text{absorb}}(k) = z_p^{\text{wait}}(k) + C_p(k) - z_p^{\text{cap}}(k), \tag{3-4a}$$

$$E_{p,1}(k)\delta_p^{\text{absorb}}(k) + E_{p,2}(k)z_p^{\text{wait}}(k) \leq E_{p,3}(k)n_p^{\text{wait}}(k) + E_{p,4}(k), \tag{3-4b}$$

$$E_{p,5}(k)\delta_p^{\text{absorb}}(k) + E_{p,6}(k)z_p^{\text{cap}}(k) \leq E_{p,7}(k)C_p(k) + E_{p,8}(k), \tag{3-4c}$$

where $\delta_p^{\text{absorb}}(k)$ are auxiliary binary variables, and $z_p^{\text{wait}}(k)$ and $z_p^{\text{cap}}(k)$ are auxiliary continuous variables. By transforming the nonlinear function into linear inequalities, (3-3) is transformed into a MILP problem.

In this thesis, $l_p(k)$ and $\delta_p^{\text{absorb}}(k)$ are the so-called "complicating variables" according to the definition used in [5], as they are integer variables; both variables are fixed as $\bar{l}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$, respectively, for the sub-problem. Since T, γ_p, and δ_p are all parameters, it follows from (3-3b) that once $l_p(k)$ is given, $f_p(k)$ is also known. All remaining variables are continuous and can be derived by solving the following sub-problem:

$$\min \quad J_{\text{time}}, \tag{3-5a}$$

subject to

$$C_p(k) = f_p(k)C_{\max} - \sum_{m \in S} n_{p,m}^{\text{train}}(k), \tag{3-5b}$$

$$n_{p,m}(k+1) = n_{p,m}(k) + \lambda_{p,m}(k)T + n_{p,m}^{\text{arr,tra}}(k) - n_{p,m}^{\text{absorb}}(k), \tag{3-5c}$$

$$n_p^{\text{wait}}(k) = n_p(k) + \lambda_p(k)T + n_p^{\text{arr,tra}}(k), \tag{3-5d}$$

$$n_{p,m}^{\text{absorb}}(k) = \alpha_{p,m}(k)n_p^{\text{absorb}}(k), \tag{3-5e}$$

$$n_{p,m}^{\text{train}}(k) = \frac{T - \bar{r}_p^{\text{pla}}}{T} n_{\text{p}^{\text{pla}}(p,m)}^{\text{depart}}(k) + \frac{\bar{r}_p^{\text{pla}}}{T} n_{\text{p}^{\text{pla}}(p,m)}^{\text{depart}}(k-1), \tag{3-5f}$$

$$n_{p,\text{sta}(p)}^{\text{alight}}(k) = n_{p,m}^{\text{train}}(k), \tag{3-5g}$$

$$n_{p,m \in S/\{\text{sta}(p)\}}^{\text{alight}}(k) = n_{p,q,m}^{\text{trans}}(k), \tag{3-5h}$$

$$n_{p,m}^{\text{depart}}(k) = n_{p,m}^{\text{train}}(k) - n_{p,m}^{\text{alight}}(k) + n_{p,m}^{\text{absorb}}(k), \tag{3-5i}$$

$$n_{q,p,m}^{\text{trans}}(k) = \chi_{q,p,m}(k)n_{q,m}^{\text{train}}(k), \tag{3-5j}$$

$$n_{p,m}^{\text{arr,tra}}(k) = \sum_{q \in \text{sta}(p)} \Big(\frac{T - \theta_{q,p}^{\text{trans}}}{T} n_{q,p,m}^{\text{trans}}(k) + \frac{\theta_{q,p}^{\text{trans}}}{T} n_{q,p,m}^{\text{trans}}(k-1) \Big), \tag{3-5k}$$

$$n_p^{\text{absorb}}(k) = z_p^{\text{wait}}(k) + C_p(k) - z_p^{\text{cap}}(k), \tag{3-5l}$$

$$E_{p,1}(k)\bar{\delta}_p^{\text{absorb}}(k) + E_{p,2}(k)z_p^{\text{wait}}(k) \leq E_{p,3}(k)n_p^{\text{wait}}(k) + E_{p,4}(k), \tag{3-5m}$$

$$E_{p,5}(k)\bar{\delta}_p^{\text{absorb}}(k) + E_{p,6}(k)z_p^{\text{cap}}(k) \leq E_{p,7}(k)C_p(k) + E_{p,8}(k), \tag{3-5n}$$

$$k = k_0, k_0 + 1, ..., k_0 + N - 1,$$

Dual variables associated with the constraints are used to formulate a Lagrangian dual problem; $u_p^{\text{capacity}}(k)$ is associated with (3-5b), $u_{p,m}^{\text{number}}(k)$ is associated with (3-5c), $u_p^{\text{wait}}(k)$ is associated with (3-5d), $u_{p,m}^{\text{absorb}}(k)$ is associated with (3-5e), $u_{p,m}^{\text{train}}(k)$ is associated with (3-5f), $u_{p,\text{sta}(p)}^{\text{alight}}(k)$ is associated with (3-5g), $u_{p,m \in S/\{\text{sta}(p)\}}^{\text{alight}}(k)$ is associated with (3-5h), $u_{p,m}^{\text{depart}}(k)$ is associated with (3-5i), $u_{q,p,m}^{\text{trans}}(k)$ is associated with (3-5j), $u_{p,m}^{\text{arrive, trans}}(k)$ is associated with

(3-5k), $u_p^{\text{absorb}}(k)$ is associated with (3-5l), $u_p^{\text{wait, auxiliary}}(k)$ is associated with (3-5m), and $u_p^{\text{capacity, auxiliary}}(k)$ is associated with (3-5n).

The dual variables are used to formulate the dual sub-problem using duality theory. The characteristics of the corresponding constraints determine the domain of the dual variables. For example, suppose a constraint has the form $ax+by \leq c$; the dual variable associated with this constraint will have a domain of $[0,+\infty]$. On the other hand, the dual variable associated with a constraint of the form $ax+by=c$ will have an unrestricted domain. The objective function of the dual sub-problem is as follows:

$$\begin{aligned}\max\ J_{\text{dsp}} = \sum_{k=k_0}^{k_0+N-1}\sum_{p\in P}\sum_{m\in S}\Big(&u_p^{\text{capacity}}(k)C_{\max}\bar{f}_p(k) - u_{p,m}^{\text{number}}(k)\lambda_{p,m}(k)T - u_p^{\text{wait}}(k)\lambda_p(k)T\\ &+ u_p^{\text{wait, auxiliary}}(k)\left(E_{p,1}(k)\bar{\delta}_p^{\text{absorb}}(k) - E_{p,4}(k)\right)\\ &+ u_p^{\text{capacity, auxiliary}}(k)\left(E_{p,5}(k)\bar{\delta}_p^{\text{absorb}}(k) - E_{p,8}(k)\right)\Big),\end{aligned} \tag{3-6}$$

where J_{dsp} represents the objective function value of the dual sub-problem. The dual sub-problem is given as follows:

$$\max \quad J_{\text{dsp}} \tag{3-7a}$$

subject to

$$u_p^{\text{capacity}}(k) = u_p^{\text{absorb}}(k) - E_7 u_p^{\text{capacity, auxiliary}}(k), \tag{3-7b}$$

$$u_{p,m}^{\text{number}}(k) = T + u_{p,m}^{n}(k-1), \tag{3-7c}$$

$$u_{p,m}^{\text{absorb}}(k) = -u_{p,m}^{\text{number}}(k) + u_{p,m}^{\text{depart}}(k), \tag{3-7d}$$

$$u_p^{\text{wait}}(k) = -E_3 u_p^{\text{wait, auxiliary}}(k), \tag{3-7e}$$

$$u_{p,m}^{\text{train}}(k) = -u_p^{\text{capacity}}(k) + u_{p,m}^{\text{depart}}(k) + \sum_{q\in \text{sta}(p)} \chi_{q,p,m}(k) u_{q,p,m}^{\text{trans}}(k) + u_{p,\text{sta}(p)}^{\text{alight}}(k), \tag{3-7f}$$

$$u_{p,\text{sta}(p)}^{\text{alight}}(k) = -u_{p,m}^{\text{depart}}(k), \tag{3-7g}$$

$$u_{p,m\in S/\{\text{sta}(p)\}}^{\text{alight}}(k) = -u_{p,m}^{\text{depart}}(k), \tag{3-7h}$$

$$u_{p,m}^{\text{depart}}(k) = \bar{r}_p(k) + \frac{T-\bar{r}_{\text{p}^{\text{pla}}(p)}}{T} u_{p,m}^{\text{train}}(k) + \frac{\bar{r}_{\text{p}^{\text{pla}}}(k)}{T} u_{p,m}^{\text{train}}(k-1), \tag{3-7i}$$

$$u_{q,p,m}^{\text{trans}}(k) = \theta_{q,p}^{\text{trans}} + u_{p,m\in S/\{\text{sta}(p)\}}^{\text{alight}}(k) + \frac{T-\theta_{q,p}^{\text{trans}}}{T} u_{p,m}^{\text{arrive,trans}}(k) + \frac{\theta_{q,p}^{\text{trans}}}{T} u_{p,m}^{\text{arrive,trans}}(k-1), \tag{3-7j}$$

$$u_{p,m}^{\text{arrive,trans}}(k) = u_{p,m}^{\text{number}}(k), \tag{3-7k}$$

$$u_p^{\text{absorb}}(k) = \sum_{m\in S} \alpha_{p,m}(k) u_{p,m}^{\text{absorb}}(k), \tag{3-7l}$$

$$u_p^{\text{wait, auxiliary}}(k) = -E_{p,2}(k) u_p^{\text{absorb}}(k), \tag{3-7m}$$

$$u_p^{\text{capacity, auxiliary}}(k) = E_{p,6}(k) u_p^{\text{absorb}}(k) + E_{p,7}(k) u_p^{\text{absorb}}(k), \tag{3-7n}$$

$$u_p^{\text{capacity}}(k), ..., u_p^{\text{absorb}}(k) \in \mathbb{R}, \tag{3-7o}$$

$$u_p^{\text{wait, auxiliary}}(k), u_p^{\text{capacity, auxiliary}}(k) \geq 0, \tag{3-7p}$$

$$k = k_0, k_0+1, ..., k_0+N-1,$$

By duality theory, if a finite solution exists, the optimal value of the dual sub-problem is equal to the optimal value of the original problem for the given $\bar{l}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$ [7]. Therefore, if the dual sub-problem is feasible and bounded, the optimal value of the objective function provides an upper bound of the original optimization problem, denoted as U_{ub}. The upper bound of the ith iteration is computed as follows: $U_{\text{ub}}^i = \min\left(U_{\text{ub}}^{i-1}, \sum_{k=k_0}^{k_0+N-1}\sum_{p\in P}\zeta\bar{f}_p(k)\bar{E}_p + \bar{J}_{\text{dsp}}^i\right)$, $\forall i \in \{1,2,...\}$, where $\bar{J}_{\text{dsp}}^i$ represents the objective function value of the dual sub-problem of the ith iteration.

Constraints (3-7b):(3-7p) constitute polyhedron Ω, which represents the feasible space of the original optimization problem for a given $\bar{l}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$. We use $\mathbb{E}$ and $\mathbb{Q}$ to denote the sets of extreme points and rays of Ω, respectively. If there is an optimal and bounded solution $[\bar{u}_p^{\text{capacity}}(k), ..., \bar{u}_p^{\text{capacity, auxiliary}}(k)]$, then this solution is a vector of extreme points belonging to the set $\mathbb{E}$. The extreme points are used to add a constraint to the master problem, i.e., an optimality cut.

If the solution is unbounded above, by duality theory, the original optimization problem is infeasible [7]. In other words, the choice for $\bar{l}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$ does not satisfy the constraints of the original optimization problem if the dual sub-problem is unbounded above. The solution $\bar{l}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$ needs to be removed from the master problem; this can be done by computing a vector of extreme rays $[\bar{q}_p^{\text{capacity}}(k), ..., \bar{q}_p^{\text{capacity, auxiliary}}(k)] \in \mathbb{Q}$ for which the dual sub-problem is unbounded above, i.e., $\sum_{k=k_0}^{k_0+N-1} J_{\text{dsp}}(k) > 0$, and adding a constraint to the master problem such that the objective function of the dual sub-problem is never positive for the given vector of extreme rays; this constraint is called the feasibility cut.

Since polyhedron $\mathbb{Q}$ is independent of $\bar{f}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$, the original optimization problem is infeasible if the dual sub-problem is infeasible. The equations for the optimality and feasibility cuts are, respectively, as follows:

$$\begin{aligned} J_{\text{opt}} = \sum_{k=k_0}^{k_0+N-1}\sum_{p\in P}\sum_{m\in S}\Big(&\bar{u}_p^{\text{capacity}}(k)C_{\max}f_p(k) - \bar{u}_{p,m}^{\text{number}}(k)\lambda_{p,m}(k)T - \bar{u}_p^{\text{wait}}(k)\lambda_p(k)T \\ &+ \bar{u}_p^{\text{wait, auxiliary}}(k)\left(E_{p,1}(k)\delta_p^{\text{absorb}}(k) - E_{p,4}(k)\right) \\ &+ \bar{u}_p^{\text{capacity, auxiliary}}(k)\left(E_{p,5}(k)\delta_p^{\text{absorb}}(k) - E_{p,8}(k)\right)\Big), \\ \bar{u}_p^{\text{capacity}}(k), \bar{u}_p^{\text{number}}(k), \bar{u}_p^{\text{wait}}(k), \bar{u}_p^{\text{wait, auxiliary}}(k), \bar{u}_p^{\text{capacity, auxiliary}}(k) &\in \mathbb{E} \end{aligned} \tag{3-8}$$

$$\begin{aligned} J_{\text{feas}} = \sum_{k=k_0}^{k_0+N-1}\sum_{p\in P}\sum_{m\in S}\Big(&\bar{q}_p^{\text{capacity}}(k)C_{\max}f_p(k) - \bar{q}_{p,m}^{\text{number}}(k)\lambda_{p,m}(k)T - \bar{q}_p^{\text{wait}}(k)\lambda_p(k)T \\ &+ \bar{q}_p^{\text{wait, auxiliary}}(k)\left(E_{p,1}(k)\delta_p^{\text{absorb}}(k) - E_{p,4}(k)\right) \\ &+ \bar{q}_p^{\text{capacity, auxiliary}}(k)\left(E_{p,5}(k)\delta_p^{\text{absorb}}(k) - E_{p,8}(k)\right)\Big), \\ \bar{q}_p^{\text{capacity}}(k), \bar{q}_p^{\text{number}}(k), \bar{q}_p^{\text{wait}}(k), \bar{q}_p^{\text{wait, auxiliary}}(k), \bar{q}_p^{\text{capacity, auxiliary}}(k) &\in \mathbb{Q} \end{aligned} \tag{3-9}$$

The extreme points are the optimal dual variables found by solving the dual sub-problem. The extreme rays can be obtained by finding a solution to a set of equations for which $\sum_{k=k_0}^{k_0+N-1} J_{\text{dsp}}(k) > 0$ for the given $\bar{l}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$. The set of equations solved to obtain extreme rays is given in Appendix B.

The master problem is constructed using the original constraints for variables $l_p(k)$, $f_p(k)$, $\delta_p^{\text{absorb}}(k)$; the optimality and feasibility cuts; and auxiliary variable $\eta \in \mathbb{R}$. The objective of the master problem is to minimize η, in addition to $\zeta f_p(k)\bar{E}_p$, as this term is not included in the objective of the dual sub-problem. The master problem is given as follows:

$$\min \qquad J_{\text{mas}} = \eta + \sum_{k=k_0}^{k_0+N-1} \sum_{p\in P} \zeta f_p(k)\bar{E}_p \tag{3-10a}$$

subject to

$$J_{\text{feas}} \leq 0, \tag{3-10b}$$

$$J_{\text{opt}} \leq \eta, \tag{3-10c}$$

$$f_p(k) \leq f_p^{\max}, \tag{3-10d}$$

$$f_p(k) = \frac{T-\gamma_p}{T} l_p\left(k-\delta_p\right) + \frac{\gamma_p}{T} l_p\left(k-\delta_p-1\right), \tag{3-10e}$$

$$\delta_p^{\text{absorb}}(k) \in \{0,1\}, \tag{3-10f}$$

$$l_p(k) \in \mathbb{Z}, \tag{3-10g}$$

$$\eta \in \mathbb{R} \tag{3-10h}$$

$$k = k_0, k_0+1, ..., k_0+N-1,$$

The optimality cuts guide the master problem to the optimal solution for $l_p(k)$ and $\delta_p^{\text{absorb}}(k)$, while the feasibility cuts ensure the feasibility of the solution. The feasible space of η is reduced with every added Benders cut; this means that — if an optimal solution is found — the objective function value of the master problem provides a lower bound on the solution of the original optimization problem. The lower bound is denoted as U_{lb}, with $U_{\text{lb}}^i = \max\left(U_{\text{lb}}^{i-1}, \bar{J}_{\text{mas}}^i\right)$, $\forall i \in \{1,2,...\}$, where $\bar{J}_{\text{mas}}^i$ represents the value of the objective function of the master problem for the ith iteration of the Benders decomposition algorithm.

The optimal solution $\bar{l}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$ is then used to solve the dual sub-problem in the next iteration until the difference between the lower bound and the upper bound is below a certain threshold, i.e., $U_{\text{ub}} - U_{\text{lb}} \geq \alpha$, where α is a positive number. If the difference between the upper and the lower bound is zero, by duality, the solution is globally optimal. The classical Benders decomposition algorithm is presented in Algorithm 1. A flowchart is given in Fig. 3-1.

