

OPTIMIZATION MODELS FOR ENERGY-EFFICIENT RAILWAY
TIMETABLES

by

Shuvomoy Das Gupta

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

© Copyright 2016 by Shuvomoy Das Gupta

Abstract

Title: Optimization Models for Energy-efficient Railway Timetables

Author: Shuvomoy Das Gupta

Master of Applied Science

Graduate Department of Electrical and Computer Engineering

University of Toronto

2016

This thesis presents two novel optimization models to calculate energy-efficient railway timetables in a railway network. The first optimization model is a mixed integer programming one, which saves energy by maximizing the total overlapping time between the braking and accelerating phases of suitable train pairs. However, it suffers from some limitations associated with \mathcal{NP} -hard computational complexity and modeling of energy saving strategy. To overcome the limitations of the first model, we propose a second optimization model consisting of two stages. The first stage of this model minimizes the total energy consumed by all trains and the second stage maximizes the transfer of regenerative braking energy between suitable train pairs. Both of these stages are solvable in polynomial time, compared to other existing models, which are \mathcal{NP} -hard. The two-stage model has proven to be very effective in practice and has been incorporated into an industrial railway timetable compiler.

Acknowledgements

Life, unlike a railway timetable, cannot be scheduled in advance. Yet, when I look back on the last few years, I understand that life has been very kind to me. I have had the great pleasure of meeting some amazing people, who have influenced me deeply with their wisdom and compassion. Here I express my gratitude to them.

First, I would like to thank my supervisor Professor Lacra Pavel. I feel privileged to have had the chance to work in your research group and to learn from your generous spirit. Thanks for your guidance and care. Thanks for patiently reading through my reports and for being tolerant of the numerous typos and grammatical mistakes in them.

I owe a great debt to Kevin Tobin, my collaborator at Thales Canada. Thanks for your inspiration, encouragement and clarity. Thanks for the discussions on optimization models, computational complexity, the happy gene, the nature of consciousness, the *narrative*, books, television shows and what not! I am very grateful to have witnessed in you a concrete example of the kind, conscientious, and generous person I aspire to be.

I was extremely fortunate to find some very good friends at the University of Toronto, who embody the core values I cherish. They were always appreciative of my few good qualities and tolerant of my many flaws. I would like to thank Alireza Mohammadi, Connor Holmes, Hu Hong, Joseph Dongchan Lee, Justin Boutilier, Sajjad Najafi, Vahid Roshanaei, Xinquan Qu, Yao Hong Kok, Zachary Kroeze, Zhaowei Wang and Zhen Dai. I apologize if I have forgotten to mention any friend's name.

Thanks to Tzu Chi Youth Group at the University of Toronto for giving me the opportunity to volunteer at Evangel Hall and Heritage Senior Home. I have been very inspired by the compassion and equanimity of the volunteers of the group, and I really hope that I have become a bit more compassionate myself by association.

Thanks to University of Toronto Operations Research Group (UTORG) for allowing me to run a discussion group on advanced convex optimization and several tutorials.

Finally, some words of gratitude to my parents. *Ma* and *Baba*, you gave me the gift of precious human life, provided me unconditional love, protection and support, and never asked for anything in return. Thank you.

Contents

Contents	iv
List of Figures	vii
List of Tables	ix
Nomenclature	x
1 Introduction	1
1.1 Motivation	1
1.2 Literature review	4
1.3 Contribution	5
1.4 Organization	6
2 Background on mathematical optimization	7
2.1 Mathematical notation	7
2.2 Optimization problems	8
2.2.1 Convex optimization problems	9
2.2.2 Linear optimization (LP) problems	10
2.2.3 Mixed integer optimization (MIP) problems	12
2.3 Some important concepts of mathematical optimization	12
2.3.1 Modeling disjunctions in optimization problems	12
2.3.2 Convex envelope and its importance in optimization problems	13
2.3.3 Least-squares problems	14
2.3.4 Locally optimal vs. globally optimal points	15
2.3.5 Robust optimization models	15
2.3.6 Hypograph (epigraph) approach	16
2.4 Lumping methods to calculate the area under a curve	17
2.5 Computational tractability of optimization problems	18

3	Modeling a feasible railway timetable	20
3.1	Definition of various terms in a railway network	20
3.2	Notation to describe a railway network	22
3.3	Constraints in a railway network	24
3.3.1	Trip time constraint	24
3.3.2	Dwell time constraint	25
3.3.3	Connection constraint	25
3.3.4	Headway constraint	26
3.3.5	Total travel time constraint	26
3.3.6	Domain of the event times	27
3.4	Relation between passenger demand, headway and number of trains . . .	27
4	A robust mixed integer optimization model	28
4.1	Relevant event times of a train around a platform	28
4.2	Synchronization process between suitable train pairs (SPSTP)	29
4.3	Structure of the optimization model	35
4.4	Modeling overlapping time for right events	38
4.5	Modeling overlapping time for left events	43
4.6	Full optimization model	44
4.7	Limitations of the model	45
5	Numerical experiments for the robust MIP model	47
5.1	Shanghai Metro network	47
5.2	Docklands Light Railway	52
6	A two-stage linear programming model	55
6.1	Motivation behind the model	56
6.2	Stage one of the linear programming model	56
6.2.1	Converting the initial notation into an equivalent constraint graph	57
6.2.2	Description of the first stage of the linear programming model . .	58
6.3	Stage two of the linear programming model	62
6.3.1	Keeping the total energy consumption at minimum	63
6.3.2	Maximizing the utilization of regenerative energy of braking trains	63
6.3.3	Description of the overlapped area between power graphs	64
6.3.4	Description of the second stage of the linear programming model .	65
6.4	Limitations of the optimization model	67

7	Numerical experiments for the two-stage LP model	69
7.1	Shanghai Metro network	69
7.2	Comparison with the robust mixed integer programming model	70
7.3	Integration with Thales timetable compiler	72
8	Conclusion and future works	74
8.1	Conclusion	74
8.2	Future works	75
	Bibliography	81

List of Figures

1.1	Optimal speed profile of a train	3
2.1	Illustration of the $\frac{1}{e}$ heuristic	17
3.1	Trip time constraint associated with a crossing-over	25
4.1	Rectangular blocks to indicate temporal events of train t around platform i (top) and the associated speed profile (bottom)	30
4.2	For platform pair $(i, j) \in \Omega$ and train $t \in \mathcal{T}_i$ the figure shows temporally closest train to t 's left \overleftarrow{t} (corresponding to the green temporal blocks) and temporally closest train to t 's right \overrightarrow{t} (corresponding to the red temporal blocks). Here $\tilde{t} = \overrightarrow{t}$	32
4.3	Applying the optimization strategy for the case $\tilde{t} = \overrightarrow{t}$, when there is no constraint. The left hand side and right hand side of the arrows correspond to the initial timetable and the final timetable respectively. The shaded regions represent the overlapping time between trains.	33
4.4	Applying the optimization strategy for the case $\tilde{t} = \overleftarrow{t}$, when there is no constraint. The left hand side and right hand side of the arrows correspond to the initial timetable and the final timetable respectively. The shaded regions represent the overlapping time between trains.	34
4.5	If we simultaneously maximize the overlapping time between acceleration phase of \overleftarrow{t} with the braking phase of t and the acceleration phase of t with the braking phase of \overrightarrow{t} , after applying such a strategy \overrightarrow{t} and \overleftarrow{t} might be very close to each other which is not desired in practical situations. The top and bottom part of the figure show the pre-optimization and the post-optimization scenario respectively for such a case.	36

4.6	If we simultaneously maximize the overlapping time between acceleration phase of t with the braking phase of \overleftarrow{t} and the acceleration phase of $\overrightarrow{t} = \overleftarrow{t'}$ with the braking phase of t' , after applying such a strategy, computational effort would be spent in trying shift the events associated with $\overrightarrow{t} = \overleftarrow{t'}$ to opposite directions, where only the first one is achievable.	37
4.7	Graphical illustration of the constraint described by Equation (4.5), which prevents the occurrence of the cross-marked situations	39
4.8	All possible overlapping times between the accelerating phase of train t and braking phase of its temporally closest train	40
5.1	First railway network considered for numerical study	48
5.2	Second railway network considered for numerical experiment	52
6.1	A flow-chart of the two stages of the optimization model	55
6.2	Example of a very simple railway network.	58
6.3	The constraint graph for the railway network of Figure 6.2.	59
6.4	Applying $\frac{1}{e}$ heuristic to power graphs	64
7.1	The integration architecture of the two stage model with Thales timetable compiler	72

List of Tables

5.1	Results of the numerical study (running time 1200 s) for the railway network in Figure 5.1	50
5.2	Results of the numerical study (running time 1200 s) for the network in Figure 5.2	53
7.1	Results of the numerical study performed to line 8 of Shanghai Metro network	71
8.1	Speed limit for line 8 of Shanghai Metro network	77

Nomenclature

$[\underline{\chi}_{ij}^{tt'}, \bar{\chi}_{ij}^{tt'}]$ The connection window between train t at platform i and train t' at platform j

$[\underline{\delta}_i^t, \bar{\delta}_i^t]$ The dwell time window for train t at platform i

$[\underline{\kappa}_{ij}^{tt'}, \bar{\kappa}_{ij}^{tt'}]$ The trip time window for train t on the crossing-over (i, j)

$[\underline{\tau}_{\mathcal{P}}^t, \bar{\tau}_{\mathcal{P}}^t]$ The total travel time window for train t to traverse its train path

$[\underline{\tau}_{ij}^t, \bar{\tau}_{ij}^t]$ The trip time window for train t from platform i to platform j

$[\underline{h}_i^{tt'}, \bar{h}_i^{tt'}]$ The headway time window between train t and t' at or from platform i

$[l_{ij}, u_{ij}]$ The time window associated with arc (i, j) of the constraint graph

α_i^t The duration of the acceleration phase of train t around platform i

$\bar{\mathcal{A}}$ The set of arcs in the constraint graph

$\bar{\mathcal{A}}_{\text{trip}}$ The set of all arcs in the constraint graph associated with trip time constraints

$\bar{\mathcal{N}}$ The set of all nodes in the constraint graph

β_i^t The duration of the braking phase of train t around platform i

χ The set of all platform pairs situated at the same interchange stations

env f Convex envelope of a function f

1 A column vector with all components being 1

card (x) The number of nonzero components of a vector x

dom f The domain of a function $f : \mathbf{R}^n \rightarrow \mathbf{R}$

epi f The epigraph of a function $f : C \rightarrow \mathbf{R}$

hypo f	The hypograph of a function $f : C \rightarrow \mathbf{R}$
\mathbf{R}	The set of real numbers
\mathbf{Z}	The set of complex numbers
\mathcal{A}	The set of all tracks in the railway network
\mathcal{A}^t	The set of all tracks visited by the train t in chronological order
\mathcal{B}_{ij}	The set of all train pairs involved in turn-around events on the crossing-over (i, j)
\mathcal{C}_{ij}	The set of connecting train pairs for a platform pair $(i, j) \in \chi$
\mathcal{E}	The set of all synchronization processes between suitable train pairs
\mathcal{H}_{ij}	The set of train-pairs who move along that track (i, j) successively in order of their departures
\mathcal{L}	The set of indices of all train lines present in the railway network
\mathcal{N}	The set of all platforms in the railway network
\mathcal{N}^t	The set of all platforms visited by the train t in chronological order
\mathcal{P}^t	The path of a train $t \in \mathcal{T}$
\mathcal{S}	The set of all stations in the railway network
\mathcal{T}	The set of all the trains in the railway network
\mathcal{T}_i	The set of all the trains which arrive at, dwell and then depart from platform i
Ω	The set of every distinct opposite platform pair powered by same substation
\overleftarrow{t}	The temporally closest train to the left of the train t
\overrightarrow{t}	The temporally closest train to the right of the train t
$\overleftarrow{\mathcal{E}}$	The set of all synchronization processes between suitable train pairs associated with left events
$\overrightarrow{\mathcal{E}}$	The set of all synchronization processes between suitable train pairs associated with right events
$\overleftarrow{\sigma}_{ij}^{tt}$	Overlapping time associated with the left event $(i, j, t, \overleftarrow{t}) \in \overleftarrow{\mathcal{E}}$

$\sigma_{ij}^{\vec{t}}$	Overlapping time associated with the right event $(i, j, t, \vec{t}) \in \vec{\mathcal{E}}$
\tilde{t}	The temporally closest train to the train t
∇_i^t	The relative distance of a_i^t from the regenerative alignment point
φ	The set of all crossing-overs in the railway network, where turn-around events take place
Δ_j^t	The relative distance of the consumptive alignment point from d_j^t
a_i^t	The arrival time of train $t \in \mathcal{T}$ at platform $i \in \mathcal{N}^t$
$c_{ij}(x_i - x_j) + b_{ij}$	Affine approximation for f_{ij}
d_j^t	The departure time of train $t \in \mathcal{T}$ from platform $j \in \mathcal{N}^t$
e_i	The vector with all components zero except for the i th component
f_{ij}	Energy consumption associated with the trip $(i, j) \in \bar{\mathcal{A}}_{\text{trip}}$
x_i	The arrival or departure time of some train from a platform in the constraint graph

Chapter 1

Introduction

In recent years, much emphasis is being placed on efficient use of energy in transportation systems. The biggest reasons behind it are the depletion of conventional sources of energy and the fluctuation in production of the renewable sources of energy. The railway system is an integral part of transportation systems in most countries. Among the multiple levels of railway planning process, timetabling is very important. A timetable not only allows the passengers to plan ahead for their trips, it also enables the railway management to satisfy the operational feasibility of the railway network. This thesis is focused on the design and investigation of optimization models and solution methodologies to construct energy-efficient railway timetables.

This introductory chapter is organized as follows. At first we discuss the motivation behind calculating energy-efficient railway timetables in Section 1.1. In the next section, we examine the relevant literature in calculating such timetables. In Section 1.3, we discuss the aim and contribution of this thesis and Section 1.4 presents its organization.

1.1 Motivation

Among all of the public transport modes, railway is often preferred by passengers for providing higher capacity and safety [2]. So, the performance of public transportation depends largely on the performance of the railway system. Reliable performance of a railway network is closely related to the quality of the railway timetable used by the railway management. Formally, a railway timetable is a data structure that contains the arrival and departure time of every train to and from all the platforms it visits over a fixed time period. An energy-efficient timetable designs these arrival and departure times of the trains to reduce the energy consumption and increase the energy saving in the railway network. Before presenting the motivations behind calculating an energy-

efficient timetable, we briefly discuss how a train consumes and produces energy during a trip.

In most railway networks, trains use electricity as their primary source of energy [44] and many of them are equipped with regenerative braking technology [39]. When a train makes a trip from an origin platform to a destination platform, its optimal speed profile consists of four phases: 1) accelerating, 2) speed holding, 3) coasting and 4) braking [20], as shown in Figure 1.1. in a qualitative manner. Most of the energy required by the train is consumed during the accelerating phase. During the speed holding phase the energy consumption is negligible compared to accelerating phase, and during the coasting phase there is no need for energy. When the train brakes, it produces energy by using a regenerative braking mechanism. This energy is called *regenerative braking energy*. Naturally, reducing the energy consumption of accelerating trains and/or proper utilization of regenerative energy of braking trains can increase energy-efficiency of the railway network by a significant amount.

A feasible strategy to utilize the regenerative braking energy of a train, that can be implemented with the current technology [10], is to synchronize its braking phase with the accelerating phase of another nearby train operating under the same electrical substation (see Section 4.2 for details). A positive overlapping time that arises from such a synchronization process enables transfer of the regenerative braking energy of the first train to the second one via the overhead contact line or a third rail [10], and can save the electrical energy that would be lost otherwise.

The motivations behind designing energy-efficient timetables are as follows. *First*, an energy-efficient railway timetable can reduce the energy costs associated with railway operation. *Second*, though a lot of emphasis on use of energy from renewable resources is being put in recent years, non-renewable energy sources such as coal still provide the significant portion. For example, Canada, in spite of being a world leader in the production and use of energy from renewable resources, can only manage a mere 18.9% of the total energy supply from renewable sources [1, page 121]. The non-renewable sources are the primary driving force behind the emission of green house gasses. So, improved energy efficiency of railway timetables can reduce the emissions of green house gasses. *Finally*, the railway management can increase the energy-efficiency of the railway network just by enforcing an energy-efficient timetable without requiring any change in the infrastructure.

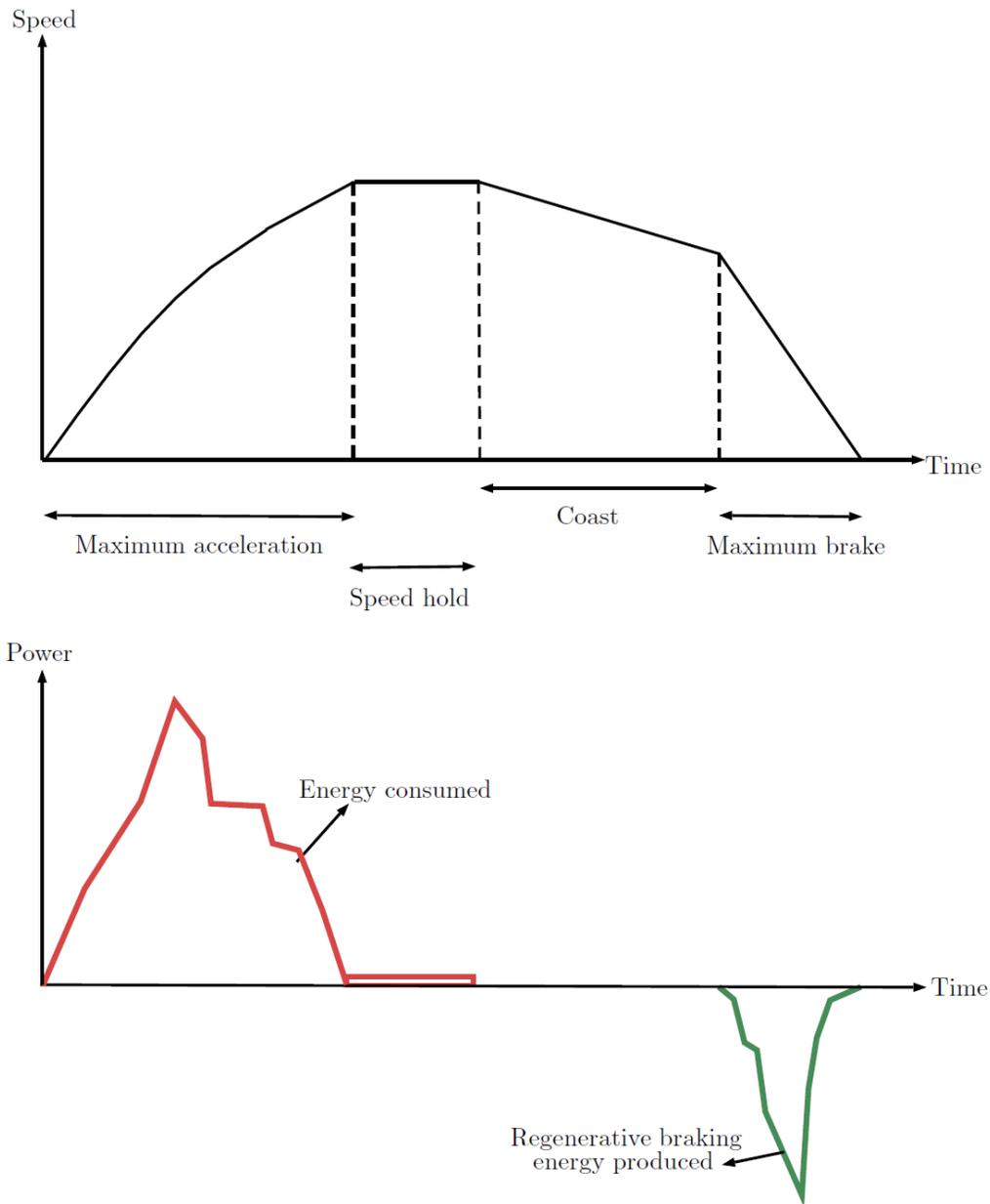


Figure 1.1: Optimal speed profile of a train

1.2 Literature review

The general timetabling problem in a railway network has been studied extensively over the past three decades [17]. However, very few results exist that can calculate energy-efficient timetables. Now we discuss the related research. We classify the related work as follows. The first two papers are mixed integer programming model, the next three are models based on meta-heuristics and the last one is an analytical study.

A Mixed Integer Programming (MIP) model, applicable only to single train-lines, is proposed by Peña-Alcaraz et al. [37] to maximize the total duration of all possible synchronization processes between all possible train pairs. The model is then applied successfully to line three of the Madrid underground system. However, the model can have some drawbacks. First, considering all train pairs in the objective will result in a computationally intractable problem even for a moderate sized railway network. Second, for a train pair in which the associated trains are far apart from each other, most, if not all, of the regenerative energy will be lost due to the transmission loss of the overhead contact line. Finally, the model assumes that the durations of braking and accelerating phases stay the same with varying trip times, which is not the case in reality.

The work in [11] proposes a more tractable MIP model, applicable to any railway network, by considering only train pairs suitable for regenerative energy transfer. The optimization model is applied numerically to the Dockland Light Railway and shows a significant increase in the total duration of the synchronization process. Although such increase, in principle, may increase the total savings in regenerative energy, the actual energy saving is not directly addressed. Similar to [37], this model too, assumes that even if the trip time changes, the duration of the associated braking and accelerating stay the same.

Other relevant works implement meta-heuristics. The work in [27] implements genetic algorithm to calculate timetables that maximize the utilization of regenerative energy while minimizing the tractive energy of the trains. Numerical studies for the model is implemented to Beijing Metro Yizhuang Line of China showing notable increase in energy efficiency. The work in [46] presents a cooperative integer programming model to utilize the use of regenerative energy of trains and proposes genetic algorithm to solve it. Similar to [27], this numerical studies have been performed to Beijing Metro Yizhuang Line of China, though the improvement is stated in the increase in overlapping time only. The work in [26] presents a nonlinear integer programming model which is solved using simulated annealing. The numerical experiments have been conducted for the island line of the mass transit system in Hong Kong.

An insightful analytical study of a periodic railway schedule appears in [28]. The model uses the KKT conditions to calculate and analyze the properties of an energy-efficient timetable. The resultant analytical model is then applied to Beijing Metro Yizhuang Line of China numerically, which shows that the model can reduce the net energy consumption considerably.

1.3 Contribution

The aim of this thesis is to design and investigate optimization models and solution methods to construct energy-efficient railway timetables. The main research questions addressed and studied in this thesis are the following:

- *How can mathematical optimization models be used to calculate energy-efficient railway timetables?*
- *What are the constraints that need to be addressed to design a functional railway timetable?*
- *What are the criteria of an energy efficient timetable and how can they be modeled in a computationally tractable manner?*
- *How can the energy-efficient timetables arising from the optimization models be computed in a reasonable amount of time?*

To that end, we propose two novel optimization models to calculate energy-efficient railway timetables in railway networks in this thesis.

The first optimization model utilizes the regenerative braking energy of trains in a railway network. This optimization model is a robust mixed integer programming one (see Section 2.2.3 for details). It calculates a railway timetable which maximizes the total overlapping time between the braking and accelerating phases of suitable train pairs to facilitate the transfer of regenerative energy to accelerating trains. We prove that all possible cases arising from possible overlapping between a suitable train pair are modeled accurately by our model via hypograph approach and interval algebra. We apply our optimization model to different instances of two railway networks for time horizon spanning six hours. Compared to the original timetables, the overlapping time increases significantly in the optimal timetables. In the first railway network we have access to relevant energy information to calculate the relative reduction in effective energy consumption

and we find that there is significant increase in utilization of regenerative energy for every instance compared to the existing timetables. However, the first model suffers from some limitations associated with computational complexity and energy saving strategy.

To overcome the limitations of the first model, we propose a second optimization model. It is a novel two-stage linear optimization model. This too calculates energy-efficient railway timetables, but in two stages. The *first* stage of the optimization model minimizes the total energy consumed by all trains subject to the constraints present in the railway network. The problem can be formulated as a linear program, with the optimal value attained by an integral vector. The *second* stage of the optimization model uses the optimal trip time from the first optimization model and maximizes the transfer of regenerative braking energy between suitable train pairs. Both the stages of our optimization model are linear programs, whereas the optimization models in the first model and in the related works are \mathcal{NP} -hard. The second model has proven to be very effective in calculating energy-efficient timetables in practice. Code based on the model has been incorporated into the railway timetable compiler (**Thales Timetable Compiler**) of Thales Inc, which has the largest installed base of communication-based train control systems worldwide. **Thales Timetable Compiler** is used by many railway management systems worldwide including: the Canada Line and Skytrain in Vancouver, Canada, Docklands Light Railway in London, UK, the West Rail Line and Ma On Shan Line in Hong Kong, China, the Red Line and Green Line in Dubai, United Arab Emirates, the Kelana Jaya Line in Kuala Lumpur, Malaysia, and the East-West Line in Singapore.

1.4 Organization

This thesis is organized as follows. Chapter 2 presents the relevant background on railway network and mathematical optimization. In Chapter 3 we discuss the constraints required to calculate a feasible railway timetable. In Chapter 4 we present the first mathematical optimization model to calculate energy-efficient railway timetable. In the next chapter, we apply our optimization model to different instances of two railway networks for time horizon spanning six hours and discuss the results. Chapter 6 presents the two-stage linear optimization model. In Chapter 7 we apply our model to different instances of an existing railway network spanning a full working day and describe the results. Chapter 8 presents the conclusion and future research direction.

Chapter 2

Background on mathematical optimization

In this chapter we introduce concepts associated with mathematical optimization relevant to this thesis. Section 2.2 describes the structure of an optimization problem and then discusses different classes of optimization problems. In Section 2.3 we discuss some important concepts in mathematical optimization that we have used in this thesis. Section 2.4 is about approximating area under a curve, for which we do not have any mathematical description available. In Section 2.5 we discuss the tractability of optimization problems and the importance of constructing a tractable optimization model in practice.

2.1 Mathematical notation

Every set described in this thesis is strictly ordered and finite unless otherwise specified. The set-cardinality (number of elements of the set) and the i th element of such a set C is denoted by $|C|$ and $C(i)$, respectively. The set of real numbers and integers are expressed by \mathbf{R} and \mathbf{Z} , respectively; subscripts $+$ and $++$ attached with either set denote non-negativity and positivity of the elements respectively. A column vector with all components one is denoted by $\mathbf{1}$. The symbol \preceq stands for componentwise inequality between two vectors. The symbols \wedge and \vee stand for conjunction and disjunction, respectively. The number of nonzero components of a vector x is called cardinality of that vector and is denoted by $\mathbf{card}(x)$. Note that, cardinality of a vector is different from set-cardinality. The i th unit vector e_i is the vector with all components zero except for

the i th component which is one, *i.e.*,

$$e_i = (0, \dots, \underbrace{1}_{\substack{\text{ith} \\ \text{position}}}, \dots, 0) \in \mathbf{R}^n.$$

The epigraph of a function $f : C \rightarrow \mathbf{R}$ (where C is any set) denoted by **epi** f is the set of input-output pairs that f can achieve along with anything above, *i.e.*,

$$\mathbf{epi} f = \{(x, t) \in C \times \mathbf{R} \mid x \in C, t \geq f(x)\}.$$

Similarly, the hypograph of a function $f : C \rightarrow \mathbf{R}$ (where C is any set) denoted by **hypo** f is the set of input-output pairs that f can achieve along with anything below, *i.e.*,

$$\mathbf{hypo} f = \{(x, t) \in C \times \mathbf{R} \mid x \in C, t \leq f(x)\}.$$

The domain of a function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is denoted by **dom** f and is defined by

$$\mathbf{dom} f = \{x \in \mathbf{R}^n \mid f(x) < +\infty\}.$$

2.2 Optimization problems

A mathematical optimization problem consists of a set of *constraints* and an *objective* to be optimized over a *decision variable*. The set of constraints represents the characteristics of a system (abstract or real) we are interested to study. The objective corresponds to a quantity or quantities associated with the system which we want to maximize or minimize. The decision variable, which is often a vector comprising of many components, is the unknown to be calculated. Our goal is to find an optimal decision variable which maximizes (minimizes) the objective.

A mathematical optimization problem has the following form

$$p^* = \begin{pmatrix} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & h_i(x) = 0, \quad i = 1, \dots, p. \end{pmatrix} \quad (2.1)$$

Here the vector $x = (x_1, \dots, x_n)$ is the decision variable of the problem. The function $f_0 : \mathbf{R}^n \rightarrow \mathbf{R}$ is the objective function. The inequalities $f_i(x) \leq 0$ are called the *inequality constraints* and the equalities $h_i(x) = 0$ are called the *equality constraints*. The set of constraints comprises of both the equality constraints and inequality constraints. The

functions $f_i : \mathbf{R}^n \rightarrow \mathbf{R}$ and $h_i : \mathbf{R}^n \rightarrow \mathbf{R}$ are called the inequality constraint functions and the equality constraint functions, respectively. Any $y \in \mathbf{R}^n$ that obeys all the constraints is called a *feasible solution* to the optimization problem. The optimal value of the problem is denoted by p^* . Our goal is to find an x^* that minimizes f_0 among all feasible solutions, such an x^* is called a *globally optimal solution* (or just *optimal solution*) to (2.1). So an optimization problem, the *problem* involves finding an optimal solution. An algorithm which can find an optimal solution reliably is called an *optimization algorithm*. There can be more than one optimal solutions to an optimization problem. Depending on the structure of the functions $f_0, f_1, \dots, f_m, h_1, \dots, h_p$ we have different types of optimization problems, *e.g.*, linear optimization problem, mixed integer optimization problem, convex optimization problem *etc.* For the purpose of this thesis, it suffices to confine our attention to convex, linear and mixed integer optimization problems.

2.2.1 Convex optimization problems

First we need the following definitions.

Definition 2.1. (*Convex set and convex hull*) A set C is convex, if the line segment between any two points in C lies in C , *i.e.*, for any $x_1, x_2 \in C$ and any $\lambda \in [0, 1]$ we have $\lambda x_1 + (1 - \lambda)x_2 \in C$. The convex hull of any set C , denoted by $\mathbf{conv} C$, is the set containing all convex combinations of points in C . Consequently, if C is nonconvex, then its best convex outer approximation is $\mathbf{conv} C$, as it is the smallest convex set containing C .

Definition 2.2. (*Convex function*) A function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is convex, if its domain is a convex set and for all $x, y \in \mathbf{dom} f$, and $\lambda \in [0, 1]$ we have

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y).$$

Now, in problem (2.1) if the inequality constraint functions f_0, f_1, \dots, f_m are convex, and the equality constraint functions h_1, \dots, h_p are affine, *i.e.*, $h_i(x) = a_i^T x - b_i$ for any $i = 1, \dots, m$, then the resultant optimization problem is called a convex optimization problem. Surprisingly, a lot of problems can be solved via convex optimization, though it can be difficult to recognize. Reliable and efficient algorithms exist to solve convex optimization problems [8].

2.2.2 Linear optimization (LP) problems

A linear programming problem is a special type of convex optimization problem. It arises a lot in practical problems. Linear programming problem has three forms: *generic*, *general* and *standard* form.

Generic linear optimization problems

In a generic linear optimization problem, we have a cost vector $c \in \mathbf{R}^n$, and our goal is to minimize a linear cost function $c^T x = \sum_{i=1}^n c_i x_i$ subject to a set of linear and inequality constraints. The decision variable is $x \in \mathbf{R}^n$. Let, M_{\leq} , M_{\geq} and $M_{=}$ denote three finite index sets. For every i belonging to any one these sets we have a vector $a_i \in \mathbf{R}^n$ and a scalar b_i to construct the constraint. Let N_{\geq} and N_{\leq} be the subsets of $\{1, \dots, n\}$ to denote which variables x_j are constrained to be nonnegative and nonpositive, respectively. Then the generic linear optimization problem can be described as follows.

$$\begin{aligned}
 & \text{minimize} && c^T x \\
 & \text{subject to} && a_i^T x \geq b_i, \quad i \in M_{\geq}, \\
 & && a_i^T x \leq b_i, \quad i \in M_{\leq}, \\
 & && a_i^T x = b_i, \quad i \in M_{=}, \\
 & && x_j \geq 0, \quad j \in N_{\geq}, \\
 & && x_j \leq 0, \quad j \in N_{\leq}.
 \end{aligned} \tag{2.2}$$

General form linear optimization problems

We can convert the problem (2.2) into a more compact form, which is known as general form linear optimization problem. Any equality constraint $a_i^T x = b_i$ can be written as two inequality constraints: $a_i^T x \geq b_i$ and $a_i^T x \leq b_i$. Any inequality constraint $a_i^T x \leq b_i$ can be written as $(-a_i)^T x \geq -b_i$. Any constraint of the form $x_j \leq 0$ and $x_j \geq 0$ can be written as $(-e_j)^T x \geq 0$ and $e_j^T x \geq 0$. So, the constraint set in any linear optimization problem can be written exclusively in terms of constraints of the form $a_i^T x \geq b_i$. If we have m number of such constraints in total, then the constraints $a_i^T x \geq b_i, \quad i = 1, \dots, m$ can be written compactly as $Ax \succeq b$, where $b = (b_1, \dots, b_m) \in \mathbf{R}^m$ and

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} \in \mathbf{R}^{m \times n},$$

and the symbol \succeq stands for componentwise inequality. So, a general form linear optimization problem can be written as:

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Ax \succeq b. \end{aligned} \tag{2.3}$$

From an algorithmic point of view, the general form linear optimization problem is not very convenient to work with.

Standard form linear optimization problems

There is another representation of linear optimization problem, which is pivotal in the development of *simplex* algorithm, known as standard form problem. Any optimization model, when solved by simplex algorithm, is converted to the standard form first by the solver [6, Chapter 3]. It has the following form:

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && Ax = b, \\ & && x \succeq 0. \end{aligned} \tag{2.4}$$

Any general form linear optimization problem can be brought into a standard form linear optimization problem as follows.

- We eliminate the unrestricted variables. We use the fact that any real number can be written as the difference of two nonnegative numbers. Any unrestricted variable x_j , which is allowed to have nonnegative or nonpositive values, can be replaced by $x_j^+ - x_j^-$, where x_j^+ and x_j^- are newly introduced nonnegative decision variables. A nonpositive variable $x_i \leq 0$ is essentially $(-x_i) \geq 0$.
- We eliminate the inequality constraints of the form $a_i^T x \leq b_i$ ($a_i^T x \geq b_i$ can be written as $(-a_i)^T x \leq (-b_i)$), where x has already been made nonnegative using step 1. We introduce a nonnegative slack variable s_i to write the constraint as $a_i^T x + s_i = b_i$.

Converting a standard form linear optimization problem into a general form one is straightforward. problem (2.4) is essentially equivalent to problem (2.2), with $M_{\succeq} = \emptyset$, $M_{\leq} = \emptyset$, $M_{=} = \{1, \dots, m\}$, $N_{\succeq} = \{1, \dots, n\}$ and $N_{\leq} = \emptyset$.

2.2.3 Mixed integer optimization (MIP) problems

A surprisingly large class of practical problem can be represented by MIP problems. A MIP problem (in standard form) has the following form:

$$\begin{aligned} & \text{minimize} && c^T x + d^T y \\ & \text{subject to} && Ax + Fy = b, \\ & && x \succeq 0, y \succeq 0. \end{aligned} \tag{2.5}$$

where $c \in \mathbf{R}^n, d \in \mathbf{R}^d, A \in \mathbf{R}^{m \times n}, F \in \mathbf{R}^{m \times d}$ and $b \in \mathbf{R}^m$, and the decision variables are $x \in \mathbf{Z}^n$ and $y \in \mathbf{R}^d$. Problem (2.5) has both discrete and continuous variables. Computationally MIP problems involve many times as much calculations to solve compared to similar sized convex optimization problems, they belong to a class of \mathcal{NP} -hard problem (see Section 2.5).