Algorithm 1: Classical Benders decomposition-based train departure frequency optimization algorithm

Input: α, ζ, N, P, S, $\theta_{q,p}^{\text{trans}}$, $\bar{E}_p$, $\bar{r}_p$, and ψ_p; estimated values for $\chi_{q,p,m}(k)$, $\alpha_{p,m}(k)$ and $\lambda_{p,m}(k)$
Set initial values:
$U_{\text{ub}}^0 \leftarrow \infty$, $U_{\text{lb}}^0 \leftarrow -\infty$, $\bar{f}_p(k) \leftarrow 0$, $\bar{\delta}_p^{\text{absorb}}(k) \leftarrow 0$, $i \leftarrow 0$
Output: $l_p(k)$, $f_p(k)$, and $\delta_p^{\text{absorb}}(k)$
while $U_{\text{ub}}^i - U_{\text{lb}}^i \geq \alpha$ **do**
  $i \leftarrow i + 1$
  Solve (3-7) using $\bar{l}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$
  if (3-7) is feasible and bounded **then**
    Obtain $\bar{J}_{\text{dsp}}$ and extreme points $[\bar{u}_p^{\text{capacity}}(k), ..., \bar{u}_p^{\text{capacity, auxiliary}}(k)] \in \mathbb{E}$
    Update upper bound:
    $U_{\text{ub}}^i \leftarrow \min\left(U_{\text{ub}}^{i-1}, \sum_{k=k_0}^{k_0+N-1} \sum_{p \in P} \zeta \bar{f}_p(k) \bar{E}_p + \bar{J}_{\text{dsp}}^i\right)$
    Add optimality cut (3-8) using extreme points
  else if (3-7) is feasible but unbounded **then**
    Compute extreme rays $[\bar{q}_p^{\text{capacity}}(k), ..., \bar{q}_p^{\text{capacity, auxiliary}}(k)] \in \mathbb{Q}$
    Add feasibility cut (3-9) using extreme rays
  else if (3-7) is infeasible **then**
    Model is infeasible and algorithm is terminated
  end if
  Solve (3-10) to obtain new $\bar{f}_p(k)$, and $\bar{\delta}_p^{\text{absorb}}(k)$
  Update lower bound:
  $U_{\text{lb}}^i \leftarrow \max\left(U_{\text{lb}}^{i-1}, \bar{J}_{\text{mas}}^i\right)$
end while

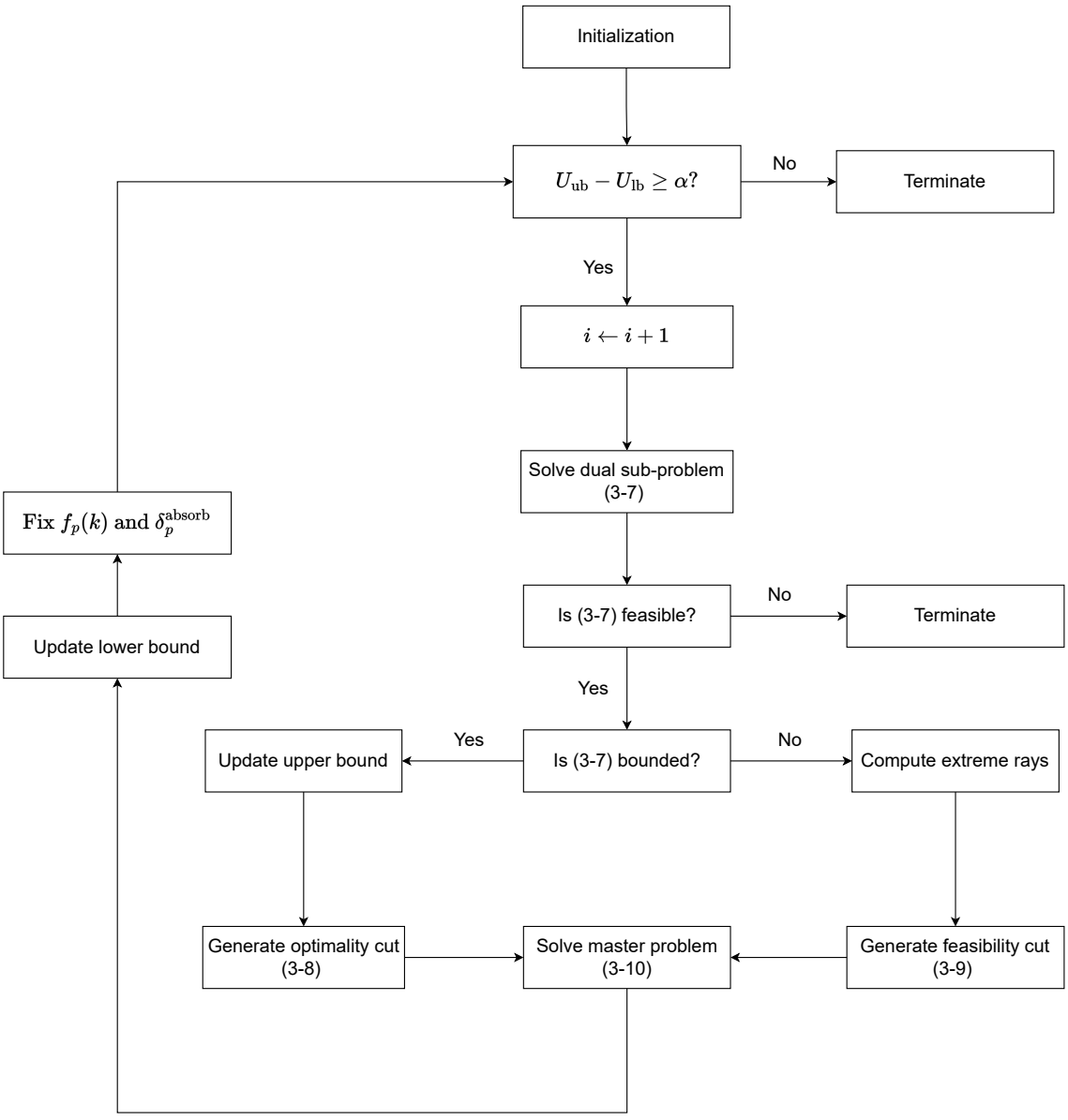


Figure 3-1: Visualization classical Benders decomposition algorithm

3-2 ϵ-optimal Benders decomposition

The classical Benders decomposition algorithm may face difficulties when many iterations are required before a solution is found, as the master problem increases in size and complexity with every added Benders cut, resulting in a high computation time. To reduce the computation time of the master problem, this section applies the ϵ-optimal Benders algorithm introduced in [23]. The new constraints of the master problem are given as follows:

$$\eta + \sum_{k=k_0}^{k_0+N-1} \sum_{p \in P} \zeta f_p(k) \bar{E}_p \leq U_{\text{ub}}(1-\epsilon) \tag{3-11a}$$

$$J_{\text{feas}} \leq 0 \tag{3-11b}$$

$$J_{\text{opt}} \leq \eta \tag{3-11c}$$

$$f_p(k) \leq f_p^{\max} \tag{3-11d}$$

$$f_p(k) = \frac{T-\gamma_p}{T} l_p\left(k-\delta_p\right) + \frac{\gamma_p}{T} l_p\left(k-\delta_p-1\right) \tag{3-11e}$$

$$\delta_p^{\text{absorb}}(k) \in \{0,1\}, \tag{3-11f}$$

$$l_p(k) \in \mathbb{Z}, \tag{3-11g}$$

$$\eta \in \mathbb{R}, \tag{3-11h}$$

$$k = k_0, k_0+1, ..., k_0+N-1$$

Instead of finding an optimal solution, the new master problem seeks a feasible solution for which the objective function of the master problem is at least $100\epsilon\%$ lower than the current upper bound U_{ub}. The master problem has therefore turned into a feasibility problem, which is generally easier to solve than an optimization problem. The feasible solution to the master problem is used for the dual sub-problem of the next iteration, similar to the classical Benders decomposition. Since the solution to the master problem is not optimal, the master problem does not provide a valid lower bound. The ϵ-optimal Benders algorithm terminates when the master problem cannot produce a feasible solution. The algorithm is guaranteed to terminate in a finite number of steps, as there is a finite number of optimal dual solutions for the sub-problem, and each optimal dual solution must improve the classical master problem objective function. A potential drawback of the ϵ-optimal Benders algorithm is that it may require more iterations than the classical Benders decomposition algorithm, as the non-optimal solutions to the master problem may also lead to non-optimal Benders cuts. Two versions of the ϵ-optimal Benders algorithm will be used: one with a constant value for ϵ and one where the value for ϵ decreases until a minimum value is reached. The ϵ-optimal Benders algorithm with constant ϵ is shown in Algorithm 2. A flowchart is given in Fig. 3-2.

Algorithm 2: ϵ-optimal Benders decomposition-based train departure frequency optimization algorithm

Input: α, ζ, N, P, S, $\theta_{q,p}^{\text{trans}}$, $\bar{E}_p$, $\bar{r}_p$, and ψ_p; estimated values for $\chi_{q,p,m}(k)$, $\alpha_{p,m}(k)$ and $\lambda_{p,m}(k)$
Set initial values:
$U_{\text{ub}}^0 \leftarrow \infty$, $\bar{f}_p(k) \leftarrow 0$, $\bar{\delta}_p^{\text{absorb}}(k) \leftarrow 0$, $i \leftarrow 0$
Output: $f_p(k)$ and $\delta_p^{\text{absorb}}(k)$
while $U_{\text{ub}}^i \geq \alpha$ **do**
  $i \leftarrow i + 1$
  Solve (3-7) using $\bar{l}_p(k)$ and $\bar{\delta}_p^{\text{absorb}}(k)$
  if (3-7) is feasible and bounded **then**
    Obtain $\bar{J}_{\text{dsp}}$ and extreme points $[\bar{u}_p^{\text{capacity}}(k), ..., \bar{u}_p^{\text{capacity, auxiliary}}(k)] \in \mathbb{E}$
    Update upper bound:
    $U_{\text{ub}}^i \leftarrow \min\left(U_{\text{ub}}^{i-1}, \sum_{k=k_0}^{k_0+N-1} \sum_{p \in P} \zeta \bar{f}_p(k) \bar{E}_p + \bar{J}_{\text{dsp}}^i\right)$
    Add optimality cut (3-8) using extreme points
  else if (3-7) is feasible but unbounded **then**
    Compute extreme rays $[\bar{q}_p^{\text{capacity}}(k), ..., \bar{q}_p^{\text{capacity, auxiliary}}(k)] \in \mathbb{Q}$
    Add feasibility cut (3-9) using extreme rays
  else if (3-11) is infeasible **then**
    Model is infeasible and algorithm is terminated
  end if
  Solve (3-11)
  if (3-11) is feasible **then**
    Obtain new $\bar{f}_p(k)$, and $\bar{\delta}_p^{\text{absorb}}(k)$
  else if (3-11) is infeasible **then**
    Break while loop
  end if
end while

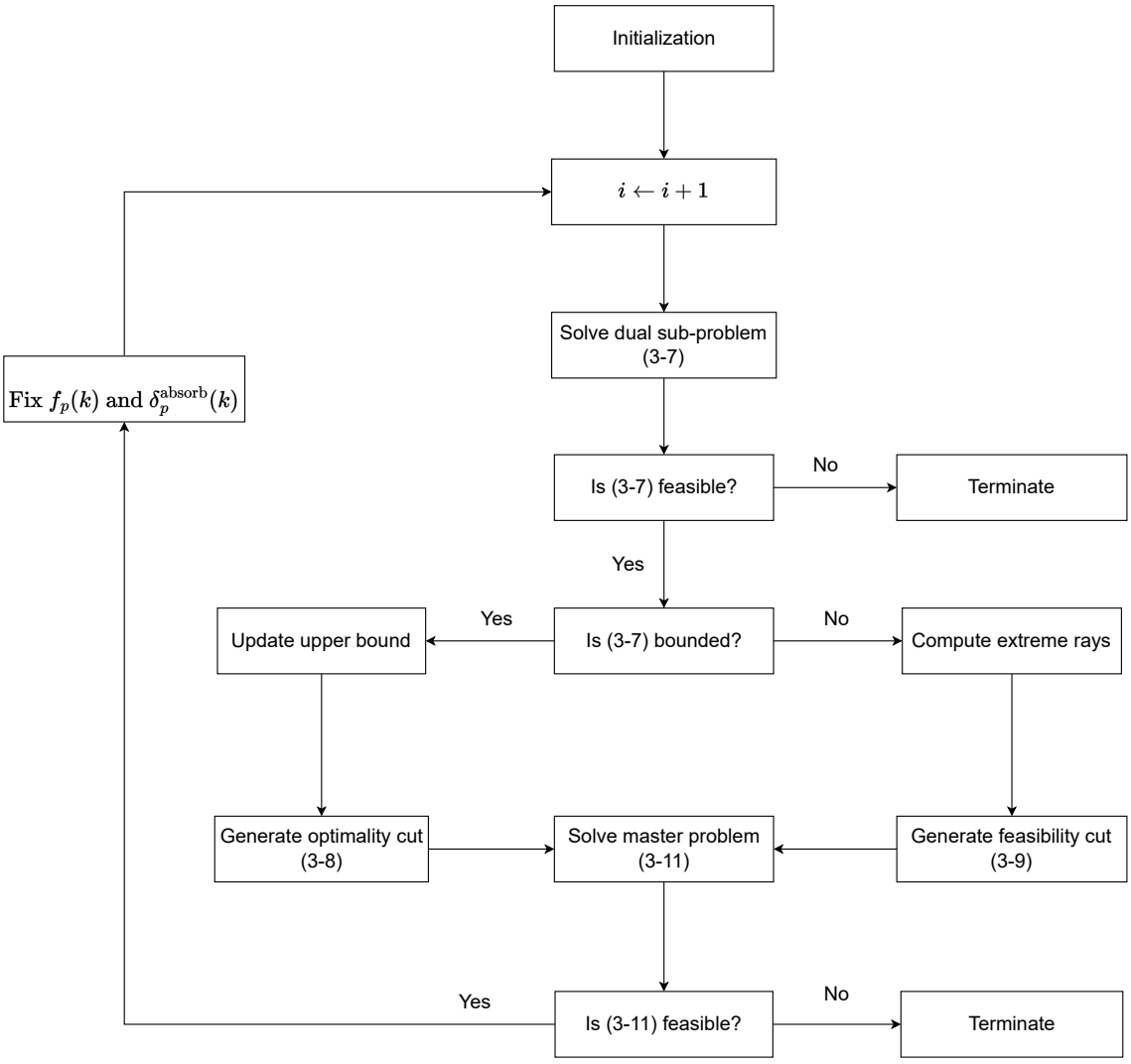


Figure 3-2: Visualization ϵ-optimal Benders decomposition algorithm

3-3 Conclusions

This thesis uses the passenger absorption model introduced in [32] to optimize train departure frequencies in a metro network. A MILP problem is derived from the passenger absorption model, for which two Benders decomposition-based algorithms are applied, i.e., the classical Benders decomposition algorithm [5] and the ϵ-optimal Benders decomposition algorithm [23]. The two decomposition methods divide the original optimization problem into a master problem and a dual sub-problem, solving both problems iteratively. The master problem is used to determine the integer and binary variables, while the dual sub-problem is used to determine the other (continuous) variables. The ϵ-optimal Benders decomposition algorithm adds a constraint to the master problem and treats the master problem as a feasibility problem instead of an optimization problem, aiming to accelerate the algorithm by reducing the computational complexity of the master problem.

Chapter 4

Case study

This chapter shows the efficiency of the Benders decomposition-based approaches for train departure frequency optimization by comparing them against a state-of-the-art solver. A simulation-based case study is performed on a grid metro network with time-varying Origin-Destination (OD) passenger demands. The Benders decomposition-based approaches are evaluated based on the objective function value and the solution time. Section 4-1 provides the details of the simulation-based case study. Section 4-2 gives the results of the simulation-based case study. Section 4-3 concludes this chapter.

4-1 Set-up

This section provides the details of the simulation-based case study. First, the metro network that was modeled for the case study is shown. Next, the relevant parameters of the metro network are given. Finally, the relevant computer specifications used for the simulations are given.

The metro network used for the case study is shown in Fig. 4-1, consisting of twenty-one stations, sixty platforms, and six bidirectional lines. The number on top of each link in Fig. 4-1 represents the average travel time between two stations and is used to determine the parameters $\bar{r}_p$ and ψ_p. Parameters $\bar{r}_p$ and ψ_p are assumed to be similar for both directions of a line. Time-varying OD passenger demands are used for the case study.

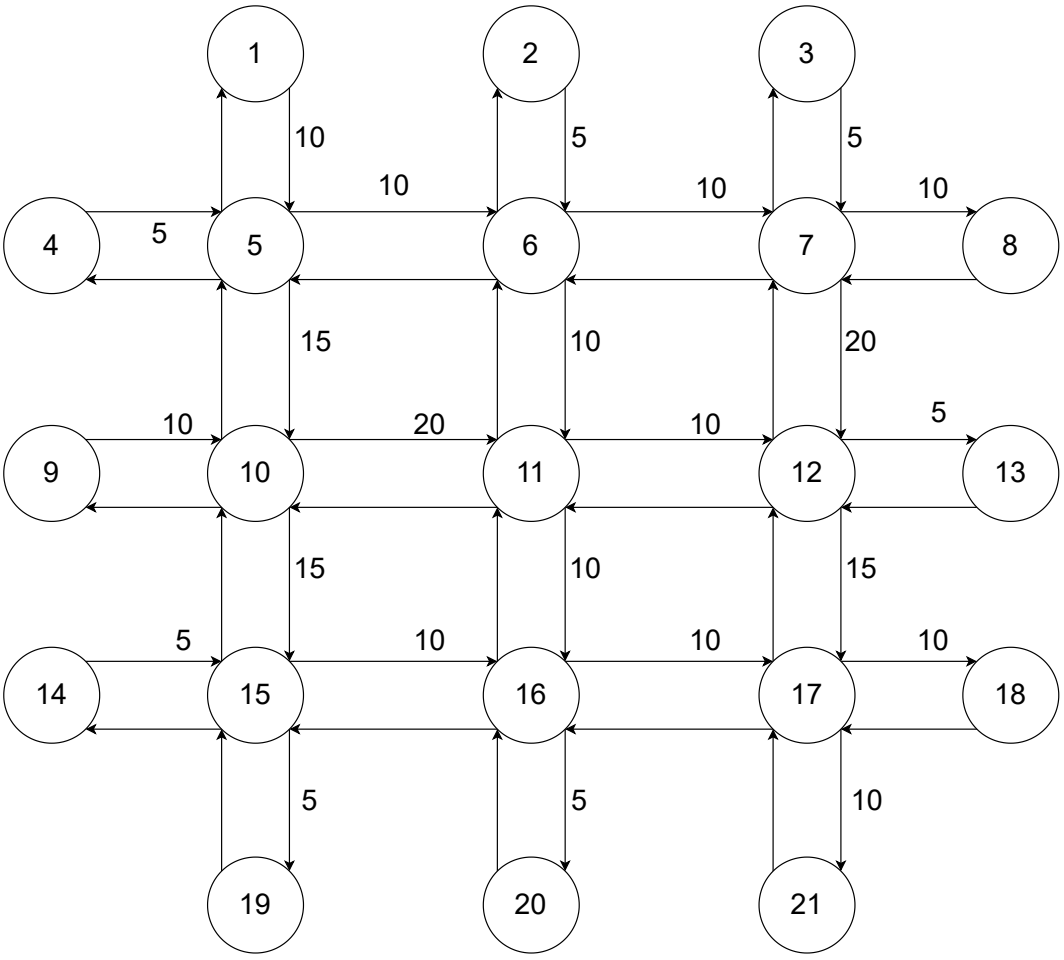


Figure 4-1: Railway operations planning

The average transfer time between two platforms $\theta_{q,p}^{\text{trans}}$ is assumed equal for each combination of platforms q and p that belong to the same station. Each platform has a maximum train frequency $f_p^{\max}$, and each train has a maximum train capacity $C_{\max}$. The cost of a train run $\bar{E}_p$ depends on its associated travel time. The two objectives of the optimization problem, i.e., the time spent by passengers in the metro network and the operational costs, are balanced with weight ζ. The classical Benders decomposition algorithm terminates when the difference between the upper and lower bound is smaller than α. Table 4-1 gives an overview of all parameters.