2.3 Some important concepts of mathematical optimization

In this section we discuss several independent but important concepts associated with mathematical optimization that have been instrumental the development of this thesis.

2.3.1 Modeling disjunctions in optimization problems

Modeling many systems requires disjunctive constraints. For example, consider scheduling of jobs on a machine, where one of them has to be scheduled before the other. This represents a disjunctive scenario. In such applications, the feasible solutions lie in the union of two or more polyhedra.

Assume we have the disjunctive following constraints

$$\begin{aligned} h_1(x) &\geq 0, \vee \\ h_2(x) &\geq 0, \vee \\ &\vdots \\ h_n(x) &\geq 0. \end{aligned}$$

Here a feasible x satisfies at least one of the constraints, but not necessarily both. To model such constraints, we can introduce n new binary decision variables $\lambda_1, \dots, \lambda_n \in$

$\{0, 1\}$, and n positive numbers M_1, \dots, M_n , and rewrite them as follows:

$$\begin{aligned} h_1(x) + M_1(1 - \lambda_1) &\geq 0, \\ h_2(x) + M_2(1 - \lambda_2) &\geq 0, \\ &\vdots \\ h_n(x) + M_n(1 - \lambda_n) &\geq 0, \\ \sum_{i=1}^n \lambda_i &\geq 1, \\ \lambda_i &\in \{0, 1\}, \quad i = 1, \dots, n. \end{aligned}$$

The numbers M_i should be large enough such that when $\lambda_i = 0$, the i th constraint becomes inactive.

2.3.2 Convex envelope and its importance in optimization problems

When the feasible set of an optimization problem is convex, but the objective function f_0 is not, the problem is nonconvex, and solving it can be a computationally challenging task. An appealing idea in such a case is to replace f_0 with some convex underestimator on the constraint set. Naturally finding the best convex underestimator of the function is of interest in this regard. The best convex approximation of a nonconvex function $f : C \rightarrow \mathbf{R}$ (where C is any set) from below is given by its convex envelope $\mathbf{env} f$ on C . The function $\mathbf{env} f$ is the largest convex function that is an under estimator of f on C , *i.e.*,

$$\mathbf{env} f = \sup\{\tilde{f} : C \rightarrow \mathbf{R} \mid \tilde{f} \text{ is convex and } \tilde{f} \leq f\},$$

where \sup stands for the supremum, *i.e.*, the least upper bound of the set. The definition implies, $\mathbf{epi} \mathbf{env} f = \mathbf{conv} \mathbf{epi} f$.

Finding convex envelope of an arbitrary function is a hard problem in general. However, some important special cases are known, *e.g.*,

- *Monomial over unit hypercube.* Consider the function $f = x_1 x_2 \cdots x_n$, known as monomial, and $C = [0, 1]^n$, also known as the unit hypercube. Then

$$\mathbf{env} f = \max \left(0, 1 - n + \sum_{i=1}^n x_i \right).$$

- *Cardinality function over l_∞ ball.* Consider the cardinality function $\mathbf{card}(x)$ over the set

$$C = \left\{ x \in \mathbf{R}^n \mid \|x\|_\infty = \max_{i \in \{1, \dots, n\}} \{|x_i|\} \leq R \right\},$$

known as l_∞ ball. Then,

$$\mathbf{env} \mathbf{card}(x) = \frac{1}{R} \|x\|_1.$$

2.3.3 Least-squares problems

Consider the linear system $Ax = b$, $A \in \mathbf{R}^{m \times n}$ is a skinny and full column rank matrix ($m > n$), $b \in \mathbf{R}^m$ and $x \in \mathbf{R}^n$ is our unknown. This type of system is called over-determined set of linear equations, as there are more equations than unknowns. In such a system, there may not exist a feasible solution x satisfying $Ax = b$, and it makes sense to find an approximate solutions which makes the residual vector $r = Ax - b$ as small as possible. One measure of the size of a vector is its Euclidean norm. Thus our goal is to find an x^* such that it minimizes $\|r\|_2$. So, the problem in consideration is

$$\text{minimize } \|r\| = \|Ax - b\|_2 \quad ,$$

with x being the decision variable. As the function y^2 is strictly monotonically increasing for nonnegative y , the problem above is equivalent to minimizing the square of the Euclidean norm as follows.

$$\text{minimize } \|r\|_2^2 = \|Ax - b\|_2^2 \tag{2.6}$$

with x being the decision variable. Problem (2.6) is an unconstrained convex optimization problem that can be solved as follows. First note that, $\|r\|_2^2 = x^T A^T A x - 2b^T A x + b^T b$. Now we set the gradient of $\|r\|_2^2$ with respect to x equal to zero and find the least-squares solution denoted by x^* .

$$\begin{aligned} \nabla_x \|r\|_2^2 &= 2A^T A x^* - 2A^T b = 0 \\ \Leftrightarrow A^T A x^* &= A^T b \\ \Leftrightarrow x^* &= (A^T A)^{-1} A^T b. \end{aligned}$$

Note that $A^T A$ is invertible as A is full column rank.

2.3.4 Locally optimal vs. globally optimal points

A point x_{local}^* is locally optimal for problem (2.1) if there exists a number $R > 0$ such that the x_{local}^* is optimal for the problem

$$p_{\text{local}}^* = \left(\begin{array}{l} \text{minimize}_x \quad f_0(x) \\ \text{subject to} \quad f_i(x) \leq 0, \quad i = 1, \dots, m, \\ \quad \quad \quad h_i(x) = 0, \quad i = 1, \dots, p, \\ \quad \quad \quad \|x - x_{\text{local}}^*\|_2^2 \leq R^2 \end{array} \right).$$

So a locally optimal point minimizes the objective but only for nearby points on the feasible set. The value of the objective function at a locally optimal point is not necessarily same as the globally optimal value. Often a locally optimal point is of no practical interest to the user. In a general optimization problem finding a globally optimal point can be a challenge, as most algorithms tend to be trapped at a locally optimal point. In this regard, the class of convex optimization problems is different, as a locally optimal point is always globally optimal in a convex optimization problem.

2.3.5 Robust optimization models

Often we need to account for the presence of uncertainty in the data describing an optimization model. In such a case, we want to obtain a solution that is robust against this uncertainty. A *robust mixed integer optimization model* is a deterministic model such that *i*) it takes into account the presence of uncertainty in the data describing the optimization problem, *ii*) obtains solutions that are proven to be robust against such uncertainty, and *iii*) some of the decision variables of the model are integer valued. A robust formulation is the formal process which models the original problem with uncertainty into a robust one.

Consider a modified version of the generic mixed integer programming model (2.5) as follows:

$$\begin{aligned} & \text{minimize} \quad c^T x + d^T y \\ & \text{subject to} \quad (\forall a_i \in U_a^{(i)}) \quad (\forall f_i \in U_f^{(i)}) \quad a_i^T x + f_i^T y = b_i, \quad i = 1, \dots, m, \\ & \quad \quad \quad x \succeq 0, y \succeq 0. \end{aligned} \quad (2.7)$$

where $c \in \mathbf{R}^n, d \in \mathbf{R}^d, a_i^T \in \mathbf{R}^{1 \times n}, f_i^T \in \mathbf{R}^{1 \times d}$ and $b_i \in \mathbf{R}$ for $i = 1, \dots, m$, and the decision variables are $x \in \mathbf{Z}^n$ and $y \in \mathbf{R}^d$. In this simple version of a robust mixed

integer model, we assume that the individual rows, a_i^T and f_i^T are known to belong to given *confidence* sets, $U_a^{(i)}$ and $U_f^{(i)}$ for $i = 1, \dots, m$. As the name suggests, the confidence sets are sets of confidence for the coefficients for the model. The confidence sets may contain a infinite number number of elements. The main idea in robust optimization is to express the constraints associated with data uncertainty in some an alternative but *tractable* form. Depending on the type and structure of the confidence sets, it may be difficult to come up with a tractable representation for the robust optimization model. However, there are notable exceptions for which a tractable representation is indeed possible, and in such a case we say that the robust model is *computationally tractable* (see Section 2.5).

2.3.6 Hypograph (epigraph) approach

Consider the following generic maximization problem

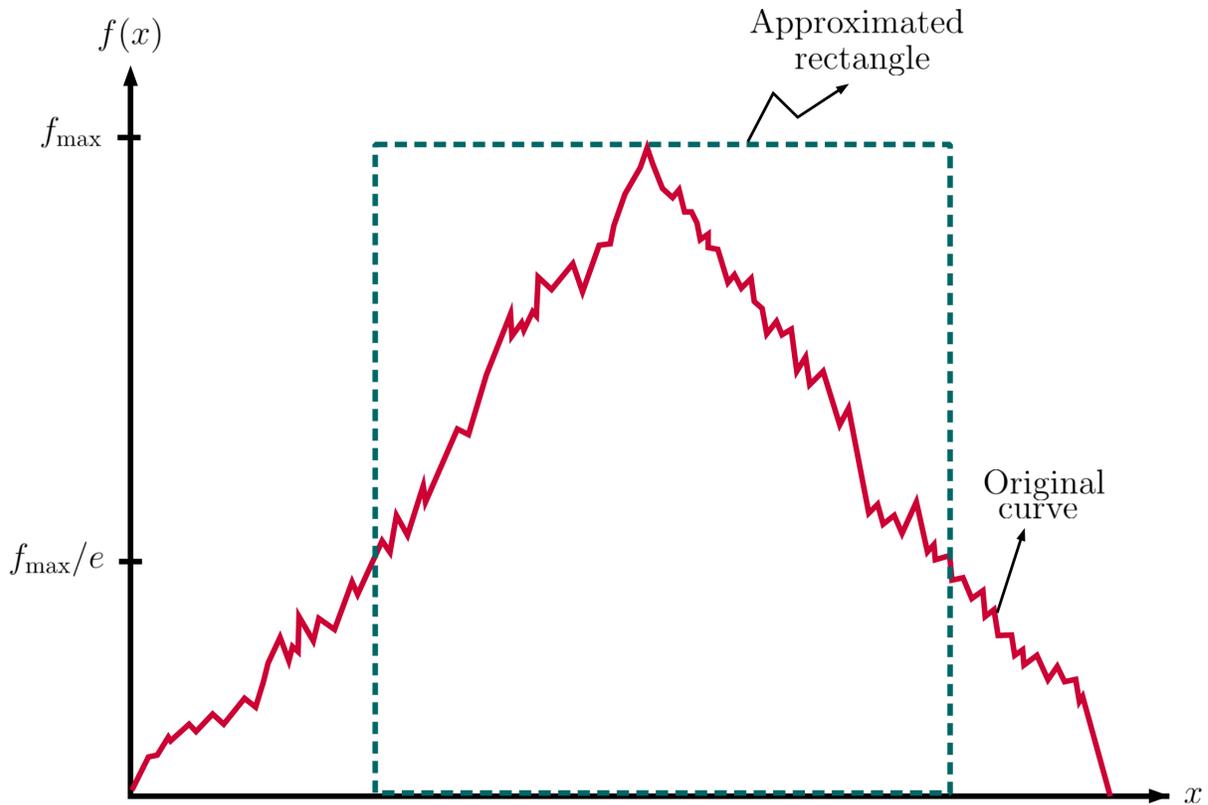
$$\begin{aligned} & \text{maximize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & && h_i(x) = 0, \quad i = 1, \dots, p, \end{aligned} \tag{2.8}$$

with $x \in \mathbf{R}^n$ being the decision variable. The *hypograph approach* remodels the problem into the following equivalent problem:

$$\begin{aligned} & \text{maximize} && t \\ & \text{subject to} && t \leq f(x) \\ & && f_i(x) \leq 0, \quad i = 1, \dots, m, \\ & && h_i(x) = 0, \quad i = 1, \dots, p, \end{aligned} \tag{2.9}$$

where the decision variable is now $(x, t) \in \mathbf{R}^{n+1}$, with $(x, t) \in \mathbf{hypo} f$ (hence the name hypograph approach). Note that the objective function $(\mathbf{0}, 1)^T(x, t)$ is a linear function. Hypograph approach is often convenient in modeling complicated objective function.

Similarly, when we are dealing with a minimization problem, we can work with an equivalent problem using *epigraph approach*. Let us replace maximize with minimize in problems (2.8) and (2.9). Then in the modified problems $(x, t) \in \mathbf{epi} f$ (hence the name epigraph approach).

Figure 2.1: Illustration of the $\frac{1}{e}$ heuristic

2.4 Lumping methods to calculate the area under a curve

Finding the area under a curve arises frequently in almost all branches of science and technology. Its enormous importance was one of the biggest, if not the biggest, practical motivations behind the development of integral calculus. Integral calculus can often find the areas under difficult curves exactly.

However, one of the underlying assumptions behind successful application of integral calculus is the exact knowledge of the function describing the curve. In practice, we may have little or no idea about the mathematical description of the associated curve. Rather, what is available at hand is the set of measurement points of the curve at different data points. So, the only way to find the area under such a curve is some numerical method. As the integral is dependent on the data associated with a specific instance, a numerical integration procedure gives little insight, can be time consuming, has little transfer value and tends to be very sensitive to small changes in the observation [33, Section 3.1].

A workaround to the issues described above is provided by *lumping methods*, which

almost always provide reasonable answers very quickly. Successful application of lumping methods goes back to the early days of spectroscopy, which was instrumental in the development of quantum mechanics [14]. A very important problem in spectroscopy is finding the area under the curve associated with the radiation absorption of a molecule at different wavelengths. Decades before digital chart recorders were invented, this area was calculated successfully using lumping methods [33, page 35].

When there is a dominant peak in the curve, one of the most successful lumping methods is the $\frac{1}{e}$ *heuristic*. In this heuristic, we approximate the curve under consideration by a rectangle as follows. The height of the rectangle is the height of the peak, and the width is the interval with extreme points corresponding to points dropped at $\frac{1}{e}$ of the peak. Such rectangles are very robust approximations to the original curve [33, page 33-34]. The area of the rectangle, which is the height times the width, tends to be very close to the area of the original curve. Figure 2.1 shows how the $\frac{1}{e}$ heuristic is used to approximate an arbitrary curve as a rectangle.

2.5 Computational tractability of optimization problems

In this section, we briefly discuss the tractability of optimization problems without entering a technical discussion on *computational complexity*. We refer the interested reader to the monograph [4] for a rigorous introduction to computational complexity. The computational tractability of an optimization problem can vary widely depending on the problem class it belongs to. Some problem classes, *e.g.*, finding a solution to a finite set of linear inequalities can be done very efficiently. However, some other classes of problems are extremely hard to deal with.

Size of any instance of a particular optimization problem is measured by the number of decision variables and constraints. An optimization problem is called *tractable*

- if a globally optimal solution can be found numerically in a reliable way for any instance, and
- if the growth of computational effort required grows gracefully (*e.g.*, logarithmically, polynomially) with the size of the problem.

If the growth in computational effort is a polynomial in the size of the optimization problem, then the associated optimization problem belongs to the class \mathcal{P} . For example, linear optimization problems discussed in 2.2.2 belongs to \mathcal{P} . For some optimization

problems there is no known way to find a globally optimal solution quickly, but if a candidate point is provided, it is possible to do the verification in polynomial time, then such problems belong to the class \mathcal{NP} (which stands for nondeterministic polynomial time) [4, Chapter 2]. A problem p is \mathcal{NP} -hard, if every problem in the class \mathcal{NP} can be reduced to p in polynomial time. Informally, a \mathcal{NP} -hard problem is at least as hard as the hardest problems in \mathcal{NP} . For example, mixed integer optimization problems described in 2.2.3 is \mathcal{NP} -hard [7, page 242].

It should be noted that the tractability of a problem often depends how the problem is formulated and modeled. Two formulations that address the same problem may not be of same tractability. A problem that may seem hard, even intractable, under a certain formulation may well become tractable under some other formulation with some more effort and intelligence. However, this may not always possible no matter how much effort is put in formulating the problem, *e.g.*, the traveling salesperson problem [4, page 40].

Chapter 3

Modeling a feasible railway timetable

This thesis is about maximizing the energy-efficiency of a feasible railway timetable. So the first question is: *what constraints do we need to consider to calculate a feasible timetable?* A feasible railway timetable needs to satisfy various requirements, which include safety regulations, service levels and restrictions that consider the operational feasibility of the railway management. In this chapter, we describe different constraints that need to be obeyed in a feasible railway timetable. First, in Section 3.1 we introduce the reader to various terms required to describe a railway network. Then in Section 3.2, we describe the notation to describe those terms. Section 3.3 describes all the constraints required to construct a feasible railway timetable. The constraints have been proven to cover most practical needs and form the constraint set for the optimization models described in Chapters 4 and 6. Section 3.4 describes the relation between passenger demand, headway and number of trains.

3.1 Definition of various terms in a railway network

In this section we define and explain the physical meanings of various terms associated with the description of a railway network. We do this for two reasons. First, though the terms are fairly standard in railway research literature [36, 29, 30], we have noticed that in some works, same terms have been used to denote different physical aspects of the railway network with out clarification, thus causing confusion. By explicitly defining the terms we avoid such confusions. Second, it helps the reader to be familiarized with different physical aspects of a general railway network.

- **Station:** A station is a place where passenger trains stop on a railway line, typically with platforms and buildings [12] .

- **Platform:** A platform is a raised structure along the side of a railway track where passengers get on and off trains at a station [12] . Along with this conventional definition, any intermediate stopping point or turn-around point is also called a platform in our report. A platform is associated with the direction of the trains arriving at and departing from that platform. Generally the direction of a train associated with a platform is fixed, *i.e.*, the trains will always arrive to and depart from that platform in the same direction, and we assume this to be the case in our report. A station generally has more than one platform.

The concept of *opposite* platforms is very important for our model. Consider two platforms situated at the same station. If the trains associated with the platforms are opposite going, then we say those two platforms are opposite to each other. In many railway networks the opposite platforms of the same railway station are supplied energy by same substation and are appropriate candidates for the occurrence of synchronization processes between suitable train pairs, because of the negligible transmission loss.

- **Track:** A directed arc between two distinct and non-opposite platforms is called a track. A track is used by a train to make a trip from one station to the next in a certain direction. We assume that the direction associated with a track is fixed. The first and second platforms associated with a track are called the origin platform and the destination platform respectively, *i.e.*, any train using that track will depart from the origin platform and arrive at the destination platform. Two tracks are called opposite to each other if the origin platform and the destination platform of one track are opposite to the destination platform and the origin platform respectively of the other track.
- **Train-line or line:** A train-line or line is a directed path with the set of nodes representing non-opposite platforms and the set of arcs representing non-opposite tracks. In the set of arcs, the origin platform of the any track (except the first one) is the same as the destination platform of the previous track and the destination platform of any track (except the last one) is the same as the origin platform of the previous track. The set of nodes are the platforms corresponding to those arcs. The origin platform of the first track and the destination platform of the final track belonging to a line are called the terminal platforms for that line.

Consider any train-line, for which there exists another train-line such that both of them has same number of platforms and arcs and each platform and track of the

first line are opposite to some unique platform and unique track of the other line respectively. Then the lines are called opposite to each other.

- **Crossing-over:** A crossing-over is a special type of directed arc that connects two train lines. After arriving at the terminal platform of a train-line, a train often turns around by traversing the crossing-over and starts traveling through another train line. Though physically the train is the same as before, from a management point of view the paths traversed by the train are different and often in opposite direction. So, the same physical train is treated functionally as two different trains by the railway management [36, page 41].
- **Train-path or path:** A train-path is a directed path, which represents all the platforms and tracks on a line covered by a train in chronological order. In all of the possible cases except two, the train-path of a train is just equal to the train-line. The only exception is when a train arrives at some platform from the depot or returns to the depot from a platform.

3.2 Notation to describe a railway network

Consider a railway network where the set of all stations is denoted by \mathcal{S} . The set of indices of the stations is denoted by $I_{\mathcal{S}} = \{1, 2, \dots, |\mathcal{S}|\}$. The i th station is denoted by $\mathcal{S}(i)$ where $i \in I_{\mathcal{S}}$. The set of all platforms in the railway network of our consideration is indicated by $\mathcal{N} = \{1, 2, \dots, |\mathcal{N}|\}$. The set of all tracks are represented by \mathcal{A} where,

$$\forall a \in \mathcal{A} \exists i \in \mathcal{N} \exists j \in \mathcal{N} \setminus \{i\} \quad (a = (i, j)).$$

The directed graph of the railway network is expressed by $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. For any two platforms $i, j \in \mathcal{N}$, we introduce a symmetric binary indicator $\pi_{ij} \in \{0, 1\}$ which is equal to one if they are opposite to each other and zero else. The indicator is symmetric, because for any two platforms $i \in \mathcal{N}$ and $j \in \mathcal{N} \setminus \{i\}$ we will always have $\pi_{ij} = \pi_{ji}$. The set of all distinct opposite platform pairs ordered lexicographically is denoted by $\mathcal{O} = \{(i, j) \in \mathcal{N} \times \mathcal{N} : (\pi_{ij} = 1) \wedge (i < j)\}$. The set of every distinct opposite platform pair powered by same substation is denoted by $\Omega = \{(i, j) \in \mathcal{O} : \text{there exists a substation which powers both } i \text{ and } j\} \subseteq \mathcal{O}$. For any two tracks $(i, j), (j', i') \in \mathcal{A}$, they are opposite to each other if $\pi_{ii'} = 1 \wedge \pi_{jj'} = 1$.

The set of indices of all train lines present in the railway network is denoted by

$\mathcal{L} = \{1, 2, \dots, |\mathcal{L}|\}$. As defined previously, the platforms belonging to the same line are non-opposite, *i.e.*,

$$\forall l \in \mathcal{L} \forall i \in \mathcal{N}_l \forall j \in \mathcal{N}_l \setminus \{i\} \quad (\pi_{ij} = 0)$$

Note that the non-oppositeness of the tracks is implied by the non-oppositeness of platforms on that line.

Consider two different lines $l, l' \in \mathcal{L}$. They are opposite two each other if:

$$|\mathcal{N}_l| = |\mathcal{N}_{l'}| \wedge (\forall i \in I_{\mathcal{N}_l} \quad (\pi_{\mathcal{N}_l(i)\mathcal{N}_{l'}(|\mathcal{N}_l|+1-i)} = 1))$$

The set of all trains to be considered in our problem is denoted by $\mathcal{T} = \{1, 2, \dots, |\mathcal{T}|\}$. The enumeration of the trains in the set \mathcal{T} is based on two properties, which we describe as follows:

- The path of a train $t \in \mathcal{T}$ is a directed path denoted by:

$$\mathcal{P}^t = (\mathcal{N}^t(1), \mathcal{A}^t(1), \mathcal{N}^t(2), \mathcal{A}^t(2), \dots, \mathcal{A}^t(|\mathcal{A}^t|), \mathcal{N}^t(|\mathcal{N}^t|))$$

where $\mathcal{N}^t \subseteq \mathcal{N}$ and $\mathcal{A}^t \subseteq \mathcal{A}$ are the set of all platforms and tracks visited by the train t in chronological order respectively. The path of any train is always limited along a unique train line, *i.e.*,

$$\forall t \in \mathcal{T} \exists ! l \in \mathcal{L} \quad (\mathcal{P}^t \subseteq \mathcal{P}_l)$$

In the existential quantifier of the equation above, the symbol $!$ stands for uniqueness of the line l . As discussed previously, in all the cases except those involving the depot, this set relation is equality, *i.e.*, the train-path is equal to the path of the associated train-line. If the same physical train crosses over to a new line, it will be labeled as another train in \mathcal{T} , even if it is physically the same.

- Different trains can have same path *only if* the time intervals in which they traverse the common path are different.

The decision variables to be determined in our problem are the arrival and departure times of trains to and from the associated platforms respectively. The decision variables in our optimization problem are also called *event times*. Let a_i^t be t and d_j^t be the departure time of train t from platform $j \in \mathcal{N}^t$.

3.3 Constraints in a railway network

3.3.1 Trip time constraint

The trip time constraints play the most important role in train energy consumption and regenerative energy production. These can be of two types as follows.

Trip time constraint associated with a track

Consider the trip of any train $t \in \mathcal{T}$ from platform i to platform j along the track $(i, j) \in \mathcal{A}^t$. The train t departs from platform i at time d_i^t , arrives at platform j at time a_j^t , and it can have a trip time between $\underline{\tau}_{ij}^t$ and $\overline{\tau}_{ij}^t$. The trip time constraint can be written as follows:

$$\forall t \in \mathcal{T} \quad \forall (i, j) \in \mathcal{A}^t \quad \underline{\tau}_{ij}^t \leq a_j^t - d_i^t \leq \overline{\tau}_{ij}^t. \quad (3.1)$$

Trip time constraint associated with a crossing-over

Recall that, a crossing-over is a special type of directed arc that connects two *train-lines*. After arriving at the terminal platform of a train-line when a train turns around by traversing the crossing-over and starts traveling through another train-line, then the same physical train is treated and labelled functionally as two different trains by the railway management (Section 3.1). Let φ be , where turn-around events occur. Consider any crossing-over $(i, j) \in \varphi$, where the platforms i and j are situated on different train-lines. Let \mathcal{B}_{ij} be the set of all train pairs involved in corresponding turn-around events on the crossing-over (i, j) . Let $(t, t') \in \mathcal{B}_{ij}$. Train $t \in \mathcal{T}$ turns around at platform i by travelling through the crossing-over (i, j) , and beginning from platform j starts traversing a different train-line as train $t' \in \mathcal{T} \setminus \{t\}$. A time window $[\underline{\kappa}_{ij}^{tt'}, \overline{\kappa}_{ij}^{tt'}]$ has to be maintained between the departure of the train from platform i (labelled as train t) and arrival at platform j (labelled as train t'). We can write this constraint as follows:

$$\forall (i, j) \in \varphi \quad \forall (t, t') \in \mathcal{B}_{ij} \quad \underline{\kappa}_{ij}^{tt'} \leq a_j^{t'} - d_i^t \leq \overline{\kappa}_{ij}^{tt'}. \quad (3.2)$$

To clearly illustrate the constraint we consider Figure 3.1. Here we have two train lines: line 1 and line 2. The terminal platform on line 1 is platform i and the first platform on line 2 is platform j . The crossing-over from line 1 to line 2 is the arc (i, j) . The train shown in the figure is labelled as t on platform i and labelled as train t' on platform j .

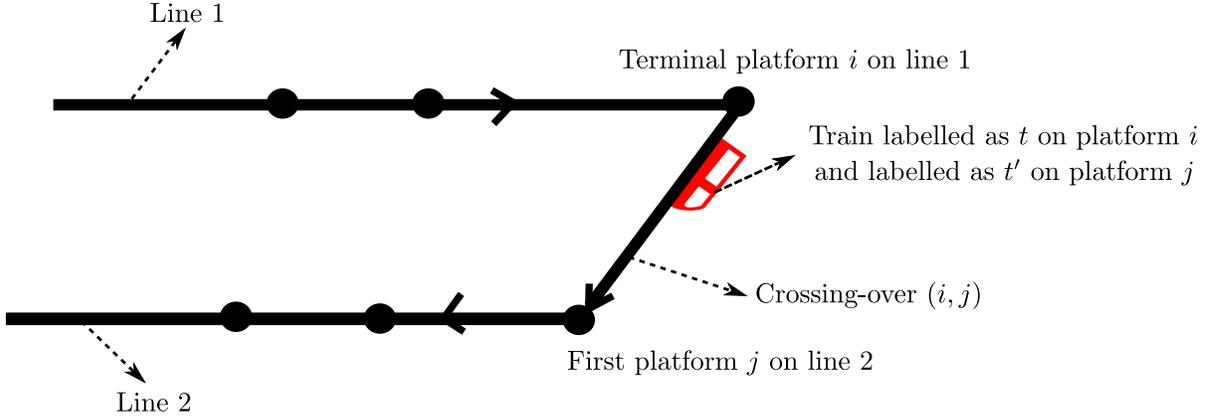


Figure 3.1: Trip time constraint associated with a crossing-over

3.3.2 Dwell time constraint

When any train $t \in \mathcal{T}$ arrives at a platform $i \in \mathcal{N}^t$, it dwells there for a certain time interval denoted by $[\underline{\delta}_i^t, \bar{\delta}_i^t]$ so that the passengers can get off and get on the train prior to its departure from platform j . The dwell time constraint can be written as follows:

$$\forall t \in \mathcal{T} \quad \forall i \in \mathcal{N}^t \quad \underline{\delta}_i^t \leq d_i^t - a_i^t \leq \bar{\delta}_i^t. \quad (3.3)$$

Every train $t \in \mathcal{T}$ arrives at the first platform $\mathcal{N}^t(1)$ in its train-path either from the depot or by turning around from some other line, and departs from the final platform $\mathcal{N}^t(|\mathcal{N}^t|)$ in order to either return to the depot or start as a new train on another line by turning around. So, the train t dwells at all platforms in \mathcal{N}^t . This is the reason why in Equation (3.3) the platform index i is varied over all elements of the set \mathcal{N}^t .

3.3.3 Connection constraint

In many cases, a single train connection might not exist between the origin and the desired destination of a passenger. To circumvent this, connecting trains are often used at interchange stations. Let $\chi \subseteq \mathcal{N} \times \mathcal{N}$ be the set of all platform pairs situated at the same interchange stations, where passengers transfer between trains. Let \mathcal{C}_{ij} be the set of connecting train pairs for a platform pair $(i, j) \in \chi$. For a train pair $(t, t') \in \mathcal{C}_{ij}$, train t is arriving at platform i and train $t' \in \mathcal{T}$ is departing from platform j . A connection time window denoted by $[\underline{\chi}_{ij}^{tt'}, \bar{\chi}_{ij}^{tt'}]$ is maintained between arrival of t and subsequent departure of t' , so that passengers can get off from the first train and get on the latter.

Let $(i, j) \in \chi$. Then the connection constraint can be written as:

$$\forall (i, j) \in \chi \quad \forall (t, t') \in \mathcal{C}_{ij} \quad \underline{\chi}_{ij}^{tt'} \leq d_j^{t'} - a_i^t \leq \overline{\chi}_{ij}^{tt'}. \quad (3.4)$$

3.3.4 Headway constraint

In any railway network, a minimum amount of time between the departures and arrivals of consecutive trains on the same track is maintained. This time is called headway time. For maintaining the quality of passenger service, many urban railway system keeps an upper bound between the arrivals and departures of successive trains on the same track, so that passengers do not have to wait too long before the next train comes. Let $(i, j) \in \mathcal{A}$ be the track between two platforms i and j , and \mathcal{H}_{ij} be the set of train-pairs who move along that track successively in order of their departures. Consider $(t, t') \in \mathcal{H}_{ij}$, and let $[\underline{h}_i^{tt'}, \overline{h}_i^{tt'}]$ and $[\underline{h}_j^{tt'}, \overline{h}_j^{tt'}]$ be the time windows that have to be maintained between the departures and arrivals of the trains t and t' from and to the platforms i and j respectively. So, the headway constraint can be written as:

$$\forall (i, j) \in \mathcal{A} \quad \forall (t, t') \in \mathcal{H}_{ij} \quad \underline{h}_i^{tt'} \leq d_i^{t'} - d_i^t \leq \overline{h}_i^{tt'} \quad \wedge \quad \underline{h}_j^{tt'} \leq a_j^{t'} - a_j^t \leq \overline{h}_j^{tt'}. \quad (3.5)$$

Similarly, headway times have to be maintained between two consecutive trains going through a crossing over. Consider any crossing over $(i, j) \in \varphi$ and two such trains, which leave the terminal platform of a train-line i labelled as t_1 and t_2 , traverse the crossing over (i, j) , and arrive at platform j of some other train-line labelled as t'_1 and t'_2 . The set of all such train quartets $((t_1, t'_1), (t_2, t'_2))$ is represented by $\tilde{\mathcal{H}}_{ij}$. Let $[\underline{h}_i^{t_1 t_2}, \overline{h}_i^{t_1 t_2}]$ be the headway time window between the departures of trains t_1 and t_2 from platform i and $[\underline{h}_j^{t'_1 t'_2}, \overline{h}_j^{t'_1 t'_2}]$ be the headway time window between the arrivals of the trains t'_1 and t'_2 to the platforms j . The associated headway constraints can be written as:

$$\forall (i, j) \in \varphi \quad \forall ((t_1, t'_1), (t_2, t'_2)) \in \tilde{\mathcal{H}}_{ij} \quad \underline{h}_i^{t_1 t_2} \leq d_i^{t_2} - d_i^{t_1} \leq \overline{h}_i^{t_1 t_2} \quad \wedge \quad \underline{h}_j^{t'_1 t'_2} \leq a_j^{t'_2} - a_j^{t'_1} \leq \overline{h}_j^{t'_1 t'_2}. \quad (3.6)$$

3.3.5 Total travel time constraint

Recall that, the train-path of a train is the directed path containing all platforms and tracks visited by it in chronological order. To maintain the quality of service in the railway network, for every train $t \in \mathcal{T}$, the total travel time to traverse its train-path \mathcal{P}^t

has to stay within a time window $[\underline{\tau}_{\mathcal{P}}^t, \bar{\tau}_{\mathcal{P}}^t]$. We can write this constraint as follows:

$$\forall t \in \mathcal{T} \quad \underline{\tau}_{\mathcal{P}}^t \leq a_{\mathcal{N}^t(|\mathcal{N}^t|)}^t - d_{\mathcal{N}^t(1)}^t \leq \bar{\tau}_{\mathcal{P}}^t, \quad (3.7)$$

where $\mathcal{N}^t(1)$ and $\mathcal{N}^t(|\mathcal{N}^t|)$ are the first and last platform in the train-path of t .

3.3.6 Domain of the event times

Without any loss of generality, we set the time of the first event of the railway service period, which corresponds to the departure of the first train of the day from some platform, to start at zero second. By setting all trip times and dwell times to their maximum possible values we can obtain an upper bound for the final event of the railway service period, which is the arrival of the last train of the day at some platform, denoted by $m \in \mathbf{Z}_{++}$. So the domain of the decision variables can be expressed by the following equation:

$$\forall t \in \mathcal{T} \quad \forall i \in \mathcal{N}^t \quad 0 \leq a_i^t \leq m, 0 \leq d_i^t \leq m. \quad (3.8)$$

In vector notation the decision variables are denoted by $a = ((a_i^t)_{i \in \mathcal{N}^t})_{t \in \mathcal{T}}$ and $d = ((d_i^t)_{i \in \mathcal{N}^t})_{t \in \mathcal{T}}$.

3.4 Relation between passenger demand, headway and number of trains

Now we discuss the relation between passenger demand, headway and number of trains. We denote the passenger demand by D , train capacity by c and utilization rate by u . If we denote the number of trains in service per hour by n , then we have $D = c \times u \times n$ [27]. Because the headway time h satisfies the relation $h = \frac{3600}{n}$, we have

$$h = \frac{3600 \times c \times u}{D}. \quad (3.9)$$

It should be noted that the train capacity c and the utilization rate u are constant parameters. However the passenger demand varies with time. As a result, in the equation above trains will have different headway at different periods.

Chapter 4

A robust mixed integer optimization model

In this chapter we present a robust mixed integer optimization model to utilize the regenerative braking energy produced by trains in a railway network. The optimization model calculates a railway timetable that saves regenerative energy of braking trains by transferring it to suitable accelerating trains in need of energy. This chapter is organized as follows. In Section 4.1 we describe the relevant event times of a train around a platform. Then, in the next section we characterize the train pairs and the associated platform pairs necessary to describe the synchronization processes between suitable train pairs. Section 4.3 describes the structure of the optimization model. Sections 4.4 and 4.5 model the objective function of the optimization problem using hypograph approach and interval algebra. Then in Section 4.6, we collect the objective and all the constraints for the optimization problem and present the full optimization problem. Section 4.7 describes the limitations of the optimization model.