Table 4-1: Comparison of different methods for constant passenger demand

Parameter	Value
Stop criterion α	1
Transfer time $\theta_{q,p}^{\text{trans}}$	1 [min]
Capacity $C_{\max}$	2000 passengers
Train cost $\bar{E}_p$	$2 \cdot \bar{r}_p$
Max departure frequency $f_p^{\max}$	20
Weight ζ	1000

In the passenger absorption model — the model used in this thesis — $\alpha_{p,m}(k)$ represents the relative fraction of passengers that board a train at platform p whose destination is station m during period k and $\chi_{q,p,m}(k)$ represents the relative fraction of passengers arriving at platform q to transfer to platform p with destination m during period k; these parameters can be estimated using historical data. Since we use a fictional metro network and do not have historical data available, parameters $\alpha_{p,m}(k)$ and $\chi_{q,p,m}(k)$ are computed by assuming that passengers will always choose the shortest path to their destination in terms of time spent in the rail network. If multiple routes are equally long in travel time, we assume that an equal number of passengers would choose each different route.

All the simulations are conducted using Matlab R2021a on a MacBook Pro 2017, which has a 2.3 GHz Dual-core Intel Core i5 processor and 8GB of RAM. The version of `gurobi` used is 9.5.2 build v9.5.2rc0 (mac64[x86]).

4-2 Case study

This section evaluates the Benders decomposition-based algorithms in a simulation-based case study. State-of-the-art solver `gurobi` is used as a benchmark. All methods are compared based on the objective function value and the computation time. Additional information is shown for the Benders decomposition-based algorithms, i.e., the number of iterations, the cumulative computation time of the dual sub-problem, the cumulative computation time of computing the extreme rays, and the cumulative computation time of the master problem.

First, the optimization problem is solved using `gurobi`. The computation time of `gurobi` only includes the part of the computation time exclusive to solving the problem using `gurobi`.

Next, simulations are run for the classical Benders decomposition algorithm. The classical Benders decomposition algorithm consists of three parts, i.e., (1) the dual sub-problem, (2) computing extreme rays when the dual sub-problem is unbounded, and (3) the master problem. All individual optimization problems are solved using `gurobi`. The total computation time consists of solving the dual sub-problem and master problem, updating the upper and lower bound, generating optimality and feasibility cuts, and obtaining extreme rays when necessary.

The ϵ-optimal Benders algorithm turns the master problem into a feasibility problem instead of an optimization problem, with the added constraint that the solution must be below $U_{\text{ub}}(1-\epsilon)$, where U_{ub} denotes the upper bound. In other words, the master problem no longer attempts to find an optimal solution for a given set of dual variables. Instead, the only requirement of the master problem is that the objective function value is below a specific factor of the upper bound. An essential parameter for the ϵ-optimal Benders decomposition algorithm is the value chosen for ϵ. The value for ϵ represents the acceptable optimality gap. For example, if ϵ is set as 0.2 and the algorithm terminates, in the worst case, the solution is 20% lower than the solution found by the ϵ-optimal Benders algorithm. Therefore, the value for ϵ must not be set too high. On the other hand, a low value for ϵ might result in a high number of iterations.

Simulations are run for five different ϵ-optimal Benders decomposition algorithms. The first three use a constant value for ϵ; the first uses a constant value of $\epsilon = 0.01$ for all iterations, the second uses a constant value of $\epsilon = 0.05$ for all iterations, and the third uses a constant value of $\epsilon = 0.1$ for all iterations. The three ϵ-optimal Benders decomposition algorithms with constant ϵ are denoted as ϵ-Benders ($\epsilon = 0.01$), ϵ-Benders ($\epsilon = 0.05$), and ϵ-Benders ($\epsilon = 0.1$), respectively.

Next, two ϵ-optimal Benders decomposition algorithms are used where the value of ϵ changes for each iteration; the two algorithms are denoted as ϵ-Benders ($\epsilon_1(i)$), and ϵ-Benders ($\epsilon_2(i)$), respectively, with $\epsilon_1(i) = \max(0.1{\cdot}0.99^i, 0.02)$, $\forall i \in \{1, 2, ...\}$, and $\epsilon_2(i) = \max(0.1{\cdot}0.985^i, 0.01)$, $\forall i \in \{1, 2, ...\}$, where i denotes the iterations of the ϵ-optimal Benders decomposition algorithm. Hence, ϵ starts at 0.1 for both algorithms and decreases with each iteration until a minimum value is reached.

Simulations are run for different planning time windows, i.e., for three hours ($N = 3$), for four

hours ($N = 4$), for five hours ($N = 5$), and for six hours ($N = 6$). The time limit for solving the optimization problem is set as three hours for all optimization approaches. The results for `gurobi`, the classical Benders decomposition algorithm, and the five ϵ-optimal Benders decomposition algorithms can be seen in table 4-2. In the table, N.A. indicates no solution was found within three hours. In the column between the objective function value and the computation time, the relative error is shown in terms of the objective function value relative to the classical Benders decomposition algorithm. The following equation is used to compute the error:

$$\text{Error} = \frac{J_{\text{Method}} - J_{\texttt{gurobi}}}{J_{\text{Method}}} \cdot 100\%, \tag{4-1}$$

where $J_{\texttt{gurobi}}$ indicates the objective function value obtained by `gurobi` and J_{Method} represents the objective function value obtained by the method for which the relative error is computed. If `gurobi` cannot find a solution within three hours, the relative error cannot be computed; this is indicated in the table as N.B.

4-2-1 Results

The performance of all methods is given in Table 4-2, showing the objective function value, the relative error in the objective function value compared to `gurobi`, and the computation time.

Solver `gurobi` can find a solution within three hours for $N = 3$ and $N = 4$, i.e., when the problem size is relatively small. The classical Benders decomposition algorithm obtains the same solution as `gurobi` for $N = 3$ and $N = 4$. If the difference between the upper and the lower bound of the classical Benders decomposition algorithm is zero, then the solution of the Benders decomposition algorithm is globally optimal. Since the classical Benders decomposition algorithm is terminated when the difference between the upper and lower bound is below one, the solution that is found is either globally optimal or very close to globally optimal, which explains why there is no error compared to `gurobi`. The relative error cannot be computed for $N = 5$ and $N = 6$, as `gurobi` cannot find a solution within three hours.

The ϵ-optimal Benders decomposition algorithm with ϵ set as 0.01 has no error for $N = 3$ and $N = 4$ compared to `gurobi`, and a small error compared to the classical Benders decomposition algorithm when $N = 5$. The ϵ-optimal Benders decomposition algorithms with ϵ set as 0.05 and 0.1 have more significant errors in the objective function value. The ϵ-optimal Benders decomposition algorithm terminates when there is no feasible solution to the master problem for which the objective function value is $\epsilon\%$ lower than the upper bound; this means that the maximum error relative to the classical Benders decomposition algorithm in the objective function values is $\epsilon\%$, which explains the difference in the relative errors.

The two ϵ-optimal Benders decomposition algorithms with varying ϵ generally perform better in terms of the objective function compared to the ϵ-optimal Benders decomposition algorithms with ϵ set as 0.05 and 0.1. The larger the problem size, the more significant the difference in objective function value between these methods. When the problem size increases, more iterations are required, resulting in ϵ reaching its minimum value during the final iterations for the ϵ-optimal Benders decomposition algorithms with varying ϵ. The lower value for ϵ during the final iterations ensures a more accurate solution.

The computation time of all methods is shown in the last column of Table 4-2. While `gurobi` finds the solution in a relatively short amount of time (around 2 minutes) when the problem size is small ($N = 3$), the computation time increases quickly when the problem size increases, as it takes `gurobi` close to two hours when $N = 4$. When the problem size increases further, `gurobi` cannot find a solution within three hours.

The classical Benders decomposition algorithm finds the solution much faster compared to `gurobi` when the problem size increases. Nevertheless, the classical Benders decomposition algorithm also requires a computation time that is too high for real-time applications when the problem size increases.

Table 4-2: Comparison of different methods. N.A. indicates that no solution was found after three hours; N.B indicates the benchmark method (`gurobi`) was not able to find a solution after three hours, which means the relative error cannot be computed; and Error denotes the relative error of a method compared to `gurobi`

N	Method	Objective function value	Error [%]	Computation time [s]
3	`gurobi`	$2.82 \cdot 10^5$		112.0
	Classical	$2.82 \cdot 10^5$	0	145.8
	ϵ-Benders ($\epsilon = 0.01$)	$2.83 \cdot 10^5$	0.35	207.0
	ϵ-Benders ($\epsilon = 0.05$)	$2.85 \cdot 10^5$	1.06	172.3
	ϵ-Benders ($\epsilon = 0.1$)	$2.86 \cdot 10^5$	1.42	161.5
	ϵ-Benders ($\epsilon_1(i)$)	$2.95 \cdot 10^5$	4.61	133.6
	ϵ-Benders ($\epsilon_2(i)$)	$2.82 \cdot 10^5$	0	159.5
4	`gurobi`	$4.00 \cdot 10^5$		6230.4
	Classical	$4.00 \cdot 10^5$	0	388.9
	ϵ-Benders ($\epsilon = 0.01$)	$4.00 \cdot 10^5$	0	358.4
	ϵ-Benders ($\epsilon = 0.05$)	$4.14 \cdot 10^5$	3.5	305.1
	ϵ-Benders ($\epsilon = 0.1$)	$4.14 \cdot 10^5$	3.5	291.3
	ϵ-Benders ($\epsilon_1(i)$)	$4.14 \cdot 10^5$	3.5	309.0
	ϵ-Benders ($\epsilon_2(i)$)	$4.09 \cdot 10^5$	2.25	337.3
5	`gurobi`	N.A.		N.A.
	Classical	$5.25 \cdot 10^5$	N.B.	7373.3
	ϵ-Benders ($\epsilon = 0.01$)	$5.28 \cdot 10^5$	N.B.	558.3
	ϵ-Benders ($\epsilon = 0.05$)	$5.48 \cdot 10^5$	N.B.	475.9
	ϵ-Benders ($\epsilon = 0.1$)	$5.48 \cdot 10^5$	N.B.	440.2
	ϵ-Benders ($\epsilon_1(i)$)	$5.34 \cdot 10^5$	N.B.	463.2
	ϵ-Benders ($\epsilon_2(i)$)	$5.27 \cdot 10^5$	N.B.	516.2
6	`gurobi`	N.A.		N.A.
	Classical	N.A.	N.B.	N.A.
	ϵ-Benders ($\epsilon = 0.01$)	$6.80 \cdot 10^5$	N.B.	1771.2
	ϵ-Benders ($\epsilon = 0.05$)	$7.07 \cdot 10^5$	N.B.	714.4
	ϵ-Benders ($\epsilon = 0.1$)	$7.14 \cdot 10^5$	N.B.	687.9
	ϵ-Benders ($\epsilon_1(i)$)	$6.77 \cdot 10^5$	N.B.	737.6
	ϵ-Benders ($\epsilon_2(i)$)	$6.77 \cdot 10^5$	N.B.	773.0

The ϵ-optimal Benders decomposition algorithms find the solution relatively quickly for all N. The computation time does not increase as fast with the problem size as for the classical Benders decomposition algorithm and `gurobi`. Moreover, the ϵ-optimal Benders decomposition algorithms with varying ϵ values are faster than the ϵ-optimal Benders decomposition algorithm with constant $\epsilon = 0.01$; the larger the problem size, the more significant the difference in computation time. Moreover, the ϵ-optimal Benders decomposition algorithms with varying ϵ values are not significantly slower than the ϵ-optimal Benders decomposition algorithm with constant $\epsilon = 0.05$ or $\epsilon = 0.1$. The high ϵ value at the start results in fast convergence of the algorithm, while the low ϵ values during the final iterations ensure a relatively low error regarding the objective function value.

Additional information for the classical Benders decomposition algorithm and the ϵ-optimal Benders decomposition algorithms regarding the number of iterations, the cumulative computation time for the dual sub-problem t_{dsp}, the cumulative computation time for computing the extreme rays t_{rays}, and the cumulative computation time for the master problem t_{mas}, are provided in Table 4-3. The computation time for updating the upper bound and generating optimality cuts is added to the solution time of the dual sub-problem t_{dsp}, the computation time for generating feasibility cuts is added to the computation time of the extreme rays t_{rays}, and the computation time for updating the lower bound is added to the computation time of the master problem t_{mas}.

For the classical Benders decomposition algorithm, the computation time of finding the extreme rays is the most time-consuming part when the problem size is small. However, when the problem size increases, the master problem becomes the most time-consuming part of the algorithm by a significant margin. By increasing the problem size, the classical Benders decomposition algorithm requires more Benders cuts, which increases the computation complexity of the master problem with each added iteration. The result is that the master problem becomes very time-consuming during the final iterations.

Table 4-3: Additional information Benders decomposition-based algorithms. N.A. indicates that no solution was found after three hours; t_{dsp} denotes the computation time of the dual sub-problem; t_{ray} denotes the computation time of finding extreme rays; and t_{mas} denotes the computation time of the master problem

N	Method	Iterations	t_{dsp}	t_{ray}	t_{mas}
3	Classical	62	50.7	74.0	21.1
	ϵ-Benders ($\epsilon = 0.01$)	79	85.1	91.8	30.1
	ϵ-Benders ($\epsilon = 0.05$)	68	62.9	89.7	19.7
	ϵ-Benders ($\epsilon = 0.1$)	63	57.0	85.6	18.9
	ϵ-Benders ($\epsilon_1(i)$)	59	49.2	69.3	15.1
	ϵ-Benders ($\epsilon_2(i)$)	72	59.4	80.8	19.3
4	Classical	85	96.1	145.6	147.3
	ϵ-Benders ($\epsilon = 0.01$)	113	141.3	158.8	58.3
	ϵ-Benders ($\epsilon = 0.05$)	99	108.6	157.9	38.6
	ϵ-Benders ($\epsilon = 0.1$)	97	103.3	149.9	38.0
	ϵ-Benders ($\epsilon_1(i)$)	94	98.8	143.3	66.9
	ϵ-Benders ($\epsilon_2(i)$)	99	113.9	163.0	60.4
5	Classical	104	135.7	208.2	7029.4
	ϵ-Benders ($\epsilon = 0.01$)	140	189.0	238.5	130.8
	ϵ-Benders ($\epsilon = 0.05$)	125	164.2	245.1	66.6
	ϵ-Benders ($\epsilon = 0.1$)	121	153.1	229.0	58.1
	ϵ-Benders ($\epsilon_1(i)$)	126	163.0	234.3	65.8
	ϵ-Benders ($\epsilon_2(i)$)	136	179.4	251.0	85.8
6	Classical	N.A	N.A	N.A	N.A
	ϵ-Benders ($\epsilon = 0.01$)	169	279.2	349.2	1142.8
	ϵ-Benders ($\epsilon = 0.05$)	151	235.5	349.4	129.4
	ϵ-Benders ($\epsilon = 0.1$)	147	237.4	355.7	94.8
	ϵ-Benders ($\epsilon_1(i)$)	158	245.6	349.9	142.1
	ϵ-Benders ($\epsilon_2(i)$)	168	268.3	361.0	143.7

The reason for the ϵ-optimal Benders decomposition algorithms being faster than the classical Benders decomposition algorithm when the problem size increases can be deduced from Table 4-3. By turning the master problem into a feasibility problem, the computation time of the master problem remains relatively low, even when the problem size increases. On the other hand, when the problem size is small, the ϵ-optimal Benders decomposition algorithms require a slightly higher computation time than the classical Benders decomposition algorithm. The ϵ-optimal Benders decomposition algorithms require more Benders cuts than the classical Benders decomposition algorithm before the solution is found because the Benders cuts are generated using non-optimal solutions to the master problem. The number of iterations, therefore, increases; the computation time is mainly determined by the number of iterations when the problem size is relatively small.

The convergence of the upper bound and the lower bound of the classical Benders decomposition algorithm can be seen in Fig. 4-2 for $N = 3$, in Fig. 4-3 for $N = 4$, and in Fig. 4-4 for $N = 5$. In all three cases, the dual sub-problem has a feasible and bounded solution only during two iterations, i.e., during the first and final iteration; this is why the upper bound only changes during the last iteration. The dual sub-problem is unbounded for all other iterations. The master problem, therefore, consists mainly of feasibility cuts; this explains why the cumulative computation time is higher for finding the extreme rays than for solving the dual sub-problem, as extreme rays are computed for all but two iterations, and the dual sub-problem is unbounded for all these iterations, which can be determined relatively fast.

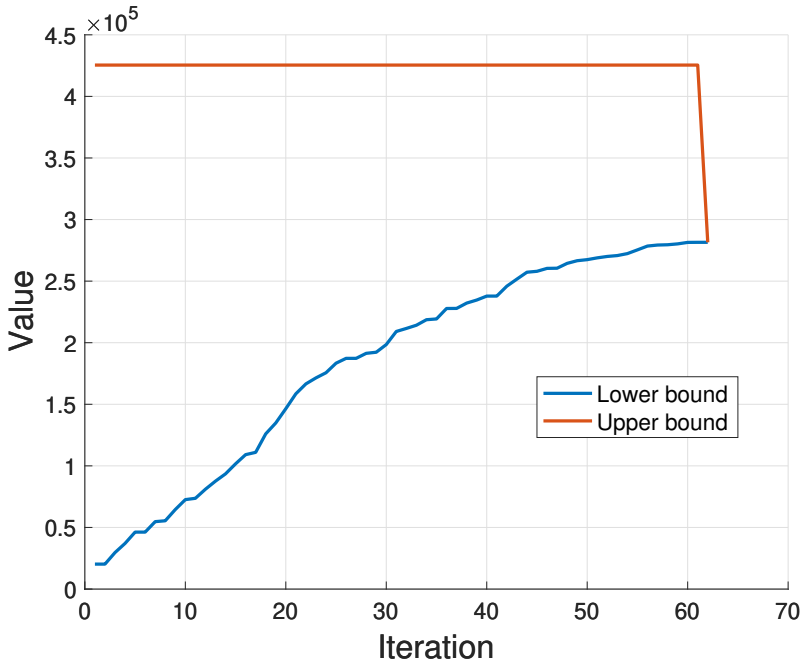


Figure 4-2: Convergence upper bound and lower bound of the classical Benders decomposition algorithm ($N = 3$)

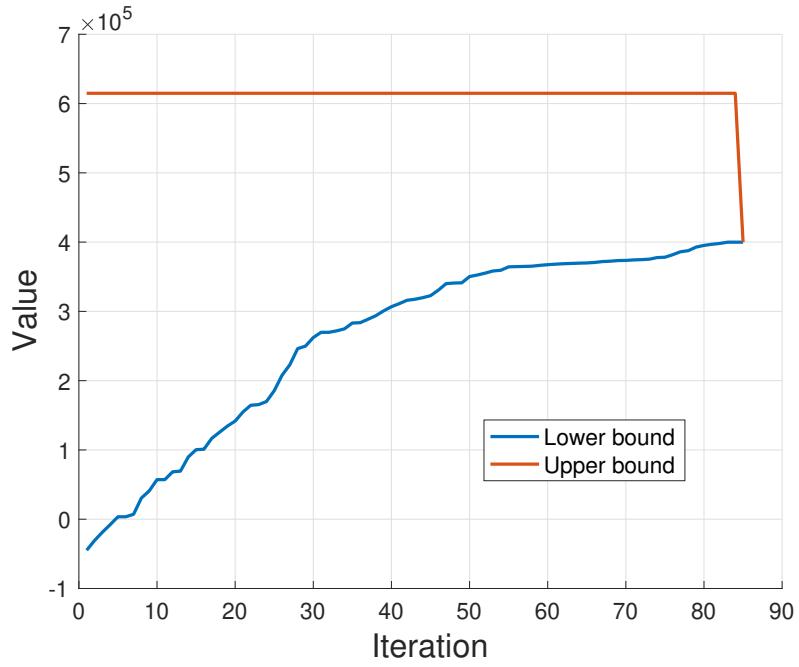


Figure 4-3: Convergence upper bound and lower bound of the classical Benders decomposition algorithm ($N = 4$)

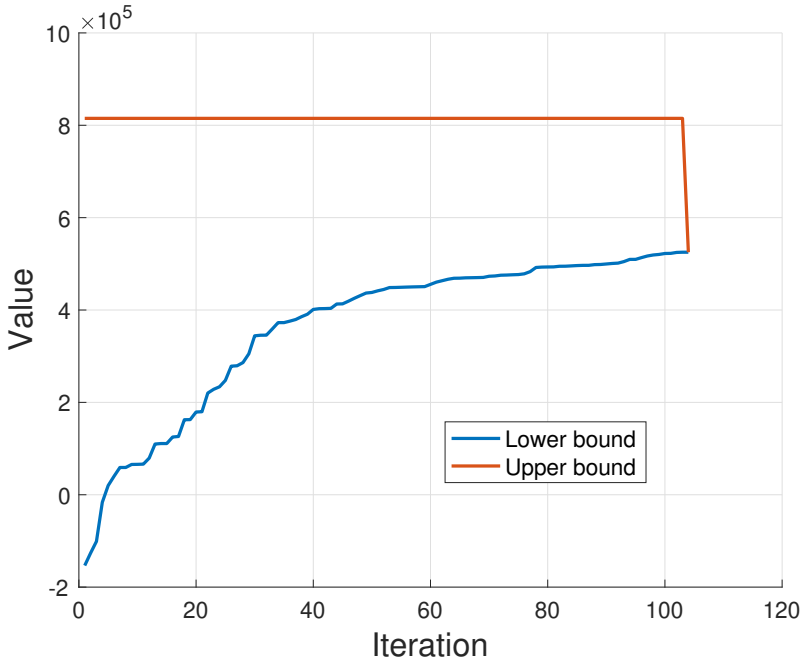


Figure 4-4: Convergence upper bound and lower bound of the classical Benders decomposition algorithm ($N = 5$)

The evolution of the upper bound throughout the iterations of the ϵ-optimal Benders decomposition algorithm is given for all N and all ϵ. Since the solution to the master problem is no longer optimal, the solution does not provide a valid lower bound; only the evolution of the upper bound is shown. The evolution of the upper bound is shown for all ϵ-optimal Benders decomposition algorithms in Fig. 4-5 for $N = 3$, in Fig. 4-6 for $N = 4$, in Fig. 4-7 for $N = 5$, and in Fig. 4-8 for $N = 6$.

While the upper bound only changes during the last iteration for the classical Benders decomposition algorithm, the upper bound does change during other iterations for the ϵ-optimal Benders decomposition algorithms. The optimal solutions to the classical Benders decomposition algorithm cause the dual sub-problem to be unbounded for most iterations, while the solutions to the master problem of the ϵ-optimal Benders decomposition algorithms — which are not necessarily optimal — do not always cause the dual sub-problem to be unbounded. The master problem of the ϵ-optimal Benders decomposition algorithms, therefore, consists of more optimality cuts than the classical Benders decomposition algorithm.

For all ϵ-optimal Benders decomposition algorithms, the upper bound remains unchanged during the beginning of the algorithm. The lower the constant value for ϵ, the faster the ϵ-optimal Benders decomposition algorithm can find a bounded solution to the dual sub-problem and use this solution to change the upper bound. The ϵ-optimal Benders decomposition algorithms with varying ϵ start with a high ϵ, which explains why it takes relatively long for the upper bound to change.

When the problem size increases, the number of times the upper bound changes increases significantly. In addition, the number of times the upper bound changes depends on the value chosen for ϵ: the lower ϵ, the higher the number of times the upper bound changes. The higher the value of ϵ, the closer more likely the solution is to be close to optimal, which results in an unbounded dual sub-problem, as we have seen from the classical Benders decomposition algorithm.

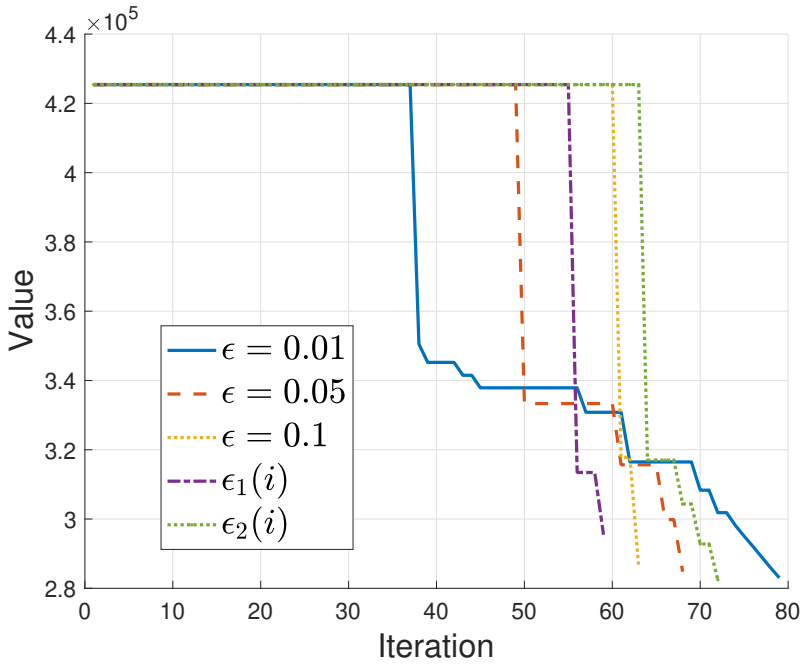


Figure 4-5: Evolution upper bound of the ϵ-optimal Benders decomposition algorithm ($N = 3$)

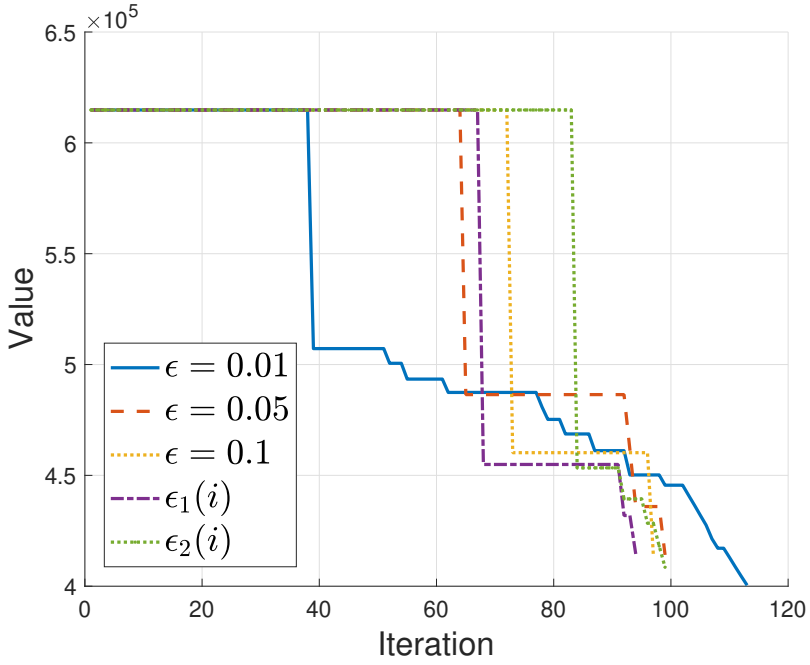


Figure 4-6: Evolution upper bound of the ϵ-optimal Benders decomposition algorithm ($N = 4$)

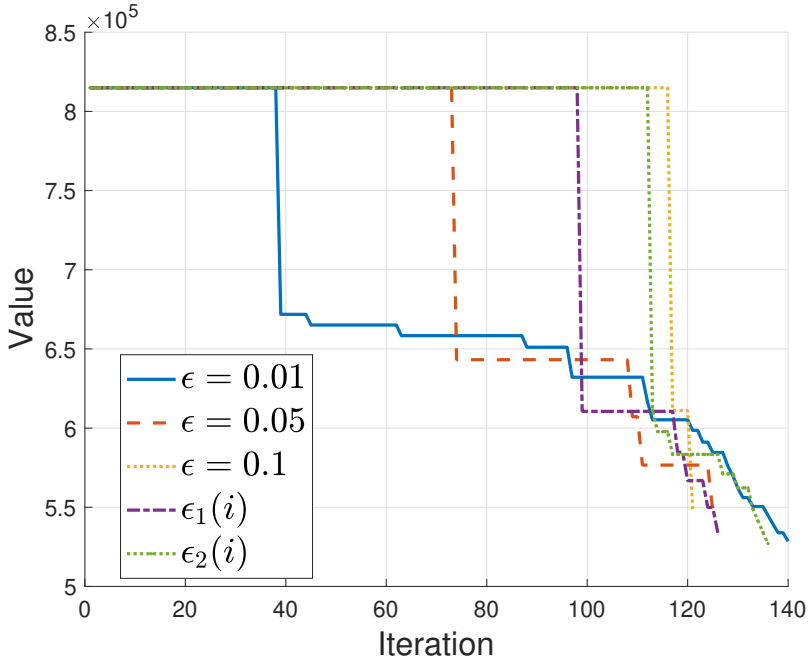


Figure 4-7: Evolution upper bound of the ϵ-optimal Benders decomposition algorithm ($N = 5$)

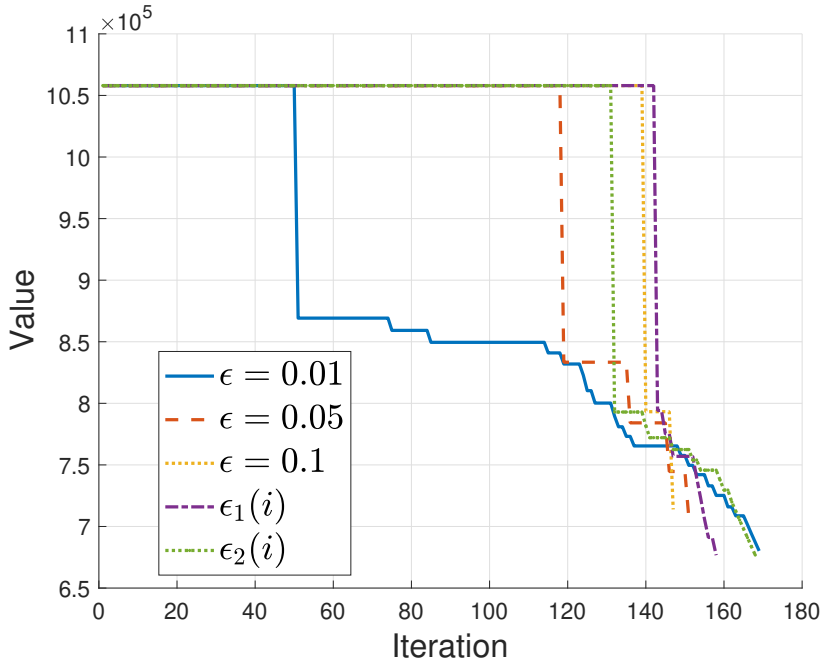


Figure 4-8: Evolution upper bound of the ϵ-optimal Benders decomposition algorithm ($N = 6$)

4-2-2 Evaluation

Solver `gurobi` is the best choice of all methods when the number of periods for which the departure frequencies are optimized is three or lower. However, when simulating for more periods and the problem size increases, `gurobi` has difficulties finding the optimal solution fast enough for real-time applications.

The classical Benders decomposition algorithm can find the exact solution in a much shorter time for $N = 4$ and $N = 5$ compared to `gurobi`. When the number of periods exceeds five, the classical Benders decomposition algorithm cannot find a solution within three hours. The high number of feasibility cuts required before the solution is found causes the master problem to increase in computation complexity with each iteration, which is the cause of the high computation time.

The ϵ-optimal Benders decomposition algorithm can find a solution much faster than both `gurobi` and the classical Benders decomposition algorithm if the problem size is large. By turning the master problem into a feasibility problem, the master problem remains relatively easy to solve, despite the high number of optimality and feasibility cuts that are added to the master problem. The ϵ-optimal Benders decomposition algorithm comes at a cost: the solution is not necessarily optimal, and the optimality gap depends on the choice for ϵ. A trade-off between solution quality and computation time can be made by choosing a value for ϵ. Furthermore, by setting ϵ as high at the beginning of the algorithm and decreasing ϵ with each iteration until a miminum value is reached, the ϵ-optimal Benders decomposition algorithm can make a better trade-off between the objective function value and the computation time.

4-3 Conclusions

This chapter evaluated the performance of the classical Benders decomposition algorithm and the ϵ-optimal Benders decomposition algorithm for the real-time optimization of train departure frequencies in metro networks.

The optimization was first solved using state-of-the-art solver `gurobi`, which was chosen as a benchmark. Next, the classical Benders decomposition algorithm was applied to the same optimization problem, after which five variations of the ϵ-optimal Benders decomposition algorithm were applied; three of these used a constant value for ϵ, while the other two decreased the value of ϵ with each iteration until a minimum value was reached. From the simulations, we can conclude that `gurobi` is not a suitable choice for real-time optimization of train departure frequencies. When increasing the number of periods over which the optimization is performed, the increase in the number of (integer) variables and constraints leads to a solution time that is too high for practical use in real-time applications.

The classical Benders decomposition algorithm was shown to be more suitable than `gurobi` when considering four or more periods for the optimization. However, the classical Benders decomposition algorithm was unsuitable for real-time optimization when simulating more than four periods. The classical Benders decomposition algorithm requires many Benders cuts before the solution is found, which causes the master problem to increase in computational complexity with every iteration until the master problem becomes a computational bottleneck.

The more periods are simulated, the larger the problem size and the more Benders cuts are required.

Much research has been carried out to accelerate Benders decomposition. A significant proportion of the proposed methods in the literature are based on the existence of multiple solutions to the dual sub-problem; these methods are unlikely to accelerate the classical Benders decomposition algorithm in this thesis, as the dual sub-problem is unbounded for all but the first and last iteration. Strengthening the feasibility cuts is a less researched topic and has mixed results.

The best method to accelerate Benders decomposition is, therefore, to reduce the computational complexity of the master problem directly. The ϵ-optimal Benders decomposition approach introduced in [23] has been shown to lead to a considerable decrease in computation time of the master problem — and therefore the whole algorithm — by turning the master problem into a feasibility problem. The best trade-off between solution time and quality can be achieved by choosing a relatively high initial value for ϵ and decreasing the value with each iteration until a minimum value is reached.