4.1 Relevant event times of a train around a platform

Recall from our discussion in Section 1.1 that the energy consumption and regeneration of a train is associated with its acceleration phase and braking phase during its trip from an origin platform to a destination platform. However, when we are calculating a timetable, our decision variables are the event times, *i.e.*, the departure and arrival times of the train (See Section 3.2). So, first we need to relate these event times to the braking and acceleration phases in a rigorous manner.

Consider any platform pair $(i, j) \in \Omega$ and consider any train t . In chronological order the relevant event times of train t around platform i are as follows:

1. Start of the braking phase of the train in order to stop at platform i denoted by a_i^{t-} .
2. The arrival of the train at platform i denoted by $a_i^t = a_i^{t-} + \beta_i^t$, where β_i^t is the duration of the braking phase. The arrival concludes the braking phase of the train around platform i and begins the dwelling process at platform i .
3. The departure of the train from platform j denoted by d_i^t , which concludes the dwelling process of train i at platform i and begins the acceleration phase of the train around platform i .
4. The acceleration process ends at $d_i^{t+} = d_i^t + \alpha_i^t$, where α_i^t is the duration of the acceleration phase.

The mentioned events above are expressed as rectangular temporal blocks in the top part of the Figure 4.1, where the bottom part shows the corresponding speed-profile. In the figure, the rectangles represent the braking phase, dwelling process and acceleration phase from left to right. The left side of a rectangle denotes the beginning of the associated process and the right side represents the end. To model the objective function, we work with these temporal blocks in this chapter.

4.2 Synchronization process between suitable train pairs (SPSTP)

At first, we need to characterize the train pairs and the associated platform pairs necessary to describe the SPSTPs. The transmission loss in transferring electrical energy between two trains is proportional to the distance between them [42, Chapter 3]. If we consider two trains who are opposite to each other on the same station, the distance between them is relatively small. As a result, the transmission loss in transferring electrical energy between them would be negligible in comparison with trains situated at different stations. So, the platform pairs to consider are those opposite to each other and powered by the same electrical substations. Recall that, the set that contains all such platform pairs is denoted by Ω . Consider any such platform pair $(i, j) \in \Omega$, and let $\mathcal{T}_i \subseteq \mathcal{T}$ be the set of all the trains which arrive at, dwell and then depart from platform i . Suppose, $t \in \mathcal{T}_i$. Now, we are interested to find another train \tilde{t} on platform j , *i.e.*, $\tilde{t} \in \mathcal{T}_j$, which along with t would form a suitable pair for the transfer of regenerative braking energy. To achieve this, we start with an initial feasible timetable for the railway, which represents

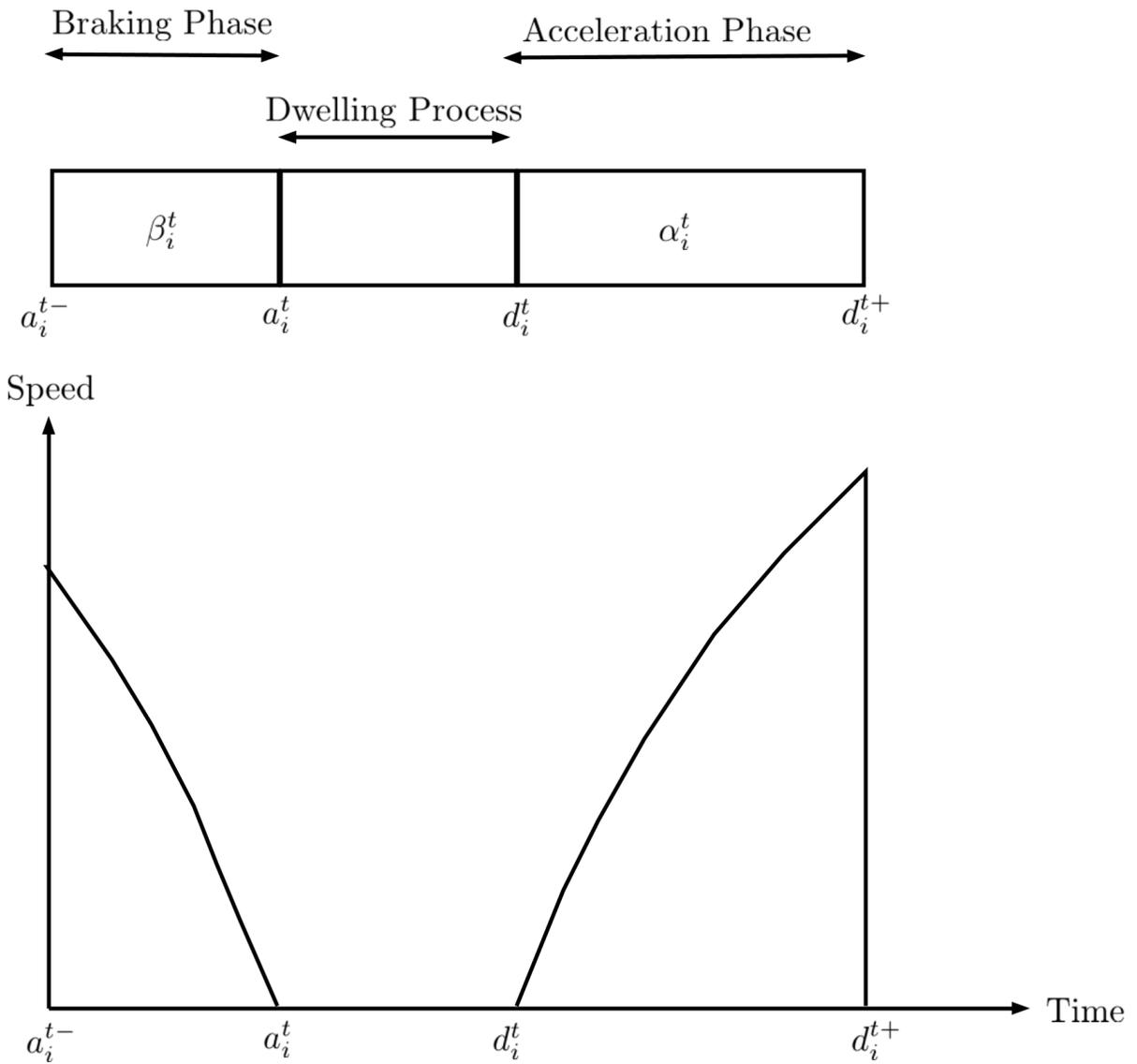


Figure 4.1: Rectangular blocks to indicate temporal events of train t around platform i (top) and the associated speed profile (bottom)

the desired service to be delivered. Too much deviation from it is not desired. For every train t , this timetable provides a feasible arrival time \bar{a}_i^t and a feasible departure time \bar{d}_i^t to and from every platform $i \in \mathcal{N}^t$ respectively. Intuitively, among all the trains that go through platform j , the one which is temporally closest to t in the initial timetable would be the best candidate to form a pair with t . The alternative choices are not feasible with the current technology as discussed below.

Two trains with same direction on the same station. For transfer of regenerative energy, one of the trains has to be in its braking phase and the other one in accelerating phase. Because we always need to maintain the safety (headway) distance between two trains on the same track, when a train goes into the braking phase, the platform it enters cannot be occupied by another train for safety reason. So picking two trains (one accelerating and one braking) at the same platform for synchronization is not feasible from a safety point of view.

Trains on different stations. Suppose we pick two trains for possible transfer which are at least one station apart. The first problem with this is just the in-feasibility of the strategy with the current technology. Currently the regenerative energy is transferred between opposite platforms by installing super-capacitors on the overhead line [10]. To do this reliably over two stations would require either 1) the entire overhead line to act as a super-capacitor (to prevent the fluctuation of the line voltage [5]) which is not feasible, or 2) transfer the energy to the power grid and then from power grid to the other station, which requires specialized technology such as reversible electrical substations [15, page 30]. Another problem is that a large portion of the regenerative energy will be lost due to transmission loss.

The temporal proximity can be of two types with respect to t , which results in the following definitions.

Definition 4.1. Consider any $(i, j) \in \Omega$. For every train $t \in \mathcal{T}_i$, the train $\vec{t} \in \mathcal{T}_j$ is called the **temporally closest train to the right of t** if

$$\vec{t} = \underset{t' \in \{x \in \mathcal{T}_j : 0 \leq \frac{\bar{a}_j^x + \bar{d}_j^x}{2} - \frac{\bar{a}_i^t + \bar{d}_i^t}{2} \leq r\}}{\operatorname{argmin}} \left\{ \left| \frac{\bar{a}_i^t + \bar{d}_i^t}{2} - \frac{\bar{a}_j^{t'} + \bar{d}_j^{t'}}{2} \right| \right\}, \quad (4.1)$$

where r is an empirical parameter determined by the timetable designer and is much smaller than the time horizon of the entire timetable.

Definition 4.2. Consider any $(i, j) \in \Omega$. For every train $t \in \mathcal{T}_i$, the train $\overleftarrow{t} \in \mathcal{T}_j$ is

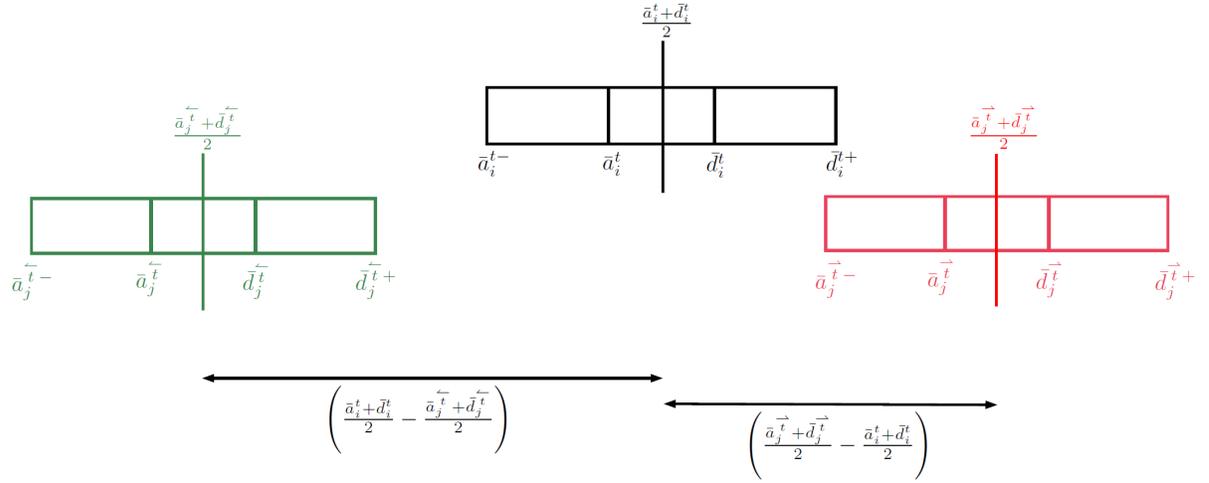


Figure 4.2: For platform pair $(i, j) \in \Omega$ and train $t \in \mathcal{T}_i$ the figure shows temporally closest train to t 's left \overleftarrow{t} (corresponding to the green temporal blocks) and temporally closest train to t 's right \overrightarrow{t} (corresponding to the red temporal blocks). Here $\tilde{t} = \overrightarrow{t}$.

called the **temporally closest train to the left of t** if

$$\overleftarrow{t} = \underset{t' \in \{x \in \mathcal{T}_j : 0 < \frac{\overline{a}_i^t + \overline{d}_i^t}{2} - \frac{\overline{a}_j^x + \overline{d}_j^x}{2} \leq r\}}{\operatorname{argmin}} \left\{ \left| \frac{\overline{a}_i^t + \overline{d}_i^t}{2} - \frac{\overline{a}_j^{t'} + \overline{d}_j^{t'}}{2} \right| \right\}. \quad (4.2)$$

Definition 4.3. Consider any $(i, j) \in \Omega$. For every train $t \in \mathcal{T}_i$, the train $\tilde{t} \in \mathcal{T}_j$ is called the **temporally closest train to t** if

$$\tilde{t} = \underset{t' \in \{\overrightarrow{t}, \overleftarrow{t}\}}{\operatorname{argmin}} \left\{ \left| \frac{\overline{a}_i^t + \overline{d}_i^t}{2} - \frac{\overline{a}_j^{t'} + \overline{d}_j^{t'}}{2} \right| \right\}. \quad (4.3)$$

If both \overrightarrow{t} and \overleftarrow{t} are temporally equidistant from t , we pick one of them arbitrarily.

Figure 4.2 illustrates the concepts discussed above, where for $(i, j) \in \Omega$ and $t \in \mathcal{T}_i$ we have shown the temporally closest train to its left and right. In the figure, $\overrightarrow{t} = \tilde{t}$. If both \overrightarrow{t} and \overleftarrow{t} are temporally equidistant from t in the sense of our proposed definitions, we pick one of them arbitrarily.

Among all the trains $t' \in \mathcal{T}_j$, train \tilde{t} is the best candidate to make a transfer of regenerative energy with t . This transfer can be of two types as follows which is going to determine our optimization strategy:

- If $\tilde{t} = \overrightarrow{t}$, then it would be convenient to synchronize the acceleration phase of t

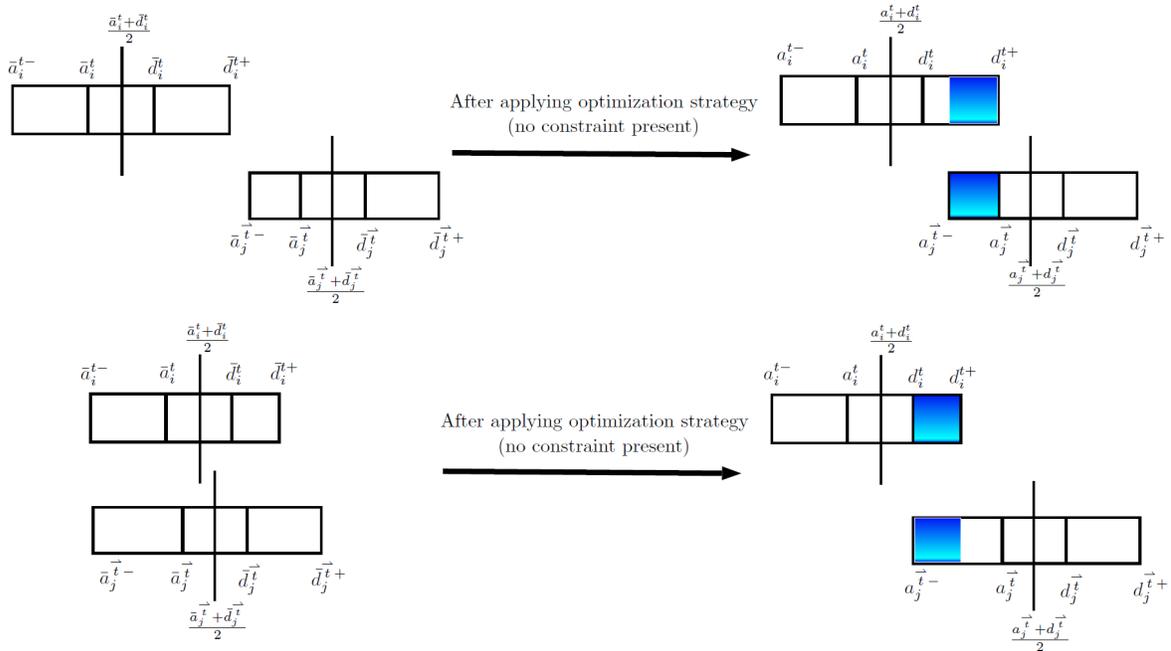


Figure 4.3: Applying the optimization strategy for the case $\tilde{t} = \vec{t}$, when there is no constraint. The left hand side and right hand side of the arrows correspond to the initial timetable and the final timetable respectively. The shaded regions represent the overlapping time between trains.

with the braking phase of \vec{t} . Our optimization strategy in this case would be to maximize the overlapping time between the two mentioned phases subject to the constraints in the railway network.

Figure 4.3 shows how our optimization strategy will maximize the overlapping time of a synchronization processes between a suitable train pair in this case, when there is no constraint. In the presence of other constraints, the maximized overlapping time will be less than or equal to the one in the figure; it might even be zero if doing otherwise violates some constraint.

- If $\tilde{t} = \overleftarrow{t}$, then it would be convenient to synchronize the acceleration phase of \overleftarrow{t} with the braking phase of t . Our optimization strategy in this case would be to maximize the overlapping time between the two mentioned phases.

Figure 4.4 shows how our optimization strategy will maximize the overlapping time of a synchronization process between a suitable train pair in this case, when there is no constraint. Just like the previous case, in the presence of other constraints, the maximized overlapping time will be less than or equal to the one in the figure; it might even be zero if doing otherwise violates some constraint.