Chapter 5

Conclusions and discussion

This thesis applied Benders decomposition to decrease the computational burden of optimizing train departure frequencies under time-varying origin-destination passenger demands in metro networks. The ϵ-optimal Benders decomposition algorithm, a variation on the classical Bender decomposition algorithm, was applied to decrease the solution time further. This chapter concludes the thesis by answering the research questions and recommending future research.

5-1 Conclusions

The main research question of this thesis was:

Are Benders decomposition-based approaches suitable for optimizing train departure frequencies in metro networks?

To answer the main question, we will first answer the two sub-questions:

1. *Can Benders decomposition reduce the computational complexity of optimizing train departure frequencies in metro networks?*

   A simulation-based case study is utilized to evaluate the Benders decomposition-based approaches by comparing them against state-of-the-art solver `gurobi`, which is used as a benchmark. Simulations are done for three hours ($N = 3$), four hours ($N = 4$), five hours ($N = 5$), and six hours ($N = 6$); each hour is modeled as one period for which the passenger arrival rate is assumed constant.

   When the problem size is small enough ($N = 3$), `gurobi` can find the solution fast enough for real-time use. However, when the problem size increases, `gurobi` takes too long to find the solution for real-time applications. The classical Benders decomposition algorithm is significantly faster compared to `gurobi` when the problem size increases. Nevertheless, the classical Benders decomposition algorithm takes too long for real-time

use when $N = 5$ or higher. The reason for the high computation time of the classical Benders decomposition algorithm when the problem size increases is the high computation time of the master problem, which is due to the many Benders cuts required before the solution is found. The Benders cuts increase the computational complexity of the master problem with each iteration, until the master problem becomes a computational bottleneck.

The ϵ-optimal Benders decomposition algorithms can find a solution for all N relatively quickly. The ϵ-optimal Benders decomposition algorithm turns the master problem into a feasibility problem, which is generally easier to solve than an optimization problem. By reducing the computation time of the master problem — the most time-consuming part of the classical Benders decomposition algorithm for larger N — the ϵ-optimal Benders decomposition algorithm significantly reduces the total computation time. The reduced computation time does come with a drawback: the solution of the ϵ-optimal Benders decomposition algorithm is not necessarily optimal. A trade-off can be made between computation time and the solution accuracy by choosing a value for ϵ.

2. *What acceleration methods can be applied to Benders decomposition when optimizing train departure frequencies in metro networks?*

   Accelerating Benders decomposition is a heavily researched topic. A survey listing various acceleration techniques used in literature is presented in [43]. Choosing a suitable acceleration method is often problem-specific.

   Much of the research on accelerating Benders decomposition focuses on using the solutions to the dual sub-problem and their corresponding optimality cuts. The classical Benders decomposition algorithm found feasible and bounded solutions to the dual sub-problem only for the first and the last iteration; the dual sub-problem was feasible but unbounded for all other iterations. The master problem, therefore, consists mainly of feasibility cuts instead of optimality cuts. Since there is no bounded solution to the dual sub-problem for most of the iterations, methods that rely on the solution to the dual sub-problem cannot are not likely to be effective for the model that is used in this thesis.

   The method which might result in the most significant improvement in computation time can be deduced by looking at the computation time of all components of the classical Benders decomposition algorithm, i.e., the dual sub-problem, computing the extreme rays, and the master problem. When the optimization problem is relatively small, and few Benders cuts are required for the master problem, the dual sub-problem and computing the extreme rays are the most time-consuming parts of the classical Benders decomposition algorithm. However, when the problem increases in size, and additional Benders cuts are required, the master problem quickly becomes the most time-consuming part of the algorithm.

   The ϵ-optimal Benders decomposition algorithm introduced in [23] focuses on directly reducing the computation time of the master problem by turning the master problem from an optimization problem into a feasibility problem. The only requirement of the solution to the master problem is that its value is a specific factor below the current

upper bound. The algorithm terminates when there is no feasible solution anymore to the master problem. The accuracy and speed of the ϵ-optimal Benders decomposition algorithm depend on the choice of ϵ. Five different ϵ-optimal Benders decomposition algorithms are tested, three of which use constant ϵ values, while the other two use a high ϵ value at the start, decreasing with each iteration. The simulation-based case study shows that the ϵ-optimal Benders decomposition algorithm can achieve a relatively low error regarding the objective function value while reducing the computation time significantly compared to the classical Benders decomposition algorithm. The larger the problem size, the more significant the difference in computation time. The best trade-off between computation time and solution accuracy is made by starting with a high value for ϵ and decreasing this value with each iteration until a minimum value is reached.

After answering the two sub-questions, we can answer the main question: Yes, Benders decomposition-based approaches are suitable for optimizing train departure frequencies in metro networks. The simulation-based case study shows that the ϵ-optimal Benders decomposition algorithm can find a relatively accurate solution fast enough for real-time applications, which means the algorithm is suitable for optimizing train departure frequencies in metro networks.

This thesis dealt with the train departure frequency optimization problem in metro networks based on the model developed in [32], which can explicitly include time-varying origin-destination passenger demands. The contributions of this thesis can be summarized as follows:

1. The main contribution of this thesis is reducing the computational burden of the train departure frequency problem by applying Benders decomposition-based algorithms. First, the classical Benders decomposition algorithm [5] is applied, after which the ϵ-optimal Benders decomposition algorithm [23] is applied to reduce the computation time further.

2. Several Benders decomposition-based algorithms are compared in a simulation-based case study to facilitate the method selection when solving train departure frequency optimization problems.

A paper based on this thesis has been written and submitted to the 26th IEEE International Conference on Intelligent Transportation Systems ITSC 2023. The paper can be found in Appendix C.

5-2 Future work

This section outlines the suggestions for future work.

Energy consumption

In this thesis, the objective function is a sum of the time passengers spend in the metro system and the operational costs of the dispatched trains. For a more accurate model, the energy consumption of the dispatched trains can be computed by considering the speed of

the trains for all speed phases, the train traction force and train braking force, the aerodynamic drag and rolling mechanical resistances, and the gradient resistances and curve resistances [54, 60, 61]. Since the computation of energy consumption is nonlinear, computational complexity significantly increases for models considering energy consumption as an objective.

Stop-skipping

In this thesis, it is assumed that every train stops at every station. In real life some stations will naturally have a higher passenger flow than other stations. Therefore, some research in timetable scheduling considers stop-skipping in their model, enabling trains to stop at high-demand stations and skip low-demand stations [27, 52, 59, 65]. There are generally two ways that stop-skipping can be applied: (1) by determining a predefined set of stations to be skipped by certain trains and (2) by allowing all trains to skip a station when deemed beneficial, typically called dynamic stop-skipping. Adding stop-skipping would likely improve the objective function of the model used in this thesis at the cost of increased computational complexity.

Pareto-optimal cuts

One of the most commonly used acceleration methods for Benders decomposition is by using so-called Pareto-optimal cuts, introduced by [38]. The core idea is that when there are multiple solutions to the dual sub-problem, there exists a solution amongst those solutions for which the resulting optimality cut leads to the fastest convergence of the algorithm. Since the dual sub-problem of the classical Benders decomposition algorithm is unbounded for all but the first and last iteration of the algorithm, it is unlikely that the Pareto-optimal method would significantly reduce the total computation time. However, the dual sub-problem of the ϵ-optimal Benders decomposition algorithm does have feasible and bounded solutions for more than two iterations, especially when the value for ϵ is set low. Finding the Pareto-optimal cut whenever there are multiple feasible and bounded solutions might result in faster convergence of the ϵ-optimal Benders decomposition algorithm. Faster convergence is not guaranteed, as solving the secondary problem (finding the Pareto-optimal cut) can be time-consuming. Different methods can be used to find the Pareto-optimal cut, such as an analytic-center cutting plane method [22, 41].

Tighter feasibility cuts

Since the solution to the dual sub-problem is unbounded for all but two iterations for the classical Benders decomposition algorithm, it would make sense to investigate the possibility of generating better feasibility cuts. By using the distance between the feasibility cuts and the line connecting feasible and infeasible points, tighter feasibility cuts were generated in [58], resulting in faster convergence. The tighter cut generation would be worth investigating, although it does involve solving a computationally expensive auxiliary problem.

Combinatorial cuts

One of the potential causes of weak feasibility cuts is the presence of big-M coefficients [15]. By searching for minimal infeasible subsystems, stronger cuts — referred to as combinatorial

cuts — were obtained in [15]. The model used in this thesis uses big-M coefficients to transform the min function into linear inequalities (see Appendix A), and it would therefore be worth investigating the effects of using combinatorial cuts.

Appendix A

Transformation of the min function

The nonlinear equation

$$n_p^{\text{absorb}}(k) = \min\left(n_p^{\text{wait}}\ (k), C_p(k)\right) \tag{A-1}$$

is used to compute the number of passengers who are able to board a train. When there is enough remaining capacity, all waiting passengers are able to board the train. When there is not enough capacity for all waiting passengers, the number of passengers who board the train is equal to remaining capacity.

In this appendix, the method in [56] is used to transform the min function into linear inequalities.

Auxiliary variable $f_p^{\text{absorb}}(k)$ is introduced, with:

$$f_p^{\text{absorb}}(k) = n_p^{\text{wait}}\ (k) - C_p(k) \tag{A-2}$$

Next, auxiliary binary variable $\delta_p^{\text{absorb}}(k)$ is introduced, which is equal to 1 if $f_p^{\text{absorb}}(k)$ is non-negative and equal to 0 if $f_p^{\text{absorb}}(k)$ is negative; this can be expressed by using the following equations:

$$f_p^{\text{absorb}}(k) \leq M_p(1 - \delta_p^{\text{absorb}}(k)) \tag{A-3a}$$

$$f_p^{\text{absorb}}(k) \geq \epsilon + (m_p - \epsilon)\delta_p^{\text{absorb}}(k)), \tag{A-3b}$$

where M_p and m_p are the maximum and minimum value of $f_p^{\text{absorb}}(k)$, respectively, and ϵ is a small positive number. Now, (A-1) is rewritten as follows:

$$n_p^{\text{absorb}}(k) = \delta_p^{\text{absorb}}(k) n_p^{\text{wait}}\ (k) + (1 - \delta_p^{\text{absorb}}(k)) C_p(k) \tag{A-4}$$

The product of variables $\delta_p^{\text{absorb}}(k)$ and $n_p^{\text{wait}}(k)$ is re-expressed using the following linear inequalities:

$$z_p^{\text{wait}}(k) \leq M_w \delta_p^{\text{absorb}}(k)), \tag{A-5a}$$
$$z_p^{\text{wait}}(k) \geq m_w \delta_p^{\text{absorb}}(k)), \tag{A-5b}$$
$$z_p^{\text{wait}}(k) \leq n_p^{\text{wait}}(k) - m_w(1 - \delta_p^{\text{absorb}}(k))), \tag{A-5c}$$
$$z_p^{\text{wait}}(k) \geq n_p^{\text{wait}}(k) - M_w(1 - \delta_p^{\text{absorb}}(k))), \tag{A-5d}$$
$$\tag{A-5e}$$

where M_w and m_w represent the maximum and minimum value of $n_p^{\text{wait}}(k)$, respectively. The product of variables $\delta_p^{\text{absorb}}(k)$ and $C_p(k)$ is re-expressed using the following linear inequalities:

$$z_p^{\text{cap}}(k) \leq M_c \delta_p^{\text{absorb}}(k)), \tag{A-6a}$$
$$z_p^{\text{cap}}(k) \geq m_c \delta_p^{\text{absorb}}(k)), \tag{A-6b}$$
$$z_p^{\text{cap}}(k) \leq C_p(k) - m_c(1 - \delta_p^{\text{absorb}}(k))), \tag{A-6c}$$
$$z_p^{\text{cap}}(k) \geq C_p(k) - M_c(1 - \delta_p^{\text{absorb}}(k))), \tag{A-6d}$$
$$\tag{A-6e}$$

where M_c and m_c represent the maximum and minimum value of $C_p(k)$, respectively. Finally, for compactness, (A-4) is written as:

$$E_{p,1}(k)\delta_p^{\text{absorb}}(k) + E_{p,2}(k) z_p^{\text{wait}}(k) \leq E_{p,3}(k) n_p^{\text{wait}}(k) + E_{p,4}(k) \tag{A-7a}$$
$$E_{p,5}(k)\delta_p^{\text{absorb}}(k) + E_{p,6}(k) z_p^{\text{cap}}(k) \leq E_{p,7}(k) C_p(k) + E_{p,8}(k) \tag{A-7b}$$

Appendix B

Computation of extreme rays

In this appendix, the equations are given that are used to compute the extreme rays when the dual sub-problem is feasible but unbounded. The set of equations is as follows:

$$J_{\text{dsp}} > 0, \tag{B-1a}$$
$$q_p^{\text{capacity}}(k) = q_p^{\text{absorb}}(k) - E_7 q_p^{\text{capacity, auxiliary}}(k), \tag{B-1b}$$
$$q_{p,m}^{\text{number}}(k) = q_{p,m}^{n}(k-1), \tag{B-1c}$$
$$q_{p,m}^{\text{absorb}}(k) = -q_{p,m}^{\text{number}}(k) + q_{p,m}^{\text{depart}}(k), \tag{B-1d}$$
$$q_p^{\text{wait}}(k) = -E_3 q_p^{\text{wait, auxiliary}}(k), \tag{B-1e}$$
$$q_{p,m}^{\text{train}}(k) = -q_p^{\text{capacity}}(k) + q_{p,m}^{\text{depart}}(k) + \sum_{q \in \text{sta}(p)} \chi_{q,p,m} q_{q,p,m}^{\text{trans}}(k) + q_{p,\text{sta}(p)}^{\text{alight}}(k), \tag{B-1f}$$
$$q_{p,\text{sta}(p)}^{\text{alight}}(k) = -q_{p,m}^{\text{depart}}(k), \tag{B-1g}$$
$$q_{p,m \in S/\{\text{sta}(p)\}}^{\text{alight}}(k) = -q_{p,m}^{\text{depart}}(k), \tag{B-1h}$$
$$q_{p,m}^{\text{depart}}(k) = \frac{T - \bar{r}_{\text{p}^{\text{pla}}(p)}}{T} q_{p,m}^{\text{train}}(k) + \frac{\bar{r}_{\text{p}^{\text{pla}}}(k)}{T} q_{p,m}^{\text{train}}(k-1), \tag{B-1i}$$
$$q_{q,p,m}^{\text{trans}}(k) = q_{p,m \in S/\{\text{sta}(p)\}}^{\text{alight}}(k) + \frac{T - \theta_{q,p}^{\text{trans}}}{T} q_{p,m}^{\text{arrive,trans}}(k) + \frac{\theta_{q,p}^{\text{trans}}}{T} q_{p,m}^{\text{arrive,trans}}(k-1), \tag{B-1j}$$
$$q_{p,m}^{\text{arrive,trans}}(k) = q_{p,m}^{\text{number}}(k), \tag{B-1k}$$
$$q_p^{\text{absorb}}(k) = \sum_{m \in S} \alpha_{p,m}(k) u_{p,m}^{\text{absorb}}(k), \tag{B-1l}$$
$$q_p^{\text{wait, auxiliary}}(k) = -E_{p,2}(k) q_p^{\text{absorb}}(k), \tag{B-1m}$$
$$q_p^{\text{capacity, auxiliary}}(k) = E_{p,6}(k) q_p^{\text{absorb}}(k) + E_{p,7}(k) u_p^{\text{absorb}}(k), \tag{B-1n}$$
$$q_p^{\text{capacity}}(k), ..., q_p^{\text{absorb}}(k) \in \mathbb{R}, \tag{B-1o}$$
$$q_p^{\text{wait, auxiliary}}(k), q_p^{\text{capacity, auxiliary}}(k) \geq 0, \tag{B-1p}$$
$$k = k_0, k_0 + 1, ..., k_0 + N - 1,$$

where J_{dsp} is given as:

$$\begin{aligned}\max \ J_{\text{dsp}} = \sum_{k=k_0}^{k_0+N-1} \sum_{p\in P} \sum_{m\in S} \Big(& u_p^{\text{capacity}}(k) C_{\max} \bar{f}_p(k) - u_{p,m}^{\text{number}}(k) \lambda_{p,m}(k) T - u_p^{\text{wait}}(k) \lambda_p(k) T \\ & + u_p^{\text{wait, auxiliary}}(k) \left(E_{p,1}(k) \bar{\delta}_p^{\text{absorb}}(k) - E_{p,4}(k) \right) \\ & + u_p^{\text{capacity, auxiliary}}(k) \left(E_{p,5}(k) \bar{\delta}_p^{\text{absorb}}(k) - E_{p,8}(k) \right) \Big)\end{aligned} \tag{B-2}$$

Solving the equations above leads to a vector of extreme rays $[\bar{q}_p^{\text{capacity}}(k), ..., \bar{q}_p^{\text{capacity, auxiliary}}(k)]$.

Appendix C

Conference paper

A paper was written based on the main findings of this thesis. The paper has been submitted to the 26th IEEE International Conference on Intelligent Transportation Systems (ITSC 2023).

Benders decomposition-based real-time optimization of train departure frequencies in metro networks

Alexander Daman, Xiaoyu Liu, Azita Dabiri, and Bart De Schutter, *Fellow, IEEE*

***Abstract*— Timetables determine the service quality for passengers and the energy consumption of trains in metro systems. In metro networks, a timetable can be made by designing train departure frequencies for different periods of the day, which is typically formulated as a mixed-integer linear programming (MILP) problem. In this paper, we first apply Benders decomposition to optimize the departure frequencies considering time-varying passenger origin-destination demands in metro networks. An ϵ-optimal Benders decomposition approach is subsequently used to reduce the solution time further. The performance of both methods is illustrated in a simulation-based case study using a grid metro network. The results show that both the classical Benders decomposition approach and the ϵ-optimal Benders decomposition approach can significantly reduce the computation time for the real-time optimization of train departure frequencies in metro networks. In addition, the ϵ-optimal Benders decomposition approach can further reduce the solution time compared to the classical Benders decomposition approach when the problem scale increases while maintaining an acceptable level of performance.**

I. Introduction

Metro systems have become essential to urban transportation, providing millions of people with fast, efficient, and sustainable travel options, especially in large cities. The metro system is particularly critical in densely populated urban areas, where an efficient and reliable timetable is paramount for passenger satisfaction and the energy efficiency of the metro system.

Efficient train scheduling approaches enable metro systems to optimize energy consumption, reduce waiting times, and adjust transport capacity to meet passenger demands of different periods. A nonlinear programming problem (NLP) was formulated in [1] to minimize the time passengers spend and the energy consumption of trains in a metro line, for which an iterative convex programming approach was proposed. A bi-directional train line was considered in [2], and a Lagrangian-based method was proposed to solve the resulting NLP problem. An adaptive large neighborhood search algorithm was developed in [3] for the timetable scheduling problem of a rail rapid transit line so as to create convenient timetables for passengers considering a dynamic demand pattern. To improve the efficiency of passenger-centric timetable scheduling in metro networks, a simplified model was developed in [4], where the resulting optimization problem is solved in a moving horizon manner for real-time timetable scheduling.

The authors are with the Delft Center for Systems and Control, Delft University of Technology, 2628CD Delft, The Netherlands alexander.daman1996@gmail.com, x.liu-20@tudelft.nl, a.dabiri@tudelft.nl, b.deschutter@tudelft.nl

In metro networks, trains typically operate with a relatively short headway, and thus the train departure frequency, which refers to the number of trains departing from a line per time unit, is crucial for the transport capacity of metro networks. To handle time-varying passenger origin-destination demands, it is necessary to implement effective strategies for optimizing departure frequencies in real time. Previous studies, such as [5], have utilized heuristic and exact methods to optimize train capacities and line frequencies within metro networks. Similarly, [6] applied mixed-integer nonlinear programming (MINLP) to optimize train capacities and line frequencies in urban metro networks. A passenger absorption model was proposed in [7] to optimize the departure frequency of trains of each line in metro networks, and the resulting problem was formulated as a mixed-integer linear programming (MILP) problem.

Real-time scheduling models for timetables often involve non-continuous variables, resulting in non-convex optimization problems that can be time-consuming to solve. Benders decomposition is an efficient methodology to reduce the computational burden in large-scale MILP problems by splitting the MILP into two small-scale problems [8], [9]. Benders decomposition has also been used in railway timetable scheduling problems. Taking into account the uncertain passengers transfer time in metro networks, a generalized Benders decomposition approach was developed in [10] to efficiently solve the resulting MILP problem. A logic-based Benders decomposition approach that can reuse the precomputed logic Benders cuts to reduce the computation burden of the timetable rescheduling problem was proposed in [11]. In [12], the solution time of the Benders decomposition algorithm was reduced by splitting the algorithm solution process into three steps to address the fact that the relation between routing and scheduling variables is absent in the master problem. The proposed Benders decomposition approaches in [10], [11], and [12] were all shown to reduce the solution time significantly; however, passenger origin-destination (OD) demands were not considered explicitly.

This paper deals with the train departure frequency optimization problem in metro networks based on the model developed in [7], which can explicitly include time-varying OD passenger demands. The main contribution of this paper is twofold: (1) Benders decomposition-based algorithms are used in the train departure frequency optimization problem to reduce the computational burden; (2) several Benders decomposition-based algorithms are compared on a simulation-based case study, which can facilitate the method

selection when solving train departure frequency optimization problems.

The remainder of this paper is structured as follows. In Section II, a problem formulation is given. In Section III, the classical and ϵ-optimal Benders decomposition algorithms used for the passenger absorption model are discussed. Simulation results are provided in Section IV. Finally, conclusions are given in Section V.

II. Problem Formulation

In this work, the model proposed in [7] is used to optimize train departure frequencies of metro networks. Time-varying passenger demands are approximated using piecewise constant functions in the model, allowing a balanced trade-off between solution time and accuracy. We briefly introduce the model and the corresponding optimization problem below, and for more details on the model, we refer to paper [7].

In the passenger absorption model, the planning time window is divided into several periods, and passenger OD demands are assumed to be constant in each period. The total travel time of passengers within a given planning time window is estimated by:

$$J_{\text{time}} = \sum_{k=k_0}^{k_0+N-1} \sum_{p\in P} \Big(n_p(k)T + n_p^{\text{depart}}(k)\bar{r}_p + n_p^{\text{arr,tra}}(k)\theta_p^{\text{trans}} \Big) + \sum_{p\in P} n_p(k_0+N)T, \tag{1}$$

where N denotes the number of periods in the planning time window; P is the set of all platforms in the metro network; T is the length of a period; $n_p(k)$ denotes the number of passenger waiting at platform p at the start of period k; $n_p^{\text{depart}}(k)$ represents the number of passenger departing from platform p during period k; $n_p^{\text{arr,tra}}(k)$ denotes the number of passengers arriving at platform p with the intention of transferring to another platform during period k; and θ_p^{trans} is the average travel time for passengers transferring from platform p. In the metro network, trains travel a predetermined route, stopping at every platform. The average travel time for a train departing from platform p to the next platform on its route is denoted as $\bar{r}_p$. The energy consumption of trains in the planning time window is estimated by:

$$J_{\text{cost}} = \sum_{k=k_0}^{k_0+N-1} \sum_{p\in P} f_p(k)\bar{E}_p, \tag{2}$$

where $f_p(k)$ is the departure frequency at platform p during period k, and $\bar{E}_p$ denotes the average operational costs associated with dispatching a train from platform p towards the next platform on its route. The optimization problem is given as:

$$\min \quad J = J_{\text{time}} + \zeta J_{\text{cost}}, \tag{3a}$$

subject to

$$f_p(k) = \frac{T-\gamma_p}{T} l_p\left(k-\delta_p\right) + \frac{\gamma_p}{T} l_p\left(k-\delta_p-1\right), \tag{3b}$$

$$f_p(k) \le f_p^{\max}, \tag{3c}$$

$$C_p(k) = f_p(k)C_{\max} - \sum_{m\in S} n_{p,m}^{\text{train}}(k), \tag{3d}$$

$$n_{p,m}(k+1) = n_{p,m}(k) + \lambda_{p,m}(k)T + n_{p,m}^{\text{arr,tra}}(k) - n_{p,m}^{\text{absorb}}(k), \tag{3e}$$

$$n_p^{\text{wait}}(k) = n_p(k) + \lambda_p(k)T + n_p^{\text{arr,tra}}(k), \tag{3f}$$

$$n_p^{\text{absorb}}(k) = \min\Big(C_p(k), n_p^{\text{wait}}(k)\Big), \tag{3g}$$

$$n_{p,m}^{\text{absorb}}(k) = \alpha_{p,m}(k) n_p^{\text{absorb}}(k), \tag{3h}$$

$$n_{p,m}^{\text{train}}(k) = \frac{T-\bar{r}_p^{\text{pla}}}{T} n_{\text{p}^{\text{pla}}(p,m)}^{\text{depart}}(k) + \frac{\bar{r}_p^{\text{pla}}}{T} n_{\text{p}^{\text{pla}}(p,m)}^{\text{depart}}(k-1), \tag{3i}$$

$$n_{p,\text{sta}(p)}^{\text{alight}}(k) = n_{p,m}^{\text{train}}(k), \tag{3j}$$

$$n_{p,m\in S/\{\text{sta}(p)\}}^{\text{alight}}(k) = n_{p,q,m}^{\text{trans}}(k), \tag{3k}$$

$$n_{p,m}^{\text{depart}}(k) = n_{p,m}^{\text{train}}(k) - n_{p,m}^{\text{alight}}(k) + n_{p,m}^{\text{absorb}}(k), \tag{3l}$$

$$n_{q,p,m}^{\text{trans}}(k) = \chi_{q,p,m}(k) n_{q,m}^{\text{train}}(k), \tag{3m}$$

$$n_{p,m}^{\text{arr,tra}}(k) = \sum_{q\in \text{pla}(p)} \Big(\frac{T-\theta_{q,p}^{\text{trans}}}{T} n_{q,p,m}^{\text{trans}}(k) + \frac{\theta_{q,p}^{\text{trans}}}{T} n_{q,p,m}^{\text{trans}}(k-1) \Big), \tag{3n}$$

$$k = k_0, k_0+1, ..., k_0+N-1,$$

where ζ is a weight used to balance both objectives; $l_p(k)$ denotes the train departure frequency of the starting platform of the line on which platform p lies; $\delta_p = \lfloor \psi_p/T \rfloor$ and $\gamma_p = \psi_p - \delta_p T$, with ψ_p denoting the average travel time for train between departing from a starting platform of a line and departing from another platform p of that same line; $f_p^{\max}$ denotes the maximum train departure frequency of platform p; $C_p(k)$ represents the remaining capacity on a train at platform p during period k with $C_{\max}$ being the maximum capacity of a train; $n_{p,m}^{\text{train}}(k)$ is the number of passengers on board of trains at platform p with destination m during period k; $n_{p,m}(k)$ denotes the number of passenger waiting at platform p with destination m during period k; $\lambda_{p,m}(k)$ is the passenger arrival rate at platform p with destination m during period k; $n_{q,p,m}^{\text{arr,tra}}(k)$ denotes the number of transferring passengers arriving at platform q to transfer to platform p with destination m during period k; $n_{p,m}^{\text{absorb}}(k)$ represents the number of passengers who board a train at platform p with destination m during period k; $n_p^{\text{wait}}(k)$ denotes the number of passengers waiting for a train at platform p with destination m during period k; and $n_{p,m}^{\text{absorb}}(k)$ denotes the number of passengers alighting a train at platform p with destination m during period k. Parameter $\alpha_{p,m}$ is the relative fraction of passengers that board a train at platform p whose destination is station m; and $\chi_{q,p,m}$ is the relative fraction of passengers arriving at platform q with destination m, who will transfer from platform q to platform p.

Note that (3g) is a nonlinear function, and we can use the method in [13] to transform (3g) into linear inequalities. Then, we obtain an MILP problem for train departure frequency optimization; for a more elaborate explanation of the

resulting MILP problem, we refer to [7].

The solution time of directly solving this MILP problem is significant. Therefore, this paper aims to present approaches to solve the resulting MILP in a time-efficient manner.

III. Benders Decomposition-based Train Departure Frequency Optimization

Benders decomposition [8] is an efficient method for solving large-scale optimization problems involving both continuous and discrete variables. In Benders decomposition, an optimization problem is divided into a master problem and a dual sub-problem that can be solved independently. The master problem is formulated as an MILP problem to determine the integer variables, while the dual sub-problem is formulated as a linear programming problem. The dual sub-problem is either feasible and bounded, after which a so-called optimality cut is added to the master problem, or is unbounded, after which a feasibility cut is added to the master problem.

A. Classical Benders Decomposition for Train Departure Frequency Optimization

The classical Benders decomposition [8] is applied in this section for the MILP problem (3) described in Section II. In this paper, according to the definition used in [8], $l_p(k)$ and $\delta_p^{\text{absorb}}(k)$ are the so-called "complicating variables", as they are integer and binary variables, respectively. the MILP problem is non-convex due to these variables. Since T, $\gamma_p(k)$, and $\delta_p(k)$ are all parameters, it follows from (3b) that once $l_p(k)$ is given, $f_p(k)$ is also known. We define a vector $\mathbf{y}(k_0)$ to collect the integer variables $l_p(k)$, binary variables $\delta_p^{\text{absorb}}(k)$, and $f_p(k)$ in the planning time window starting from period k_0. Then, all other variables related to the number of passengers in the planning time window starting from period k_0 are collected in a vector $\mathbf{x}(k_0)$. For compactness, we can write problem (3) as:

$$\min_{\mathbf{x}(k_0),\mathbf{y}(k_0)} \quad J = \mathbf{c}^{\mathrm{T}}(k_0)\mathbf{x}(k_0) + \mathbf{g}^{\mathrm{T}}(k_0)\mathbf{y}(k_0) \tag{4a}$$

$$\text{s.t.} \quad A(k_0)\mathbf{x}(k_0) + B(k_0)\mathbf{y}(k_0) = \mathbf{b}(k_0), \tag{4b}$$

$$D(k_0)\mathbf{x}(k_0) + E(k_0)\mathbf{y}(k_0) \leq \mathbf{d}(k_0), \tag{4c}$$

$$\mathbf{x}(k_0) \in \mathbb{R}^{n_1}, \tag{4d}$$

$$\mathbf{y}(k_0) \in \mathbb{Y}^{n_2}, \tag{4e}$$

$$\mathbf{x}(k_0) \geq 0, \tag{4f}$$

where (4a) represents the objective function (3a), (4b) collects the equality constraints, (4c) collects the inequality constraints, and $\mathbb{Y}^{n_2}$ defines the feasible set for $\mathbf{y}(k_0)$.

By fixing $\mathbf{y}(k_0)$ as $\bar{\mathbf{y}}(k_0)$ in Benders decomposition, the sub-problem turns into a linear programming problem, and by using duality theory and introducing dual variables $\mathbf{u}_1(k_0)$ and $\mathbf{u}_2(k_0)$, the dual sub-problem becomes:

$$\max_{\mathbf{u}_1(k_0),\mathbf{u}_2(k_0)} J_{\text{dsp}} = \mathbf{u}_1^{\mathrm{T}}(k_0)\big(B(k_0)\bar{\mathbf{y}}(k_0) - \mathbf{b}(k_0)\big) \tag{5a}$$

$$+ \mathbf{u}_2^{\mathrm{T}}(k_0)\big(E(k_0)\bar{\mathbf{y}}(k_0) - \mathbf{d}(k_0)\big) + \mathbf{g}^{\mathrm{T}}(k_0)\bar{\mathbf{y}}(k_0)$$

$$\text{s.t.} \quad \mathbf{u}_1^{\mathrm{T}}(k_0)A(k_0) + \mathbf{u}_2^{\mathrm{T}}(k_0)D(k_0) = \mathbf{c}^{\mathrm{T}}(k_0), \tag{5b}$$

$$\mathbf{u}_1(k_0) \in \mathbb{R}^{m_1}, \tag{5c}$$

$$\mathbf{u}_2(k_0) \in \mathbb{R}^{m_2}_{\geq 0}. \tag{5d}$$

If the feasible set of (5) is not empty, the dual sub-problem can be either unbounded or feasible for any arbitrary choice of $\bar{\mathbf{y}}(k_0)$. If the dual sub-problem is unbounded, there exists a pair of extreme rays $\bar{\mathbf{r}}_{q1}(k_0) \in \mathbb{Q}_1$ and $\bar{\mathbf{r}}_{q2}(k_0) \in \mathbb{Q}_2$, with $\mathbb{Q}_1$ and $\mathbb{Q}_2$ being the sets of extreme rays, for which $\bar{\mathbf{r}}_{q1}^{\mathrm{T}}(k_0)(B(k_0)\mathbf{y}(k_0) - \mathbf{b}(k_0)) + \bar{\mathbf{r}}_{q2}^{\mathrm{T}}(k_0)(E(k_0)\mathbf{y}(k_0) - \mathbf{d}(k_0)) > 0$. To avoid this, the following feasibility cut is added to the master problem:

$$\bar{\mathbf{r}}_{q1}^{\mathrm{T}}(k_0)\big(B(k_0)\mathbf{y}(k_0) - \mathbf{b}(k_0)\big) + \bar{\mathbf{r}}_{q2}^{\mathrm{T}}(k_0)\big(E(k_0)\mathbf{y}(k_0) - \mathbf{d}(k_0)\big) \leq 0. \tag{6}$$

While there may be multiple possible extreme rays which lead to unboundedness in the dual sub-problem, only one pair of extreme rays is used for the feasibility cut.

If a feasible and bounded solution can be found for dual sub-problem (5), the solution for the dual variables can be denoted as the extreme points, i.e., $\bar{\mathbf{u}}_{e1}(k_0) \in \mathbb{E}_1$ and $\bar{\mathbf{u}}_{e2}(k_0) \in \mathbb{E}_2$, with $\mathbb{E}_1$ and $\mathbb{E}_2$ being the sets of extreme points. We use J_{dsp} to denote the value of the objective function of the dual sub-problem. The optimal value of the objective function provides an upper bound of the original optimization problem, which is denoted as U_{ub}. For the ith iteration of the Benders decomposition algorithm, the upper bound is updated as follows: $U_{\text{ub}}^i = \min\left(U_{\text{ub}}^{i-1}, J_{\text{dsp}}^i\right)$. In addition, an optimality cut is added to the master problem:

$$\bar{\mathbf{u}}_{e1}^{\mathrm{T}}(k_0)(B(k_0)\mathbf{y}(k_0) + \mathbf{b}(k_0)) - \bar{\mathbf{u}}_{e2}^{\mathrm{T}}(k_0)(E(k_0)\mathbf{y}(k_0) - \mathbf{d}(k_0)) \geq -\eta. \tag{7}$$

Finally, the master problem (MP) is formulated as:

$$\min_{\mathbf{y}(k_0),\eta} \quad J_{\text{mp}} = \mathbf{g}^{\mathrm{T}}(k_0)\mathbf{y}(k_0) + \eta \tag{8}$$

$$\text{s.t.} \quad (4\text{e}), (6), (7)$$

The solution $\bar{\mathbf{y}}(k_0)$ to the master problem is used for dual sub-problem (5) in the next iteration and is also used to update the lower bound: $U_{\text{lb}}^i = \min\left(U_{\text{lb}}^{i-1}, J_{\text{mp}}^i\right)$, where J_{mp} denotes the objective function value of the master problem.

The procedure of the classical Benders decomposition-based train departure frequency optimization algorithm used is presented in Algorithm 1.