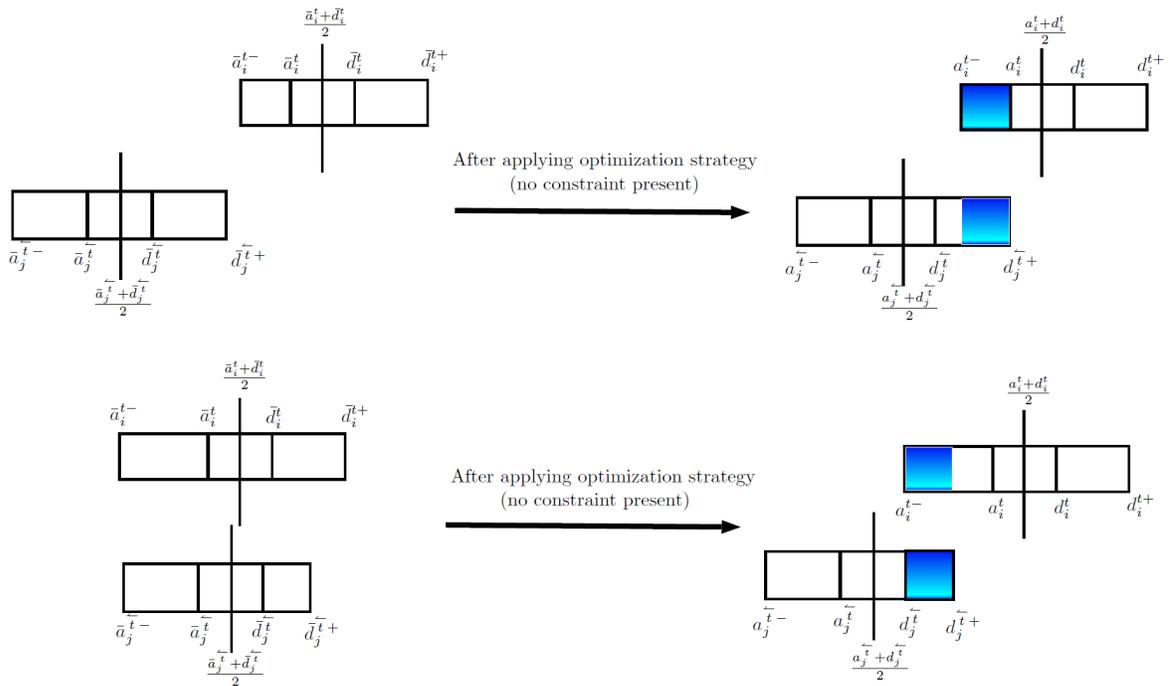


Figure 4.4: Applying the optimization strategy for the case $\tilde{t} = \bar{t}$, when there is no constraint. The left hand side and right hand side of the arrows correspond to the initial timetable and the final timetable respectively. The shaded regions represent the overlapping time between trains.

We do not model our objective to simultaneously maximize the intersecting time between acceleration phase of \overleftarrow{t} with the braking phase of t and the acceleration phase of t with the braking phase of \overleftarrow{t} , because this strategy will not be very effective in practice. The reasons behind that are as follows:

1. If such a strategy is followed and achieved, then the temporal events corresponding \overleftarrow{t} and \overleftarrow{t} will be very close to each other, which might lead to constraint violation. Even if no constraint is violated, keeping two trains \overleftarrow{t} and \overleftarrow{t} so close is not desired from a practical point of view, as any sort of disturbance might result in some constraint violation. This is shown in Figure 4.5.

1. Consider any track $(k, l) \in \mathcal{A}$ and any train pair (t, t') on the same track. Let (k', l') be the opposite track of (k, l) . It may happen that, $\overleftarrow{t} = \overleftarrow{t'}$, *i.e.*, what is the temporally closest train to the right of train t is the temporally closest train to the left for train t' . Now consider the case when $\tilde{t} = \overleftarrow{t}$ and $\tilde{t}' \neq t'$, then the strategy (the strategy that we have discarded, not the original one) will explicitly try to shift the events associated with $\overleftarrow{t} = \overleftarrow{t'}$ to opposite directions, which is not desired as compared to original optimization strategy in this case twice the computational effort will be spent to find a cost increasing direction, yet the post-optimization scenario will be the same. Figure 4.6 illustrates this case.

4.3 Structure of the optimization model

Any SPSTP can be described by specifying the corresponding i, j, t and \tilde{t} by using the definitions above. We construct a set of all the SPSTPs, which we denote by \mathcal{E} . Each element of this set is a tuple of the form (i, j, t, \tilde{t}) . Because \tilde{t} is unique for any t in each element of \mathcal{E} , we can partition \mathcal{E} into two sets denoted by $\overrightarrow{\mathcal{E}}$ and $\overleftarrow{\mathcal{E}}$, which contain elements of the form $(i, j, t, \overrightarrow{t})$ and $(i, j, t, \overleftarrow{t})$ respectively. For every $(i, j, t, \overrightarrow{t}) \in \overrightarrow{\mathcal{E}}$ (called *right event*), our strategy is to synchronize the accelerating phase of t with the braking phase of \overrightarrow{t} . On the other hand, for every $(i, j, t, \overleftarrow{t}) \in \overleftarrow{\mathcal{E}}$ (called *left event*), it is convenient to synchronize the accelerating phase of t with the braking phase of t . For every $(i, j, t, \overrightarrow{t}) \in \overrightarrow{\mathcal{E}}$, the corresponding overlapping time is denoted by $\sigma_{ij}^{\overrightarrow{t}t}$, (called *right event overlapping time*) and for every $(i, j, t, \overleftarrow{t}) \in \overleftarrow{\mathcal{E}}$, the corresponding overlapping time is denoted by $\sigma_{ij}^{\overleftarrow{t}t}$ (called *left event overlapping time*). Our objective is to maximize the sum of the right event and left event overlapping times over all the elements of $\overrightarrow{\mathcal{E}}$ and $\overleftarrow{\mathcal{E}}$,

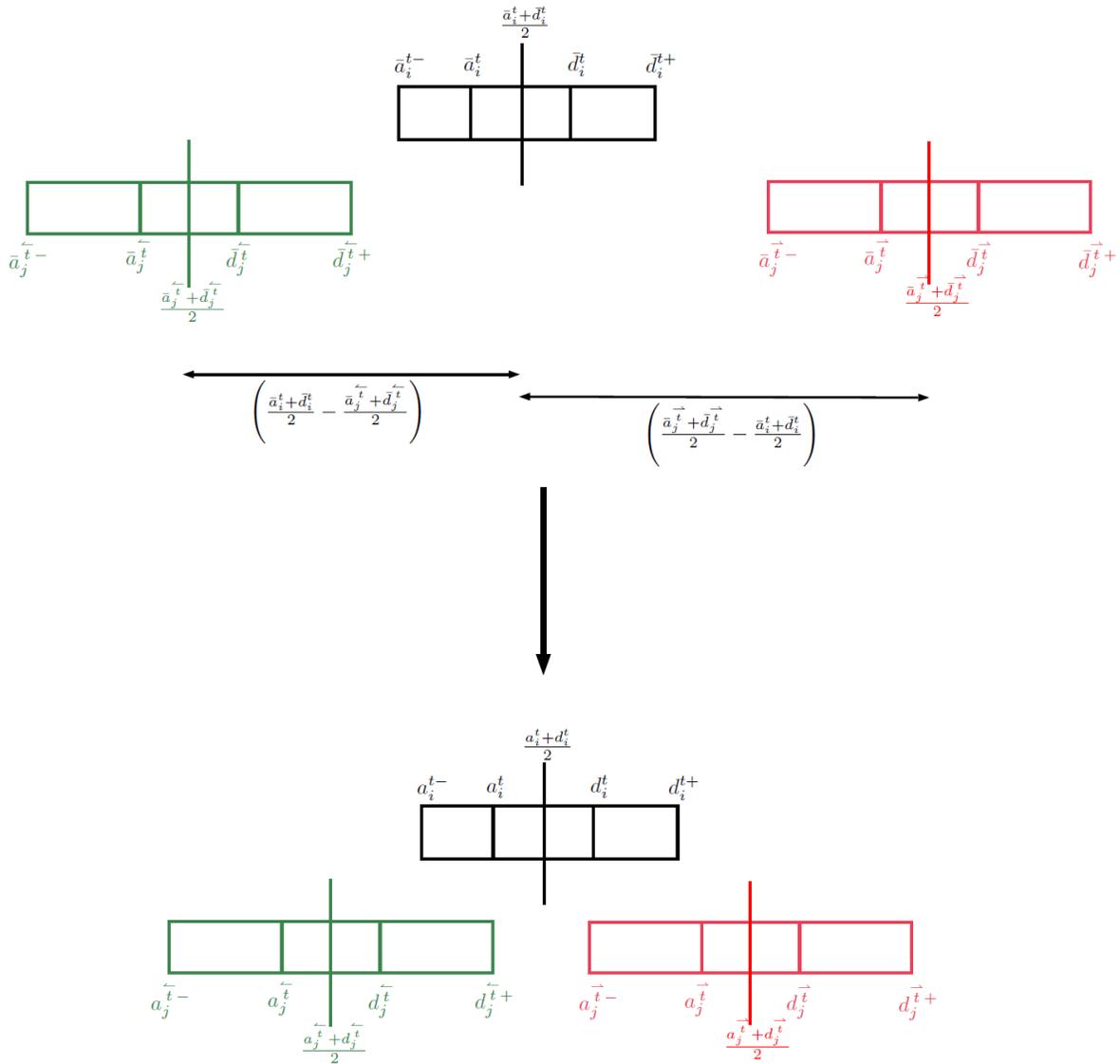


Figure 4.5: If we simultaneously maximize the overlapping time between acceleration phase of \bar{t} with the braking phase of \bar{t} and the acceleration phase of \underline{t} with the braking phase of \underline{t} , after applying such a strategy \bar{t} and \underline{t} might be very close to each other which is not desired in practical situations. The top and bottom part of the figure show the pre-optimization and the post-optimization scenario respectively for such a case.

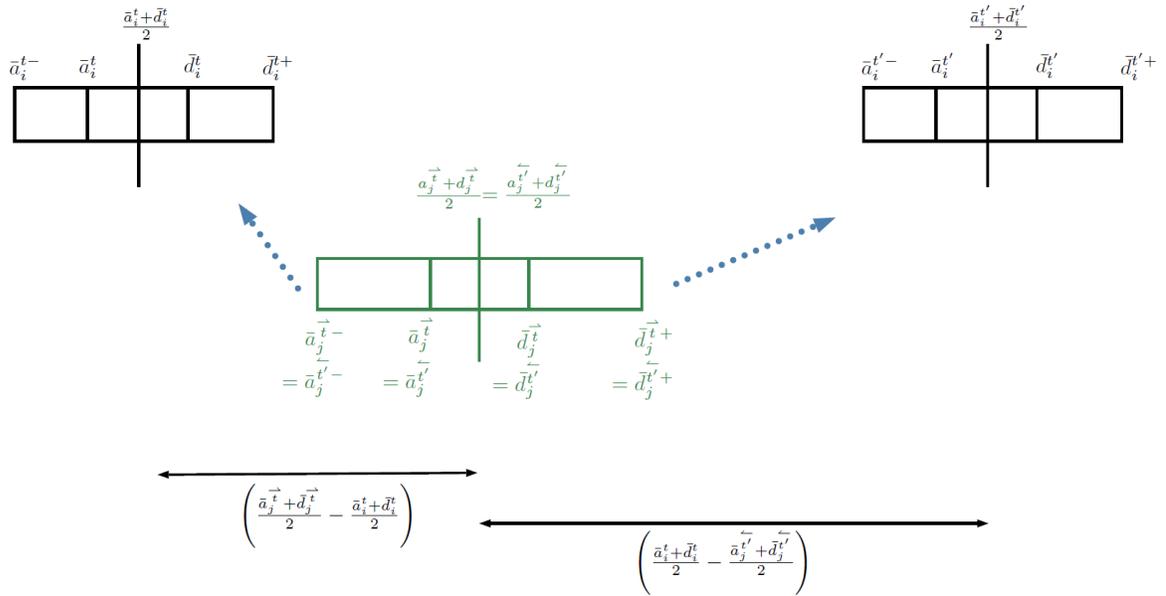


Figure 4.6: If we simultaneously maximize the overlapping time between acceleration phase of t with the braking phase of \bar{t} and the acceleration phase of $\bar{t} = \bar{t}'$ with the braking phase of t' , after applying such a strategy, computational effort would be spent in trying shift the events associated with $\bar{t} = \bar{t}'$ to opposite directions, where only the first one is achievable.

i.e., the objective function is

$$\sum_{(i,j,t,\bar{t}) \in \bar{\mathcal{E}}} \sigma_{ij}^{\bar{t}t} + \sum_{(i,j,t,\bar{t}) \in \bar{\mathcal{E}}} \sigma_{ij}^{\bar{t}t}, \quad (4.4)$$

and the optimization problem is

$$\text{maximize} \quad \sum_{(i,j,t,\bar{t}) \in \bar{\mathcal{E}}} \sigma_{ij}^{\bar{t}t} + \sum_{(i,j,t,\bar{t}) \in \bar{\mathcal{E}}} \sigma_{ij}^{\bar{t}t}$$

subject to

Equations (3.1), (3.3), (3.4), (3.2), (3.5) and (3.7)

$$\forall t \in \mathcal{T} \quad \forall i \in \mathcal{N}^t \quad (a_i^t \geq 0, d_i^t \geq 0)$$

$$\forall (i,j,t,\bar{t}) \in \mathcal{E} \quad (\sigma_{ij}^{\bar{t}t} \geq 0),$$

We model $\sigma_{ij}^{\bar{t}t}$ for all $(i,j,t,\bar{t}) \in \bar{\mathcal{E}}$ and $\sigma_{ij}^{\bar{t}t}$ for all $(i,j,t,\bar{t}) \in \bar{\mathcal{E}}$ in terms of the arrival and departure times of trains. Consider the case, when $(i,j,t,\bar{t}) \in \bar{\mathcal{E}}$. We need to ensure that after we apply the optimization strategy, \bar{t} still stays the temporally closest train to the right of t . Otherwise, the only way to achieve a positive overlapping time is to synchronize the braking phase of t with the accelerating phase of \bar{t} , which might result in a large deviation of event times compared to the original timetable, especially when there is no or very little overlapping to begin with. We write this constraint as follows:

$$\frac{(a_j^{\bar{t}} + d_j^{\bar{t}} - a_i^t - d_i^t)}{(\bar{a}_j^{\bar{t}} + \bar{d}_j^{\bar{t}} - \bar{a}_i^t - \bar{d}_i^t + \epsilon)} \geq 0. \quad (4.5)$$

Here ϵ is a very small positive number to prevent division by zero. A graphical illustration of this constraint is shown in Figure 4.7.

4.4 Modeling overlapping time for right events

Let us denote the start of the braking phase of train t before arriving at platform i by a_i^{t-} and the end of its accelerating phase after departing from the same platform by d_i^{t+} . For all $t \in \mathcal{T}$ and for all $i \in \mathcal{N}^t$, the duration of the associated braking phase is $\beta_i^t = a_i^t - a_i^{t-}$ and the duration of the associated accelerating phase is $\alpha_i^t = d_i^{t+} - d_i^t$. Depending on the trip time of the associated trip in consideration, both the durations are within some time bounds, *i.e.*, $\alpha_i^t \in [\underline{\alpha}_i^t, \bar{\alpha}_i^t]$ and $\beta_i^t \in [\underline{\beta}_i^t, \bar{\beta}_i^t]$. Though we do not know the optimal

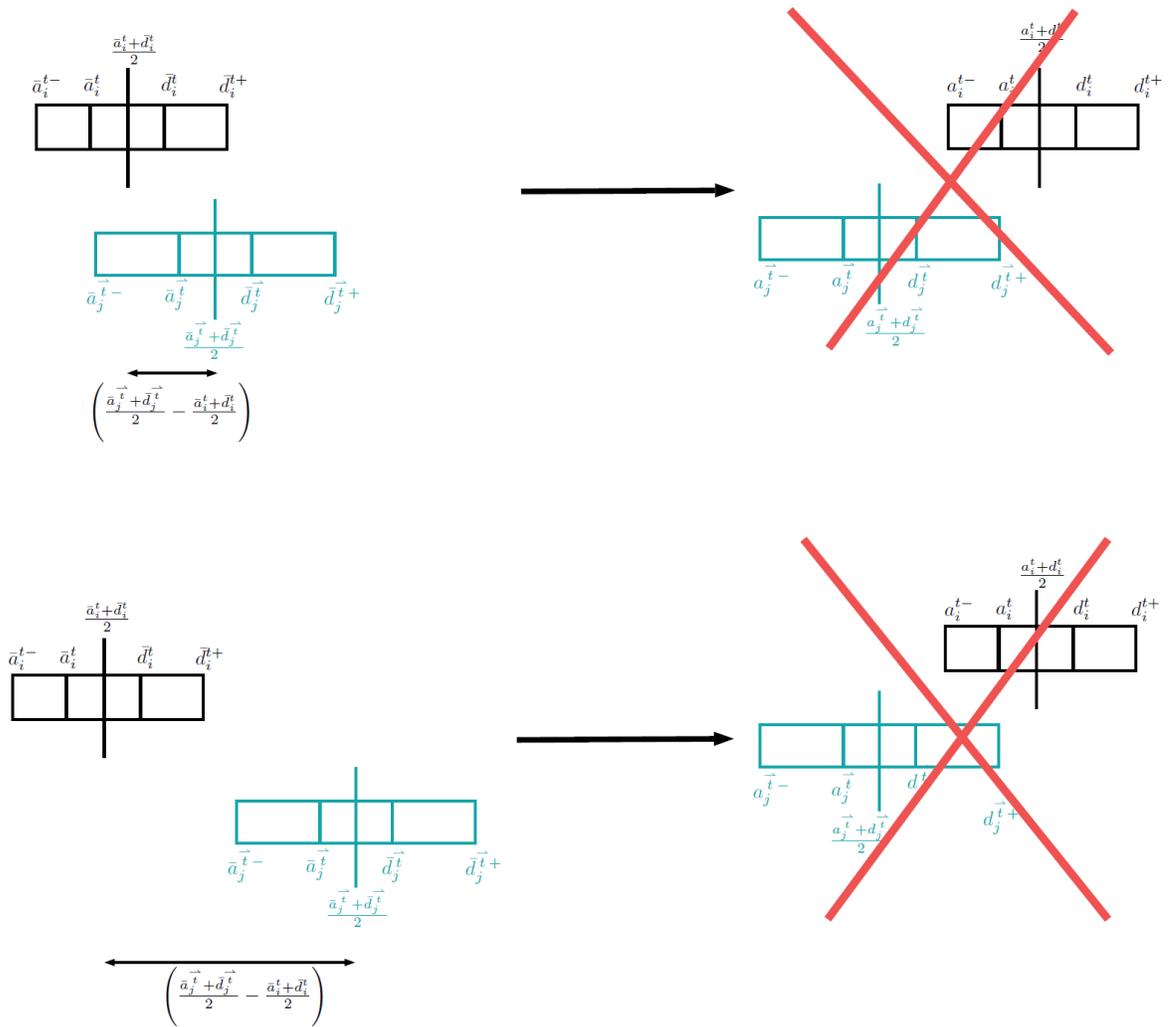


Figure 4.7: Graphical illustration of the constraint described by Equation (4.5), which prevents the occurrence of the cross-marked situations

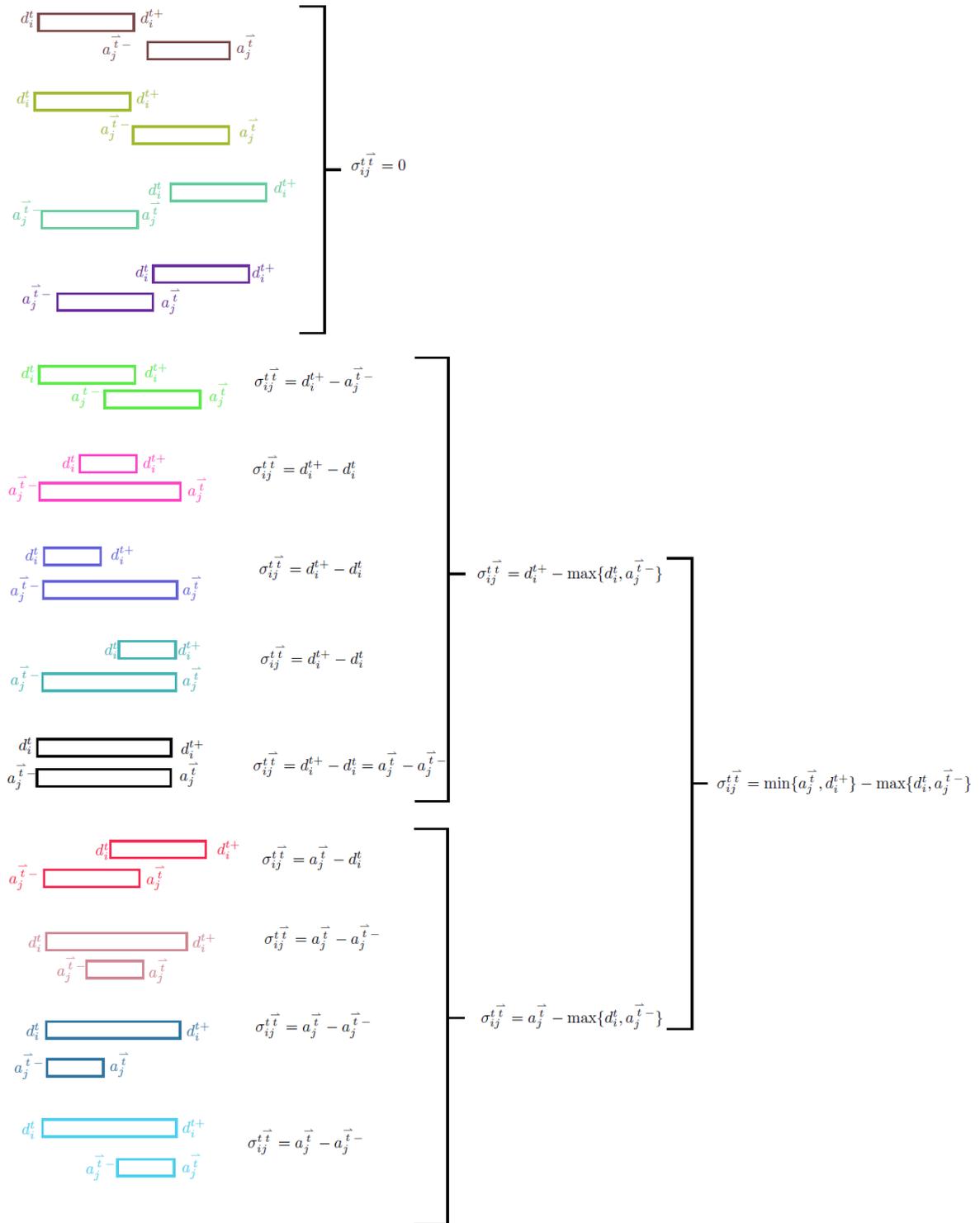


Figure 4.8: All possible overlapping times between the accelerating phase of train t and braking phase of its temporally closest train

trip time of the trains in advance, these lower and upper bounds can be calculated by existing software [43, page 3]. For the same reason, the start of the braking phase and end of the accelerating phase are within time bounds described by $a_i^{t-} \in [a_i^t - \bar{\beta}_i^t, a_i^t - \underline{\beta}_i^t]$ and $d_i^{t+} \in [d_i^t + \underline{\alpha}_i^t, d_i^t + \bar{\alpha}_i^t]$. All the time bounds are on the order of seconds, as the trip time variation are on the order of seconds, so it is reasonable to pursue a robust formulation. To model the right event overlapping time $\sigma_{ij}^{\vec{t}\vec{t}}$ for all $(i, j, t, \vec{t}) \in \vec{\mathcal{E}}$, we propose the following lemma.

Lemma 4.4. *For all $(i, j, t, \vec{t}) \in \vec{\mathcal{E}}$, the right event overlapping time $\sigma_{ij}^{\vec{t}\vec{t}}$ in Equation 4.4 satisfies the following constraints*

$$a_j^{\vec{t}} - d_i^t + \epsilon \leq \underline{\alpha}_i^t + \underline{\beta}_j^{\vec{t}} + M(1 - \lambda_{ij}^{\vec{t}\vec{t}}), \quad (4.6)$$

$$d_i^t - a_j^{\vec{t}} + \epsilon \leq M(1 - \lambda_{ij}^{\vec{t}\vec{t}}), \quad (4.7)$$

$$\sigma_{ij}^{\vec{t}\vec{t}} \geq 0, \quad (4.8)$$

$$\sigma_{ij}^{\vec{t}\vec{t}} \leq \underline{\alpha}_i^t \lambda_{ij}^{\vec{t}\vec{t}}, \quad (4.9)$$

$$\sigma_{ij}^{\vec{t}\vec{t}} \leq \underline{\beta}_j^{\vec{t}} \lambda_{ij}^{\vec{t}\vec{t}}, \quad (4.10)$$

$$\sigma_{ij}^{\vec{t}\vec{t}} \leq d_i^t - a_j^{\vec{t}} + \underline{\alpha}_i^t + \underline{\beta}_j^{\vec{t}} + M(1 - \lambda_{ij}^{\vec{t}\vec{t}}), \quad (4.11)$$

$$\sigma_{ij}^{\vec{t}\vec{t}} \leq a_j^{\vec{t}} - d_i^t + M(1 - \lambda_{ij}^{\vec{t}\vec{t}}). \quad (4.12)$$

where M is a large positive number, ϵ is a small positive number smaller than time granularity considered and $\lambda_{ij}^{\vec{t}\vec{t}}$ is a binary variable which is one if and only if $\sigma_{ij}^{\vec{t}\vec{t}}$ is positive.

Proof. We provide a robust formulation to prove the lemma. We use the hypograph approach to model the overlapping time $\sigma_{ij}^{\vec{t}\vec{t}}$ in terms of the associated event times [8, page 75, 134]. Consider any $a_i^{t-} \in [a_i^t - \bar{\beta}_i^t, a_i^t - \underline{\beta}_i^t]$ and any $d_i^{t+} \in [d_i^t + \underline{\alpha}_i^t, d_i^t + \bar{\alpha}_i^t]$ for some trip time. From Interval algebra [3], we know that there can be thirteen different kinds of overlapping possible between the accelerating phase of train t and the braking phase of train \vec{t} as shown in Figure 4.8. However, there is structure among these thirteen relationships when we are concerned with the associated overlapping time $\sigma_{ij}^{\vec{t}\vec{t}}$. Among them, when $a_j^{\vec{t}-} \geq d_i^{t+}$ or $a_j^{\vec{t}} \leq d_i^t$ there will be no overlapping, *i.e.*, $\sigma_{ij}^{\vec{t}\vec{t}} = 0$. We will model this scenario using the binary indicator variable $\lambda_{ij}^{\vec{t}\vec{t}}$. If any of the conditions $a_j^{\vec{t}-} \geq d_i^{t+}$ or $a_j^{\vec{t}} \leq d_i^t$ occurs, then $\lambda_{ij}^{\vec{t}\vec{t}}$ is zero, *i.e.*,

$$\left((a_j^{\vec{t}-} \geq d_i^{t+}) \vee (a_j^{\vec{t}} \leq d_i^t) \right) \Rightarrow (\lambda_{ij}^{\vec{t}\vec{t}} = 0) \quad (4.13)$$

If $\lambda_{ij}^{\vec{t}}$ is zero, then the overlapping time will be zero, *i.e.*,

$$\begin{aligned} (\lambda_{ij}^{\vec{t}} = 0) &\Rightarrow (\sigma_{ij}^{\vec{t}} = 0) \\ \Leftrightarrow (\sigma_{ij}^{\vec{t}} > 0) &\Rightarrow (\lambda_{ij}^{\vec{t}} = 1), \quad [\text{as } \sigma_{ij}^{\vec{t}} \not\leq 0]. \end{aligned} \quad (4.14)$$

The Equations (4.13)-(4.14) are not in mathematical programming format. Using integer programming modelling rules [45, pages 166, 172-174, 183-184], we can model them as follows:

$$a_j^{\vec{t}-} - d_i^{t+} + \epsilon \leq M(1 - \lambda_{ij}^{\vec{t}}), \quad (4.15)$$

$$d_i^t - a_j^{\vec{t}} + \epsilon \leq M(1 - \lambda_{ij}^{\vec{t}}), \quad (4.16)$$

$$\sigma_{ij}^{\vec{t}} \geq 0, \quad (4.17)$$

$$\sigma_{ij}^{\vec{t}} \leq M\lambda_{ij}^{\vec{t}}. \quad (4.18)$$

When $(a_j^{\vec{t}-} < d_i^{t+})$ and $(a_j^{\vec{t}} > d_i^t)$, we can see from Figure 4.8 that the overlapping time is $\min\{d_i^{t+}, a_j^{\vec{t}}\} - \max\{d_i^t, a_j^{\vec{t}-}\} > 0$. Now we want to model the following logical condition

$$(\lambda_{ij}^{\vec{t}} = 1) \Rightarrow (\sigma_{ij}^{\vec{t}} = \min\{d_i^{t+}, a_j^{\vec{t}}\} - \max\{d_i^t, a_j^{\vec{t}-}\})$$

using mathematical programming. Using integer programming modelling rules we can model this situation as follows:

$$\sigma_{ij}^{\vec{t}} \leq \alpha_i^t \lambda_{ij}^{\vec{t}}, \quad (4.19)$$

$$\sigma_{ij}^{\vec{t}} \leq \beta_j^{\vec{t}} \lambda_{ij}^{\vec{t}}, \quad (4.20)$$

$$\sigma_{ij}^{\vec{t}} \leq d_i^{t+} - a_j^{\vec{t}-} + M(1 - \lambda_{ij}^{\vec{t}}), \quad (4.21)$$

$$\sigma_{ij}^{\vec{t}} \leq a_j^{\vec{t}} - d_i^t + M(1 - \lambda_{ij}^{\vec{t}}). \quad (4.22)$$

Combining Equations (4.15)-(4.18) and Equations (4.19)-(4.22), and using the fact that $M \gg \max\{\alpha_i^t, \beta_j^{\vec{t}}\}$, for any $a_i^{\vec{t}-} \in [a_i^t - \bar{\beta}_i^{\vec{t}}, a_i^t - \underline{\beta}_i^{\vec{t}}]$ and any $d_i^{t+} \in [d_i^t + \underline{\alpha}_i^t, d_i^t + \bar{\alpha}_i^t]$ we

arrive at the following set of equations:

$$\begin{aligned}
a_j^{\bar{t}-} - d_i^{t+} + \epsilon &\leq M(1 - \lambda_{ij}^{t\bar{t}}), \\
d_i^t - a_j^{\bar{t}} + \epsilon &\leq M(1 - \lambda_{ij}^{t\bar{t}}), \\
\sigma_{ij}^{t\bar{t}} &\geq 0, \\
\sigma_{ij}^{t\bar{t}} &\leq \alpha_i^t \lambda_{ij}^{t\bar{t}}, \\
\sigma_{ij}^{t\bar{t}} &\leq \beta_j^{\bar{t}} \lambda_{ij}^{t\bar{t}}, \\
\sigma_{ij}^{t\bar{t}} &\leq d_i^{t+} - a_j^{\bar{t}-} + M(1 - \lambda_{ij}^{t\bar{t}}), \\
\sigma_{ij}^{t\bar{t}} &\leq a_j^{\bar{t}} - d_i^t + M(1 - \lambda_{ij}^{t\bar{t}}).
\end{aligned}$$

Using $a_i^{t-} \in [a_i^t - \bar{\beta}_i^t, a_i^t - \underline{\beta}_i^t]$ and $d_i^{t+} \in [d_i^t + \underline{\alpha}_i^t, d_i^t + \bar{\alpha}_i^t]$, the equations above can be transformed into the following robust formulation:

$$\begin{aligned}
a_j^{\bar{t}} - d_i^t + \epsilon &\leq \underline{\alpha}_i^t + \bar{\beta}_j^{\bar{t}} + M(1 - \lambda_{ij}^{t\bar{t}}), \\
d_i^t - a_j^{\bar{t}} + \epsilon &\leq M(1 - \lambda_{ij}^{t\bar{t}}), \\
\sigma_{ij}^{t\bar{t}} &\geq 0, \\
\sigma_{ij}^{t\bar{t}} &\leq \underline{\alpha}_i^t \lambda_{ij}^{t\bar{t}}, \\
\sigma_{ij}^{t\bar{t}} &\leq \bar{\beta}_j^{\bar{t}} \lambda_{ij}^{t\bar{t}}, \\
\sigma_{ij}^{t\bar{t}} &\leq d_i^t - a_j^{\bar{t}} + \underline{\alpha}_i^t + \bar{\beta}_j^{\bar{t}} + M(1 - \lambda_{ij}^{t\bar{t}}), \\
\sigma_{ij}^{t\bar{t}} &\leq a_j^{\bar{t}} - d_i^t + M(1 - \lambda_{ij}^{t\bar{t}}),
\end{aligned}$$

□

4.5 Modeling overlapping time for left events

Now consider the case when $(i, j, t, \bar{t}) \in \mathcal{E}$. Like the previous case, we need to ensure that after we apply the optimization strategy, \bar{t} still stays the temporally closest train to the left of t . An analogous constraint to that of Equation (4.5) can be easily found by

replacing t and \overleftarrow{t} in Equation (4.5) with \overleftarrow{t} and t respectively as follows:

$$\frac{\left(a_i^t + d_i^t - a_j^{\overleftarrow{t}} - d_j^{\overleftarrow{t}}\right)}{\left(\overline{a}_i^t + \overline{d}_i^t - \overline{a}_j^{\overleftarrow{t}} - \overline{d}_j^{\overleftarrow{t}}\right)} \geq 0. \quad (4.23)$$

Note that the denominator can never be zero on the left hand side of the equation above because of the definition of \overleftarrow{t} in Equation (4.2). To model the left event overlapping time $\sigma_{ij}^{\overleftarrow{t}}$ for all $(i, j, t, \overleftarrow{t}) \in \mathcal{E}$, we propose the following lemma.

Lemma 4.5. *For all $(i, j, t, \overleftarrow{t}) \in \mathcal{E}$, the left event overlapping time $\sigma_{ij}^{\overleftarrow{t}}$ in Equation 4.4 is represented by the following constraints*

$$a_i^t - d_j^{\overleftarrow{t}} + \epsilon \leq \underline{\alpha}_i^{\overleftarrow{t}} + \underline{\beta}_j^t + M(1 - \lambda_{ij}^{\overleftarrow{t}}), \quad (4.24)$$

$$d_j^{\overleftarrow{t}} - a_i^t + \epsilon \leq M(1 - \lambda_{ij}^{\overleftarrow{t}}), \quad (4.25)$$

$$\sigma_{ij}^{\overleftarrow{t}} \geq 0, \quad (4.26)$$

$$\sigma_{ij}^{\overleftarrow{t}} \leq \underline{\alpha}_j^{\overleftarrow{t}} \lambda_{ij}^{\overleftarrow{t}}, \quad (4.27)$$

$$\sigma_{ij}^{\overleftarrow{t}} \leq \underline{\beta}_i^t \lambda_{ij}^{\overleftarrow{t}}, \quad (4.28)$$

$$\sigma_{ij}^{\overleftarrow{t}} \leq d_j^{\overleftarrow{t}} - a_i^t + \underline{\alpha}_i^{\overleftarrow{t}} + \underline{\beta}_j^t + M(1 - \lambda_{ij}^{\overleftarrow{t}}), \quad (4.29)$$

$$\sigma_{ij}^{\overleftarrow{t}} \leq a_i^t - d_j^{\overleftarrow{t}} + M(1 - \lambda_{ij}^{\overleftarrow{t}}), \quad (4.30)$$

where M is a large positive number, ϵ is a small positive number smaller than time granularity considered and $\lambda_{ij}^{\overleftarrow{t}}$ is a binary variable which is one if and only if $\sigma_{ij}^{\overleftarrow{t}}$ is positive.