B. ϵ-Optimal Benders Decomposition for Train Departure Frequency Optimization

To reduce the computation time of the master problem, [14] proposed a variant of Benders decomposition where the master problem stops as soon as a feasible solution is found, as opposed to an optimal solution. The algorithm is then guaranteed to terminate in a finite number of steps, as there is a finite number of optimal dual solutions for the sub-problem. Like the classical Benders decomposition, the

Algorithm 1 Classical Benders decomposition-based train departure frequency optimization algorithm

Input: α, ζ, N, P, S, $\theta^{\text{trans}}_{q,p}$, and $\bar{E}_p$; $\bar{r}_p$, ψ_p, $\chi_{q,p,m}(k)$; estimated values of $\beta_{j,p,m}(k)$, $\lambda^{\text{station}}_{j,m}(k)$, and $\alpha_{p,m}(k)$
Set initial values:
$U^0_{\text{ub}} \leftarrow \infty$, $U^0_{\text{lb}} \leftarrow -\infty$, $\bar{f}_p(k) \leftarrow 0$, $\bar{\delta}^{\text{absorb}}_p(k) \leftarrow 0$, $i \leftarrow 0$
Output: $f_p(k)$, $\delta^{\text{absorb}}_p(k)$
while $U^i_{\text{ub}} - U^i_{\text{lb}} \geq \alpha$ **do**
  $i \leftarrow i + 1$
  Solve (5) using $\bar{f}_p(k)$ and $\bar{\delta}^{\text{absorb}}_p(k)$
  if (5) is feasible and bounded **then**
    Obtain J_{dsp}, $\bar{\mathbf{u}}_{e1}(k_0)$, and $\bar{\mathbf{u}}_{e2}(k_0)$
    Update upper bound:
    $U^i_{\text{ub}} \leftarrow \min\left(U^{i-1}_{\text{ub}}, J^i_{\text{dsp}}\right)$
    Add optimality cut (7) using extreme points
  else if (5) is feasible but unbounded **then**
    Compute extreme rays $\bar{\mathbf{r}}_{q1}(k_0)$ and $\bar{\mathbf{r}}_{q2}(k_0)$
    Add feasibility cut (6) using extreme rays
  end if
  Solve (8) to obtain new $\bar{f}_p(k)$, and $\bar{\delta}^{\text{absorb}}_p(k)$
  Update lower bound:
  $U^i_{\text{lb}} \leftarrow \min\left(U^{i-1}_{\text{lb}}, J^i_{\text{mp}}\right)$
end while

feasible solution to the master problem is used for the dual sub-problem of the next iteration. Since the solution to the master problem is no longer optimal, the master problem no longer provides a valid lower bound. Instead, the ϵ-optimal Benders algorithm terminates when the master problem cannot produce a feasible solution. The master problem is then turned into a feasibility problem in the form of (9) instead of an optimization problem and hence is generally easier to handle, especially for large-scale problems.

$$\begin{aligned} &\mathbf{g}^{\mathrm{T}}(k_0)\mathbf{y}(k_0) + \eta \leq U_{\text{ub}}(1-\epsilon) \\ \text{s.t.} \quad &(4\text{e}), (6), (7) \end{aligned} \tag{9}$$

where $\epsilon \in (0, 1)$ is the slackness variable. A higher value for ϵ might result in faster convergence to the solution at the cost of a potentially worse solution.

A potential drawback of the ϵ-optimal Benders decomposition algorithm is that it may require more iterations than the classical Benders decomposition algorithm, as the non-optimal solutions to the master problem may also lead to non-optimal Benders cuts.

The detailed procedure of ϵ-optimal Benders decomposition-based train departure frequency optimization algorithm is shown in Algorithm 2.

IV. Case Study

In this section, we conduct a case study to compare the Benders decomposition-based algorithms for train departure frequency optimization.

A. Set-up

The metro network that is used for the case study is shown in Fig. 1, which consists of 21 stations, 60 platforms, and 6

Algorithm 2 ϵ-optimal Benders decomposition-based train departure frequency optimization algorithm

Input: α, ζ, N, P, S, $\theta^{\text{trans}}_{q,p}$, $\bar{E}_p$, and ϵ; $\bar{r}_p$, ψ_p, $\chi_{q,p,m}(k)$; estimated values of $\beta_{j,p,m}(k)$, $\lambda^{\text{station}}_{j,m}(k)$, and $\alpha_{p,m}(k)$
Set initial values:
$U^0_{\text{ub}} \leftarrow \infty$, $\bar{f}_p(k) \leftarrow 0$, $\bar{\delta}^{\text{absorb}}_p(k) \leftarrow 0$, $i \leftarrow 0$
Output: $f_p(k)$, $\delta^{\text{absorb}}_p(k)$
while $U^i_{\text{ub}} \geq \alpha$ **do**
  $i \leftarrow i + 1$
  Solve (5) using $\bar{f}_p(k)$ and $\bar{\delta}^{\text{absorb}}_p(k)$
  if (5) is feasible and bounded **then**
    Obtain J_{dsp}, $\bar{\mathbf{u}}_{e1}(k_0)$, and $\bar{\mathbf{u}}_{e2}(k_0)$
    Update upper bound:
    $U^i_{\text{ub}} \leftarrow \min\left(U^{i-1}_{\text{ub}}, J^i_{\text{dsp}}\right)$
    Add optimality cut (7) using extreme points
  else if (5) is feasible but unbounded **then**
    Compute extreme rays $\bar{\mathbf{r}}_{q1}(k_0)$ and $\bar{\mathbf{r}}_{q2}(k_0)$
    Add feasibility cut (6) using extreme rays
  end if
  Solve (9)
  if (9) is feasible **then**
    Obtain new $\bar{f}_p(k)$, and $\bar{\delta}^{\text{absorb}}_p(k)$
  else if (9) is infeasible **then**
    Break while loop
  end if
end while

bidirectional lines. The number on top of each link in Fig. 1 represents the average travel time between two stations and is used to determine the parameters $\bar{r}_p$ and ψ_p.

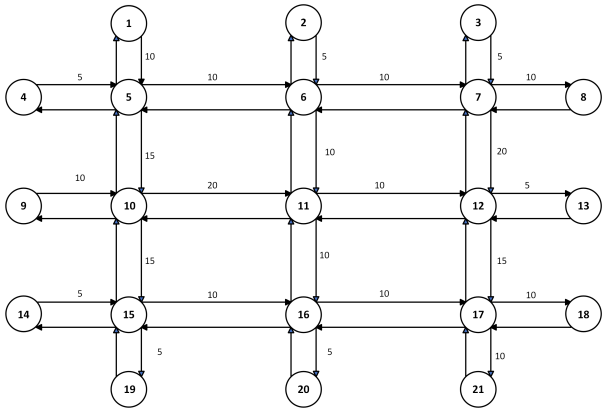


Fig. 1. Railway operations planning

We use time-varying passenger OD demands data, and passenger demands are considered to be constant for one period. The length of one period is set to 60 minutes. The average transfer time between two platforms is assumed to be equal. The cost per train run E_p is a function associated with the travel time $\bar{r}_p$. The values of the parameters are given in Table I.

In general, parameters $\alpha_{p,m}$ and $\chi_{q,p,m}$ can be estimated using historical data. However, since we use a fictional metro

TABLE I

PARAMETER VALUES

Parameter	Value
Stop criterion α	1
Transfer time θ_p^{trans}	1 [min]
Capacity $C_{\max}$	2000 passengers
Operational cost E_p	$2 \cdot \bar{r}_p$
Max departure frequency $f_p^{\max}$	20
Weight ζ	1000
Epsilon benders ϵ	0.01, 0.05, and 0.1

network, Dijkstra's algorithm [15] is utilized to compute the values for $\beta_{j,p,m}$, $\alpha_{p,m}$, and $\chi_{q,p,m}$, considering the average travel time between stations and assuming that passengers will always choose the shortest path to their destination in terms of time spent in the metro network.

We first apply the classical Benders decomposition-based algorithm for optimizing train departure frequencies. The total computation time consists of solving the dual sub-problem and master problem, updating the upper and lower bound, generating optimality and feasibility cuts, and obtaining extreme rays when necessary. The computation time for the ϵ-optimal Benders decomposition-based algorithm is computed in the same manner as for the classical Benders decomposition-based algorithm. Three different ϵ values are compared, i.e. 0.01, 0.05, and 0.1. For comparison, the resulting MILP problem is also directly solved by using `gurobi`, i.e., a state-of-the-art commercial solver for mixed integer programming problems.

The algorithms will be compared based on the objective function value and the required computation time. All the simulations are conducted using Matlab R2021a on a MacBook Pro 2017 with 2.3 GHz Dual-core Intel Core i5 processor and 8GB RAM. For the direct MILP algorithm, we use the commercial solver `gurobi` v9.5.2rc0 (mac64[x86]). Simulations are done for different planning time windows, i.e., from 2 ($N = 2$) to 6 hours ($N = 6$). The time limit for solving the resulting train departure frequency optimization problem for all the algorithms is set to 2 hours.

B. Results

The simulation results for all methods can be seen in Table II, where N.A. is used to indicate that no solution was found within 2 hours. For the sake of simplicity, we use "Gurobi" for the results obtained by solving the MILP problem using `gurobi`, "Benders" to denote the classical Benders decomposition algorithm, and "ϵ-Benders" to denote the ϵ-optimal Benders decomposition algorithm.

From Table II, we can find that when $N = 2$, `gurobi` has a better performance than both Benders decomposition-based methods. However, this changes when $N = 4$; the solution time of `gurobi` is significantly higher than that of both the Benders algorithm and the ϵ-Benders algorithm. For $N = 4$ and $N = 6$, both `gurobi` and the classical Benders decomposition algorithm cannot find the solution within 2 hours. The ϵ-optimal Benders decomposition algorithm outperforms the classical Benders decomposition algorithm when $N = 6$ in terms of solution time; this is because the master problem increases significantly in computation complexity with each added feasibility or optimality cut. When $N = 6$, the classical Benders decomposition algorithm cannot find the solution within 2 hours due to the master problem taking too long. The ϵ-optimal Benders decomposition can significantly reduce the computational complexity of the master problem.

TABLE II

COMPARISON OF DIFFERENT METHODS

N	Method	Objective function value	CPU time [s]
2	`gurobi`	1.70×10^5	16.1
	Classical Benders	1.70×10^5	66.9
	ϵ-Benders (0.01)	1.70×10^5	79.4
	ϵ-Benders (0.05)	1.71×10^5	70.6
	ϵ-Benders (0.1)	1.87×10^5	69.2
4	`gurobi`	4.00×10^5	6230.4
	Classical Benders	4.00×10^5	388.9
	ϵ-Benders (0.01)	4.00×10^5	358.4
	ϵ-Benders (0.05)	4.14×10^5	305.1
	ϵ-Benders (0.1)	4.14×10^5	291.3
6	`gurobi`	N.A.	N.A.
	Classical Benders	N.A.	N.A.
	ϵ-Benders (0.01)	6.80×10^5	1771.2
	ϵ-Benders (0.05)	7.07×10^5	714.4
	ϵ-Benders (0.1)	7.14×10^5	687.9

TABLE III

SIMULATION RESULTS FOR BENDERS DECOMPOSITION APPROACHES

N	Method	Iterations	t_{sub} [s]	t_{ray} [s]	t_{mas} [s]
2	Classical Benders	43	24.8	33.9	8.2
	ϵ-Benders (0.01)	51	32.4	36.5	10.4
	ϵ-Benders (0.05)	47	26.7	35.4	8.4
	ϵ-Benders (0.1)	44	25.3	35.9	7.9
4	Classical Benders	85	96.1	145.6	147.3
	ϵ-Benders (0.01)	113	141.3	158.8	58.3
	ϵ-Benders (0.05)	99	108.6	157.9	38.6
	ϵ-Benders (0.1)	97	103.3	149.9	38.0
6	Classical Benders	N.A	N.A.	N.A.	N.A.
	ϵ-Benders (0.01)	169	279.2	349.2	1142.8
	ϵ-Benders (0.05)	151	235.5	349.4	129.4
	ϵ-Benders (0.1)	147	237.4	355.7	94.8

To further illustrate the results, the number of iterations and the total time spent in each part of the algorithm are given in Table III. The evolution process of the different algorithms for $N = 4$ is also given. The convergence of the upper and lower bound is shown in Fig.2 for the classical Benders decomposition algorithm. The upper bound of the classical Benders decomposition algorithm changes only once. The dual sub-problem is unbounded for all other iterations. Since the ϵ-optimal Benders decomposition algorithm does not produce a valid lower bound, only the evolution of the upper bound is provided. The evolution of the upper bound for the different ϵ values is displayed in Fig.3 The upper bound of the ϵ-optimal Benders decomposition algorithm changes several times; the lower the value of ϵ, the more times the upper bound changes. As the master problem of the ϵ-optimal Benders decomposition algorithms becomes a feasibility problem, the computation time of solving the

master problem is reduced.

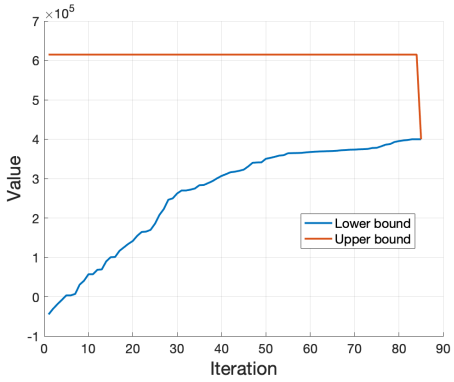


Fig. 2. Convergence upper bound and lower bound of classical Benders decomposition algorithm ($N = 4$)

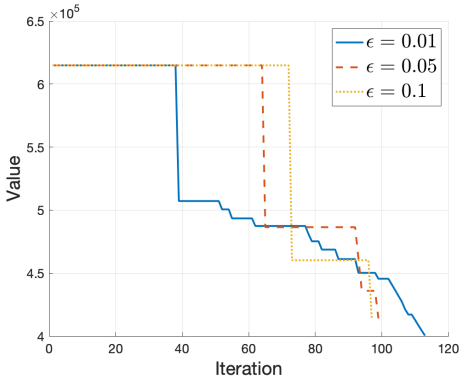


Fig. 3. Evolution of upper bound of ϵ-optimal Benders decomposition algorithm for different ϵ values ($N = 4$)

The simulation shows that the ϵ-optimal Benders decomposition algorithm is suitable for real-time optimization of train departure frequencies in metro networks. When simulating for multiple cycles, increasing the number of (integer) variables and constraints leads to long solution times for `gurobi`. While the classical Benders decomposition approach outperforms `gurobi` in terms of solution time when the number of cycles is four or higher, the high number of feasibility cuts required before the solution is found leads to a computationally complex master problem, which takes too long to solve to be effective in real-time applications. The ϵ-optimal Benders decomposition algorithm has been shown to be able to provide a solution fast enough for real-time use at the cost of some accuracy. By changing the value for ϵ, train operations can make a balanced trade-off between solution time and performance.

V. Conclusions

The real-time optimization of the departure frequencies in metro networks can be formulated as a mixed-integer linear programming problem. This paper has applied the Benders decomposition approach to reduce the computational burden of the train departure frequency optimization problem. To further improve the efficiency of the Benders decomposition-based approach, an ϵ-optimal strategy is used, which reduces the solution time by turning the master problem of the Benders decomposition into a feasibility problem. Simulation results indicate that the Benders-decomposition-based methods can reduce the computational time of train departure frequency optimization problems when the problem size increases. The ϵ-optimal Benders decomposition algorithm can further reduce the solution time of the classical Benders decomposition algorithm when the problem size increases.

In the future, we will focus on further reducing the computation time of Benders decomposition-based approaches. The potential approaches are to improve the efficacy of feasibility cuts and generate them based on the problem's structure.

References

[1] Y. Wang, B. Ning, T. Tang, T. J. van den Boom, and B. De Schutter, "Efficient real-time train scheduling for urban rail transit systems using iterative convex programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3337–3352, 2015.

[2] E. Hassannayebi, S. H. Zegordi, and M. Yaghini, "Train timetabling for an urban rail transit line using a lagrangian relaxation approach," *Applied Mathematical Modelling*, vol. 40, no. 23-24, pp. 9892–9913, 2016.

[3] E. Barrena, D. Canca, L. C. Coelho, and G. Laporte, "Single-line rail rapid transit timetabling under dynamic passenger demand," *Transportation Research Part B: Methodological*, vol. 70, pp. 134–150, 2014.

[4] X. Liu, A. Dabiri, Y. Wang, and B. De Schutter, "Modeling and efficient passenger-oriented control for urban rail transit networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3325–3338, 2023.

[5] A. De-Los-Santos, G. Laporte, J. A. Mesa, and F. Perea, "Simultaneous frequency and capacity setting in uncapacitated metro lines in presence of a competing mode," *Transportation Research Procedia*, vol. 3, pp. 289–298, 2014.

[6] D. Canca, E. Barrena, A. De-Los-Santos, and J. L. Andrade-Pineda, "Setting lines frequency and capacity in dense railway rapid transit networks with simultaneous passenger assignment," *Transportation Research Part B: Methodological*, vol. 93, pp. 251–267, 2016.

[7] X. Liu, A. Dabiri, and B. De Schutter, "Timetable scheduling for passenger-centric urban rail networks: Model predictive control based on a novel absorption model," in *2022 IEEE Conference on Control Technology and Applications (CCTA)*, IEEE, 2022, pp. 1147–1152.

[8] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[9] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The Benders decomposition algorithm: A literature review," *European Journal of Operational Research*, vol. 259, no. 3, pp. 801–817, 2017.

[10] Y. Hu, S. Li, M. M. Dessouky, L. Yang, and Z. Gao, "Computationally efficient train timetable generation of metro networks with uncertain transfer walking time to reduce passenger waiting time: A generalized Benders decomposition-based method," *Transportation Research Part B: Methodological*, vol. 163, pp. 210–231, 2022.

[11] F. Leutwiler, G. B. Filella, and F. Corman, "Accelerating logic-based Benders decomposition for railway rescheduling by exploiting similarities in delays," *Computers & Operations Research*, vol. 150, p. 106 075, 2023.

[12] K. Keita, P. Pellegrini, and J. Rodriguez, "A three-step Benders decomposition for the real-time railway traffic management problem," *Journal of Rail Transport Planning & Management*, vol. 13, p. 100 170, 2020.

[13] H. P. Williams, *Model Building in Mathematical Programming*. John Wiley & Sons, 2013.

[14] A. M. Geoffrion and G. W. Graves, "Multicommodity distribution system design by Benders decomposition," *Management Science*, vol. 20, no. 5, pp. 822–844, 1974.

[15] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

Bibliography

[1] Gurobi optimizer reference manual. https://www.gurobi.com/wp-content/plugins/hd_documentations/documentation/9.0/refman.pdf.

[2] IBM ILOG CPLEX optimization studio v12.10.0 documentation. https://www.ibm.com/docs/en/icos/12.10.0?topic=SSSA5P_12.10.0/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.htm.

[3] E. Barrena, D. Canca, L. C. Coelho, and G. Laporte. Exact formulations and algorithm for the train timetabling problem with dynamic control. *Computers & Operations Research*, 44:66–74, 2014.

[4] E. Barrena, D. Canca, L. C. Coelho, and G. Laporte. Single-line rail rapid transit timetabling under dynamic passenger demand. *Transportation Research Part B*, 70:123–150, 2014.

[5] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(3):238–252, 1962.

[6] S. Binder, Y. Maknoon, and M. Bierlaire. The multi-objective railway timetable rescheduling problem. *Transportation Research Part C: Emerging Technologies*, 78:78–94, 2017.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[8] S. Bucak and T. Demirel. Train timetabling for a double track urban rail transit line under dynamic passenger demand. *Computers & Industrial Engineering*, 163:107858, 2022.

[9] T. Bunsen, J. Dunant, M. Gorner, R. Isaac, G. Kamiya, S. Scheffer, R. Schuitmaker, and J. Tattini. The future of rail. *International Energy Agency*, 2019.

[10] V. Cacchiani, J. Qi, and L. Yang. Robust optimization models for integrated train stop planning and timetabling with passenger demand uncertainty. *Transportation Research Part B*, 136:1–29, 2020.

[11] D. Canca, E. Barrena, A. De-Los-Santos, and J. Andrade-Pineda. Setting lines frequency and capacity in dense railway rapid transit networks with simultaneous passenger assignment. *Transportation Research Part B: Methodological*, 93:251–267, 2016.

[12] D. Canca, M. Sabido, and E. Barrena. A rolling stock circulation model for railway rapid transit systems. *Transportation Research Procedia*, 3:680–689, 2014.

[13] G. Cavone, L. Blenkers, T. J. J. van den Boom, M. Dotoli, C. Seatzu, and B. De Schutter. Railway disruption: a bi-level rescheduling algorithm. *6th International Conference on Control, Decision and Information Technologies*, pages 54–59, 2019.

[14] Z. Chen, S. Li, H. Zhang, Y. Wang, and L. Yang. Distributed model predictive control for real-time train regulation of metro line based on Dantzig-Wolfe decomposition. *Transportmetrica B: Transport Dynamics*, 11(1), 2023.

[15] G. Codato and M. Fischetti. Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research*, 4:756–766, 2006.

[16] F. Corman, A. D'Ariano, A. D. Marra, D. Pacciarelli, and M. Sama. Integrating train scheduling and delay management in real-time railway traffic control. *Transportation Research Part E*, 105:213–239, 2017.

[17] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C: Emerging Technologies*, 20(1):79–94, 2012.

[18] G. B. Dantzig and M. N. Thapa. *Linear Programming 2*. Springer Series in Operations Research and Financial Engineering, 2003.

[19] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.

[20] A. De-Los-Santos, G. Laporte, J. A. Mesa, and F. Perea. Simultaneous frequency and capacity setting in uncapacitated metro lines in presence of a competing mode. *Transportation Research Procedia*, 3:289–298, 2014.

[21] G. Feng, P. Xu, D. Cui, X. Dai, H. Liu, and Q. Zhang. Multi-stage timetable rescheduling for high-speed railways: a dynamic programming approach with adaptive state generation. *Complex and Intelligent Systems*, pages 1407–1428, 2021.

[22] B. Fortz and M. Poss. An improved Benders decomposition applied to a multi-layer network design problem. *Operations Research Letters*, 37:359–364, 2009.

[23] A. Geoffrion and G. Graves. Multicommodity distribution system design by Benders decomposition. *Management Science*, 20(5):822–844, 1974.

[24] E. Hassannayebi and S. H. Zegordi. Variable and adaptive neighborhood search algorithms for rail rapid transit timetabling problem. *Computers & Operations Research*, 78:439–453, 2017.

[25] E. Hassannayebi, S. H. Zegordi, and M. Yaghini. Train timetabling for an urban rail transit line using a Lagrangian relaxation approach. *Applied Mathematical Modelling*, 40:9892–9913, 2016.

[26] Y. Hu, S. Li, M. M. Dessouky, L. Yang, and Z. Gao. Computationally efficient train timetable generation of metro networks with uncertain transfer walking time to reduce passenger waiting time: A generalized Benders decomposition-based method. *Transportation Research Part B: Methodological*, 163:210–231, 2022.

[27] M. Jiang, H. Li, X. Xu, S. Xu, and J. Miao. Metro passenger flow control with station-to-station cooperation based on stop-skipping and boarding limiting. *Journal of Central South University*, 24:236–244, 2017.

[28] K. Keita, P. Pellegrini, and J. Rodriguez. A three-step Benders decomposition for the real-time railway traffic management problem. *Journal of Rail Transport Planning Management*, 13:525–540, 2020.

[29] F. Leutwiler and F. Corman. A logic-based Benders decomposition for microscopic railway timetable planning. *European Journal of Operational Research*, 303:525–540, 2022.

[30] F. Leutwiler, G. B. Filella, and F. Corman. Accelerating logic-based Benders decomposition for railway rescheduling by exploiting similarities in delays. *Computers Operations Research*, 150:106075, 2023.

[31] S. Li, B. De Schutter, L. Yang, and Z. Gao. Robust model predictive control for train regulation in underground railway transportation. *IEEE Transactions on Control Systems Technology*, 24:1075–1083, 2015.

[32] X. Liu, A. Dabiri, and B. De Schutter. Timetable scheduling for passenger-centric urban rail networks: Model predictive control based on a novel absorption model. *2022 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1147–1152, 2022.

[33] X. Liu, A. Dabiri, Y. Wang, and B. De Schutter. Modeling and efficient passenger-oriented control for urban rail transit networks. *IEEE Transactions on Intelligent Transportation Systems*, 24(3):3325–3338, 2023.

[34] S. Long, L. Meng, X. Luan, A. Trivella, J. Miao, and F. Corman. A stochastic programming approach for scheduling extra metro trains to serve passengers from uncertain delayed high-speed railway trains. *Journal of Advanced Transportation*, 2020.

[35] X. Luan, B. De Schutter, L. Meng, and F. Corman. Decomposition and distributed optimization of real-time traffic management for large-scale railway networks. *Transportation Research Part B*, 141:72–97, 2020.

[36] X. Luan, Y. Wang, B. De Schutter, L. Meng, G. Lodewijks, and F. Corman. Integration of real-time traffic management and train control for rail networks - part 1: Optimization problems and solution approaches. *Transportation Research Part B*, 115:41–71, 2018.

[37] X. Luan, Y. Wang, B. De Schutter, L. Meng, G. Lodewijks, and F. Corman. Integration of real-time traffic management and train control for rail networks - part 2: Extensions towards energy-efficient train operations. *Transportation Research Part B*, 115:72–94, 2018.

[38] T. L. Magnanti and R. T. Wong. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.

[39] P. Mo, L. Yang, A. D'Ariano, J. Yin, Y. Yao, and Z. Gao. Energy-efficient train scheduling and rolling stock circulation planning in a metro line: a linear programming approach. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3621–3633, 2020.

[40] P. Mo, L. Yang, Y. Wang, and J. Qi. A flexible metro train scheduling approach to minimize energy cost and passenger waiting time. *Computers & Industrial Engineering*, 132:412–432, 2019.

[41] J. Naoum-Sawaya and S. Elhedhli. An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, 210:33–55, 2013.

[42] T. Nishi, Y. Muroi, and M. Inuiguchi. Column generation with dual inequalities for railway crew scheduling problems. *Public Transport*, 3:25–42, 2011.

[43] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259:801–817, 2017.

[44] S. S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley and Sons, 2009.

[45] T. Schettini, O. Jabali, and F. Malucelli. A Benders decomposition algorithm for demand-driven metro scheduling. *Computers and Operations Research*, 138:525–540, 2022.

[46] Z. Su, A. Jamshidi, A. Núñez, S. Baldi, and B. De Schutter. Integrated condition-based track maintenance planning and crew scheduling of railway networks. *Transportation Research. Part C: Emerging Technologies*, 105:359–384, 2019.

[47] H. Sun, J. Wu, H. Ma, X. Yang, and Z. Gao. A bi-objective timetable optimization model for urban rail transit based on the time-dependent passenger volume. *IEEE Transactions On Intelligent Transportation Systems*, 20:604–615, 2018.

[48] X. Sun, S. Zhang, H. Dong, Y. Chen, and H. Zhu. Optimization of metro train schedules with a dwell time model using the Lagrangian duality theory. *IEEE Transactions On Intelligent Transportation Systems*, 16:1285–1293, 2015.

[49] M. Temürhan and H. Lol Stipdonk. Coaches and road safety in Europe; an indication based on available data 2007-2016. *Wetenschappelijk onderzoek verkeersveiligheid*, 2019.

[50] 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*, 118:193–227, 2018.

[51] Y. Wang, B. De Schutter, T. J. J. van den Boom, B. Ning, and T. Tang. Real-time scheduling for single lines transit in urban rail transit systems. *Proceedings of the 2013 IEEE International Conference on Intelligent Transportation Systems*, pages 1334–1339, 2013.

[52] Y. Wang, B. De Schutter, T. J. J. van den Boom, B. Ning, and T. Tang. Efficient bi-level approach for urban rail transit operation with stop-skipping. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2658–2670, 2014.

[53] Y. Wang, B. Ning, T. Tang, T. J. J. van den Boom, and B. De Schutter. Efficient real-time train scheduling for urban rail transit systems using iterative convex programming. *IEEE Transactions on Intelligent Transportation systems*, 16(6):3337–3352, 2015.

[54] Y. Wang, T. Tang, B. Ning, T. J. J. van den Boom, and B. De Schutter. Passenger demand oriented train scheduling for an urban rail transit network. *Transportation Research Part C*, 60:1–23, 2015.

[55] Y. Wang, S. Zhu, L. Yang, and B. De Schutter. Hierarchical model predictive control for on-line high-speed railway delay management and train control in a dynamic operations environment. *IEEE Transactions on Control Systems Technology*, 30(6):1–16, 2022.

[56] H. P. Williams. *Model Building in Mathematical Programming.* John Wiley & Sons, 2013.

[57] X. Yang, X. Li, Z. Gao, H. Wang, and T. Tang. A cooperative scheduling model for timetable optimization in subway systems. *IEEE Transactions On Intelligent Transportation Systems*, 12(1):438–447, 2013.

[58] Y. Yang and J. M. Lee. A tighter cut generation strategy for acceleration of Benders decomposition. *Computers and Chemical Engineering*, 44:84–93, 2012.

[59] S. Yanga, J. Wua, X. Yang, F. Liao, D. Li, and Y. Wei. Analysis of energy consumption reduction in metro systems using rolling stop-skipping patterns. *Computers Industrial Engineering*, 127:129–142, 2019.

[60] X. Yanga, X. Lib, B. Ninga, and T. Tan. An optimisation method for train scheduling with minimum energy consumption and travel time in metro rail systems. *Transportmetrica B: Transport Dynamics*, 3(2):79–98, 2015.

[61] 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*, 97:182–213, 2017.

[62] C. Ying, A. Chow, Y. Wang, and K. Chin. An actor-critic deep reinforcement learning approach for metro train scheduling with rolling stock circulation under stochastic demand. *Transportation Research Part B*, 140:210–235, 2021.

[63] C. Ying, A. Chow, Y. Wang, and K. Chin. Adaptive metro service schedule and train composition with a proximal policy optimization approach based on deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 21:1–12, 2021.

[64] Y. Yue, J. Han, S. Wang, and X. Liu. Integrated train timetabling and rolling stock scheduling model based on time-dependent demand for urban rail transit. *Computer-Aided Civil and Infrastructure Engineering*, 31:856–873, 2017.

[65] P. Zhang, Z. Sun, and X. Liu. Optimized skip-stop metro line operation using smart card data. *Journal of Advanced Transportation*, 2017.

[66] X. Zhou and M. Zhong. Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B*, 41:320–341, 2007.

[67] Y. Zhu and R. Goverde. Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions. *Transportation Research Part B*, 123:149–181, 2019.

[68] Y. Zhu and R. Goverde. Dynamic railway timetable rescheduling for multiple connected disruptions. *Transportation Research Part C*, 125, 2021.

[69] Y. Zhu, B. Mao, L. Liu, and M. Li. Timetable design for urban rail line with capacity constraints. *Computers & Operations Research*, 2015.

[70] Y. Zhu, H. Wang, and R. Goverde. Reinforcement learning in railway timetable rescheduling. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems ITSC*, 2020.

Glossary

List of Acronyms

MINLP	Mixed-Integer Nonlinear Programming
MILP	Mixed-Integer Linear Programming
OD	Origin-Destination
BB	Branch-and-Bound
LP	Linear Programming
NLP	Nonlinear Programming

List of Symbols

Symbols related to the passenger absorption model

N	Number of periods in planning time window
P	Set of platforms
S	Set of stations
T	Period length [s]
p	Platform index
m	Station index
k	Period index
J_{time}	Time spent in the metro network by passengers in a given time planning [s] window
J_{cost}	Operational costs of dispatched trains in the metro network in a given time planning window
J	Objective function
ζ	Weighting term
$n_p(k)$	Number of passengers waiting at platform p at the start of period period k
$n_{p,m}(k)$	Number of passengers waiting at platform p with destination m at the start of period k
$n_p^{\text{depart}}(k)$	Number of passengers departing from platform p during period k
$n_{p,m}^{\text{depart}}(k)$	Number of passengers departing from platform p with destination m during period k
$n_p^{\text{arr,tra}}(k)$	Number of transferring passengers arriving at platform p during period k
$n_{p,m}^{\text{arr,tra}}(k)$	Number of transferring passengers arriving at platform p with destination m during period k
$f_p(k)$	Train departure frequency at platform p during period k
$l_p(k)$	Train departure frequency of the starting platform of the line on which platform p lies
$C_p(k)$	Remaining capacity on a train at platform p during period k
$n_{p,m}^{\text{train}}(k)$	Number of passengers on board of trains at platform p with destination m during period k
$n_{q,p,m}^{\text{trans}}(k)$	Number of transferring passengers arriving at platform q to transfer to platform p with destination m during period k
$n_p^{\text{absorb}}(k)$	Number of passengers who board a train at platform p during period k
$n_{p,m}^{\text{absorb}}(k)$	Number of passengers who board a train at platform p with destination m during period k
$n_p^{\text{wait}}(k)$	Number of passengers waiting for a train at platform p during period k
$n_{p,m}^{\text{wait}}(k)$	Number of passengers waiting for a train at platform p with destination m during period k
$n_{p,m}^{\text{alight}}(k)$	Number of passengers alighting a train at platform p with destination m during period k
$\delta_p^{\text{absorb}}(k)$	Auxiliary (binary) variables used to transform the min function
$z_p^{\text{wait}}(k)$	Auxiliary variable used to transform the min function
$z_p^{\text{cap}}(k)$	Auxiliary variable used to transform the min function

$f_p^{\max}$	Maximum train departure frequency of platform p
$\lambda_p(k)$	Passenger arrival rate at platform p during period k [passengers/s]
$\lambda_{p,m}(k)$	Passenger arrival rate at platform p with destination m during period k [passengers/s]
$C_{\max}$	Maximum capacity of a train and departing from another platform p of that same line
$\alpha_{p,m}(k)$	Relative fraction of passengers that board a train at platform p whose destination is station m
$\chi_{q,p,m}(k)$	Relative fraction of passengers arriving at platform q with destination m, who will transfer from platform q to platform p
p^{pla}	Predecessor platform of platform p
$\text{sta}(p)$	Set of platforms belong to the same station as platform p
$\bar{E}_p$	Average operational costs associated with dispatching a train from platform p towards the next platform on its route
ψ_p	Average time for a train between departing from the starting platform of a line [s]
θ_p^{trans}	Average travel time for passengers transferring from platform p to another platform [s]
$\theta_{q,p}^{\text{trans}}$	Average travel time for passengers transferring from platform p to platform q [s]
$\bar{r}_p$	Average travel time for a train travelling from platform p to the next platform [s]

Symbols related to Benders decomposition

i	Iteration index
J_{dsp}	Objective function dual sub-problem
J_{mas}	Objective function master problem
J_{opt}	Optimality cut
J_{feas}	Feasibility cut
U_{ub}	Upper bound
U_{lb}	Lower bound
η	Auxiliary variable used for master problem
$u_p^{\text{capacity}}(k)$	Dual variable associated with constraint (3-5b)
$u_{p,m}^{\text{number}}(k)$	Dual variable associated with constraint (3-5c)
$u_p^{\text{wait}}(k)$	Dual variable associated with constraint (3-5d)
$u_{p,m}^{\text{absorb}}(k)$	Dual variable associated with constraint (3-5e)
$u_{p,m}^{\text{train}}(k)$	Dual variable associated with constraint (3-5f)
$u_{p,m}^{\text{alight}}(k)$	Dual variable associated with constraints (3-5g) and (3-5h)
$u_{p,m}^{\text{depart}}(k)$	Dual variable associated with constraint (3-5i)
$u_{q,p,m}^{\text{trans}}(k)$	Dual variable associated with constraint (3-5j)
$u_{p,m}^{\text{arrive, trans}}(k)$	Dual variable associated with constraint (3-5k)
$u_p^{\text{absorb}}(k)$	Dual variable associated with constraint (3-5l)
$u_p^{\text{wait, auxiliary}}(k)$	Dual variable associated with constraint (3-5m)
$u_p^{\text{capacity, auxiliary}}(k)$	Dual variable associated with constraint (3-5n)
Ω	Feasible space of the dual sub-problem
$\mathbb{E}$	Set of extreme points of Ω
$\mathbb{Q}$	Set of extreme rays of Ω