Proof. The lemma can be easily proved by replacing i, j, t and \overleftarrow{t} with j, i, \overleftarrow{t} and t respectively in Lemma 4.4. \square

4.6 Full optimization model

In this section we collect the objective and all the constraints discussed in the previous sections, and propose our optimization problem to maximize the total duration of overlapping times of the SPSTPs in order to utilize regenerative braking energy produced by

trains in a railway network. The full optimization model is as follows:

$$\begin{aligned}
& \text{maximize} && \sum_{(i,j,t,\tilde{t}) \in \vec{\mathcal{E}}} \sigma_{ij}^{t\tilde{t}} + \sum_{(i,j,t,\tilde{t}) \in \overleftarrow{\mathcal{E}}} \sigma_{ij}^{t\tilde{t}} \\
& \text{subject to} && \\
& && \text{Equations (3.1), (3.3), (3.4), (3.2), (3.5) and (3.7)} \\
& && \forall (i,j,t,\tilde{t}) \in \vec{\mathcal{E}} \quad \text{Equations (4.5),(4.6)-(4.12)} \\
& && \forall (i,j,t,\tilde{t}) \in \overleftarrow{\mathcal{E}} \quad \text{Equations (4.23),(4.24)-(4.30)} \\
& && \forall t \in \mathcal{T} \quad \forall i \in \mathcal{N}^t \quad (a_i^t \geq 0, d_i^t \geq 0) \\
& && \forall (i,j,t,\tilde{t}) \in \mathcal{E} \quad (\lambda_{ij}^{t\tilde{t}} \in \{0,1\}, \sigma_{ij}^{t\tilde{t}} \geq 0),
\end{aligned}$$

where the decision variables are a_i^t , d_i^t , $\lambda_{ij}^{t\tilde{t}}$ and $\sigma_{ij}^{t\tilde{t}}$.

As the model is a MIP with bounds, the optimization problem is \mathcal{NP} -hard (Section 2.5). However, in the next chapter we show that for the size of the railway data spanning six hours, the running time is quite acceptable.

4.7 Limitations of the model

In this section we discuss the limitations of the optimization model, which are as follows.

- The model presented in this chapter has \mathcal{NP} -hard computational complexity. As a result, solving the underlying optimization problem will always require an exponential amount of time in the worst case. When it comes to real life implementation an optimization model, computational tractability is a really important factor. In this regard, seeking an optimization model that can be solved in polynomial time is of vital importance.
- The model tries to transfer regenerative energy from the braking train to the accelerating train by maximizing the overlapping time between the temporal blocks associated with the acceleration and braking phases. Though this is a reasonable strategy, maximizing the overlapping time may not necessarily maximize the transfer of regenerative energy. We address this issue in the second optimization model (Chapter 6).
- The goal of this model is to increase energy efficiency of the timetable by saving regenerative energy of braking trains. However a significant amount electrical energy is also spent by trains while making trips from origin platforms to destination

platforms. So, incorporating an objective to minimize the energy consumption associated with these trips can increase the overall energy efficiency of a timetable. This will be addressed in Chapter 6.

Chapter 5

Numerical experiments for the robust MIP model

In this section we apply our model described Chapter 4 to two different railway networks associated with the Shanghai Metro Network (shown in Figure 5.1) and Dockland Light Railway (shown in Figure 5.2) respectively. For the first study we have access to energy data for the trains. This allows us to calculate the reduction in *effective energy consumption* (explained in Subsection 5.1 below) for the optimal timetables in comparison with the initial timetables. However, for the second study, we do not have access to such energy data, and we can only compare the increase in the total overlapping time. In model (4.6), we have taken $M = 1000$ and $\epsilon = 0.005$.

5.1 Shanghai Metro network

In this subsection we apply our model to nine different problem instances to service PES2-SFM2 of line 8 of Shanghai Metro network, as shown in Figure 5.1. The number of trains, headway times, speed of the involved trains, the grades of the tracks and nature of the energy profile of the associated acceleration and braking phases are different for different instances. Shanghai Metro is the world's largest rapid transit system by route length, second largest by number of stations (after Beijing), and third oldest rapid transit system in mainland China. Line 8, opened on December of 2007, is one of the 14 lines of the Shanghai Metro Network. It passes by some of Shanghai's densest neighborhoods, and has a daily ridership of approximately 1.08 million (2014 peak). This line is 37.4 km long with 28 stations in operation [16]. There are two lines in this network: Line 1 and Line 2. There are fourteen stations in the network denoted by all capitalized words in the figure. Each station has two platforms each on different train lines, *e.g.*, LXM is station

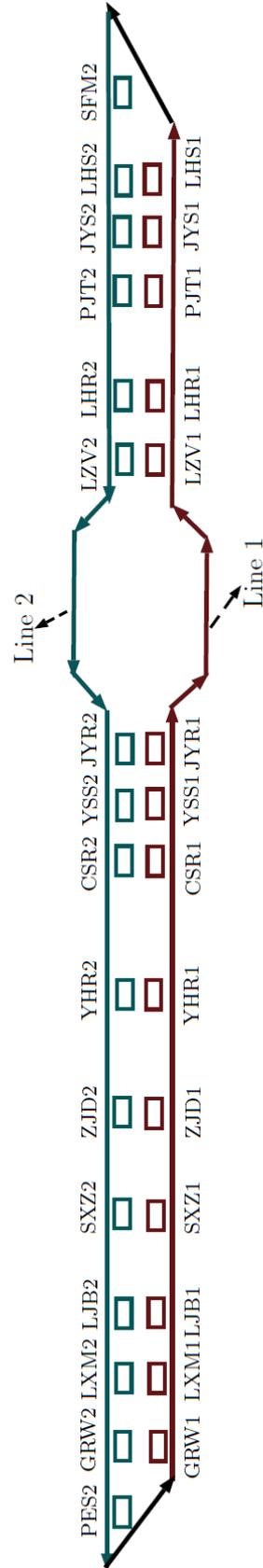


Figure 5.1: First railway network considered for numerical study

that has two opposite platforms: LXM1 and LXM2 on Line 1 and Line 2 respectively. The platforms are denoted by rectangles. The platforms indicated by PES2 and SFM2 are the turn-around points on Line 2, with the crossing-overs being PES2-GRW1 and LHS1-SFM2.

Now we provide some relevant information regarding the railway network in consideration. The line is 37 km long. The average distance between two stations in this network is 1.4 km, with the minimum distance being 738 m (between YHR and ZJD) and maximum distance being 2.6 km (between PJT and LHR). The slope of the track is in $[-2.00453^\circ, 2.00453^\circ]$. The maximum allowable acceleration of a train is 1.04 m/s^2 at accelerating phase. The maximum allowable deceleration rate of a train during coasting phase is -0.2 m/s^2 . The maximum allowable deceleration of a train during braking phase is -0.8 m/s^2 . The conversion factor from electricity to kinetic energy is 0.9, and the conversion factor from kinetic to regenerative braking energy is 0.76. The transmission loss factor of regenerative electricity is 0.1. The mass of the train is in $[229370, 361520]$ kg, with the average mass being 295445 kg.

The data on speed limit is described by Table 8.1 in Appendix. The speed limit data are based on grade and curvature of the tracks, and operational constraints present in the system. For the railway network, the tracks have piece-wise speed limits for trains, *i.e.*, each track (except CSR2-YHR2) is divided into multiple segments, where each segment has a constant speed limit. The track CSR2-YHR2 has only one segment (itself) with a speed limit of 60 km/h. Table 8.1 is provided to us by Thales Canada Inc.

In each instance we have an initial feasible timetable with a duration of six hours. In many railway networks the duration of the off-peak or rush hours is smaller than or equal to six hours, so a timetable that six hours can be sufficient for practical purpose. However there can be exceptions, and in such cases this model may not be very efficient. The number of trains increases as the average headway time decreases. The results of the numerical study are shown in Table 5.1.

The feasible timetables are provided to us by Thales Canada Inc. We have applied our optimization model to find the optimal timetable that maximizes the total overlapping time of the SPSTPs. We see from Table 5.1 that for each instance, our optimization model produces an optimal timetable with significant increase in the total overlapping time in comparison with the initial timetable. Such increase in the total overlapping time would make it possible to save significant amount of electrical energy produced by the braking trains by transferring it to the accelerating trains via the overhead contact lines. We can see that, in all of the cases our model has found the optimal timetables very quickly, the largest runtime being 86.64s. Though the problem is \mathcal{NP} -hard, the hardness

Table 5.1: Results of the numerical study (running time 1200 s) for the railway network in Figure 5.1

Instance number	Number of Trains	Number of Constraints	Binary Variables	Continuous Variables	CPU Time (s)	Nodes Explored	Initial Overlapping Time (s)	Final Overlapping Time (s)	Increase in Overlapping Time (%)	Initial Effective Energy Consumption (kWh)	Final Effective Energy Consumption (kWh)	Reduction in Effective Energy Consumption (%)
1	100	10437	183	3133	86.64	4534	454.26	2828.09	522.57	8594.27	7698.28	10.42
2	112	12166	272	3576	7.26	459	1307.23	4354.50	233.11	14500.05	13740.82	5.23
3	124	12684	188	3846	0.28	0	1190.89	3119.72	161.96	8524.96	7618.28	10.63
4	134	13186	128	4081	0.25	0	315.85	1950.88	517.66	5451.29	4637.33	14.93
5	146	16378	426	4733	0.45	0	2412.92	7314.48	203.14	20223.63	16998.21	15.95
6	156	16264	278	4880	15.34	6	2096.23	4226.93	101.64	14324.78	13319.20	7.02
7	168	18154	390	5346	56.94	1164	1823.10	6299.66	245.55	20274.44	17817.97	12.12
8	178	19650	472	5723	0.51	0	2874.67	8195.41	185.09	21685.04	17619.35	18.75
9	190	22947	785	6390	9.60	0	5858.58	12924.23	120.60	35613.96	33574.48	5.73

comes from the constraints associated with the objective. The feasibility constraints are linear. So, if we had looked for a feasible timetable, it would have been a linear programming problem, which is in complexity class \mathcal{P} and can be solved very efficiently. At the root node the solver relaxes the original problem. So, at the root node we will have a feasible but possibly suboptimal timetable. Depending on the starting point the solver may reach the optimal solution after exploring a small number of nodes. The solution time in this case will be very short, as checking the optimality of such a solution can be done in polynomial time [41, pages 299-300]. In our numerical study we see that the number of nodes explored is quite small for every instance, and for five out of nine instances an optimal solution is found at the root node.

After we get the final timetable, we calculate the total *effective energy consumption* by all the trains involved in SPSTPs and compare it with the original timetables. The effective energy consumption of a train for a trip is defined as the difference between total energy required to make a trip and the amount of energy that is being supplied by a braking train during synchronization process. So, the effective energy consumption is the energy that will be consumed from the electrical substations.

The energy calculation is done using SPSIM [43], and Cubature [23]. SPSIM is a proprietary software owned by Thales Canada Inc. that calculates the power versus time graphs of all the active trains for the original and optimal timetables. Cubature is an open-source Julia package written by Steven G. Johnson, which is used to calculate the effective area under the power versus time graphs to determine 1) the total energy required by the trains during the trips, 2) the total transferred regenerative energy during the SPSTPs, and 3) the effective energy consumption that the difference of the first two quantities. The effective energy consumption of the optimal timetables in comparison with the original ones gets reduced for every instance, with smallest reduction being 5.23% and the largest reduction being 18.74%. Increase in the overlapping time results in energy saving in all cases, but there is a variability in the reduction of the effective energy consumption, which we explain as follows. The effective energy saving from an SPSTP depends on parameters such as speed of the involved trains and nature of the energy profile of the associated acceleration and braking phases. None of these parameters admit any closed form and incorporating these effects in the model may make it computationally intractable. As a result, the overlapping time between two trains may be the same, but the associated energy saving may be completely different.

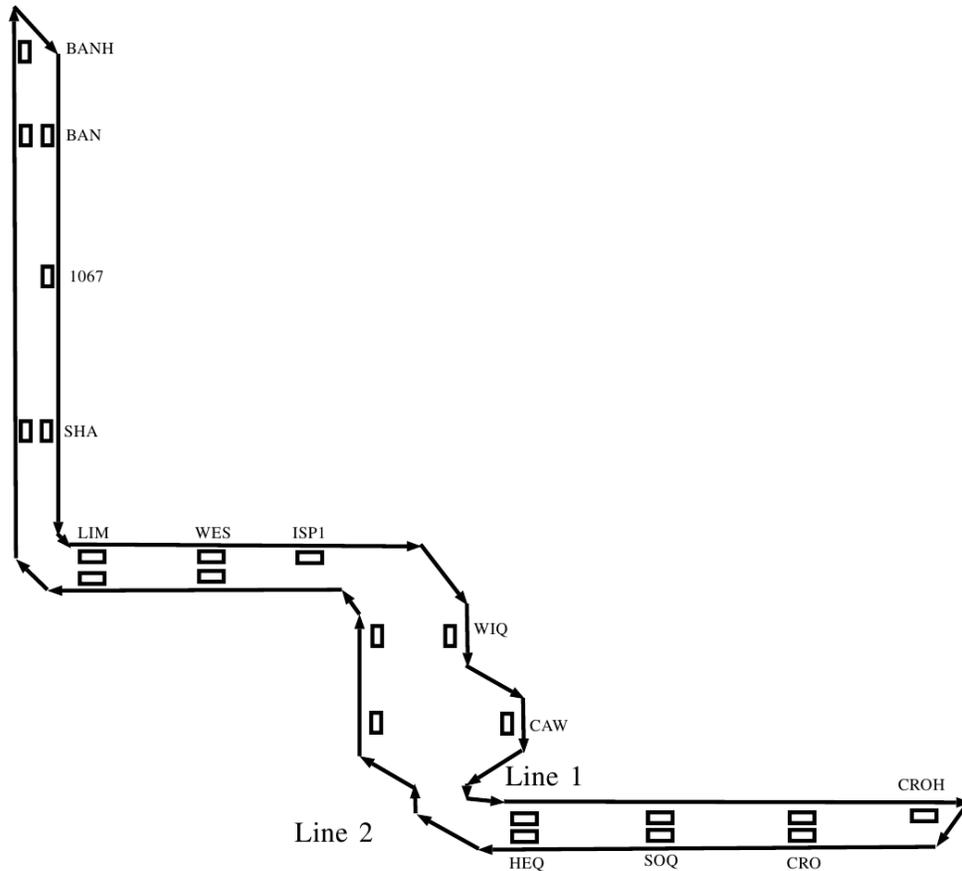


Figure 5.2: Second railway network considered for numerical experiment

5.2 Docklands Light Railway

In this subsection, we present results of our numerical study associated with 17 different problem instances of varying size to a railway network that is a part of the Docklands Light Railway as shown in Figure 5.2. All the experiments were executed on a Intel Core i5-3317U 1.70GHz CPU with 4096 MB of RAM running the Windows 8.1 operating system. We have used IBM ILOG CPLEX Optimization Studio 12.6 academic version with OPL as our modelling language to perform the optimization.

The railway network has two train-lines denoted by Line 1 and Line 2. There are ten stations in this network denoted by capitalized words and each station has two opposite platforms, *e.g.*, BAN is a station which has two opposite platforms: one on Line 1 and the other on Line 2. The platforms denoted by 1067 and ISP1 are intermediate stopping points on Line 1. The platforms indicated by BANH and CROH are turn-around points on Line 1 and Line 2 respectively.

For our numerical study, we have considered 17 different instances with varying head-

Table 5.2: Results of the numerical study (running time 1200 s) for the network in Figure 5.2

Instance number	Number of Trains	Number of Constraints	Binary Variables	Continuous Variables	Explored Nodes	CPU Time (s)	Initial Overlapping Time	Final Overlapping Time	Increase in Overlapping Time (%)
1	108	4223	100	2469	0	1.09	0	1000	Infinity
2	121	6370	346	2999	30	3.8	1887	3460	83.36
3	144	8496	542	3699	349	9.54	2151	5420	151.98
4	156	7253	308	3729	6	4.52	222	3080	1287.39
5	180	9619	534	4479	285	11.48	379	5340	1308.97
6	192	11610	762	4971	228676	1200	2887	7425	157.19
7	204	12047	768	5241	819	25.44	2337	7680	228.63
8	216	14108	1006	5743	203	17.5	3837	10060	162.18
9	228	15645	1170	6167	1450	43.54	3661	11700	219.58
10	240	19400	1650	6911	1672	66.71	6968	16500	136.80
11	252	17401	1308	6833	1428	69.51	4752	13080	175.25
12	263	18096	1356	7121	1027	56.81	5284	13560	156.62
13	275	15425	918	6947	156	17.06	4318	9180	112.60
14	288	20722	1614	7927	1423	122.39	5515	16140	192.66
15	300	20319	1500	8077	1475	116.11	3989	15000	276.03
16	311	18746	1224	8041	1224	115.8	4372	12240	179.96
17	336	27714	2388	9753	16200	1200	9666	23680	144.98

way times and number of trains. As the headway time decreases, the number of trains in the network increases. In each instance we have an initial feasible timetable with a duration of six hours. We have applied our optimization model to find the optimal timetable that maximizes the total overlapping time of the SPSTPs.

We see from Table 5.2 that for each instance, our optimization model produces an optimal timetable with significant increase in the total overlapping time in comparison with the initial timetable, with the minimum one being 83.36%. Such increase in the total overlapping time would make it possible to save significant amount of electrical energy produced by the braking trains by transferring it to the accelerating trains via a *third rail*. A third rail is a semi-continuous conductor of electricity, which is placed between two opposite tracks. Optimal solutions are obtained for all the instances within the running time of 1200 s, except two instances - instances 6 and 17 (the optimality gap being 0.97% and 0.31% respectively). In both cases, a 95% optimal solution (5% optimality gap) is found within less than 2 minutes. If we remove the time limit, and let the algorithm run indefinitely, then instances 6 and 17 reach optimal solutions after 2 hours 27 minutes and 3 hours 5 minutes, respectively.

These two instances for which optimality could not be reached within the time limit can be explained as follows. The CPLEX solver finds the first candidate solution via general purpose (closed source) heuristics with no guarantee of being close to the optimal point [21]. Though empirically the CPLEX heuristics tend to perform quite well for most instances, it can also happen that the first candidate solution is far away from the optimal solution. In such a case, reaching the optimal solution can take longer and may not even be reached within the time limit.

Chapter 6

A two-stage linear programming model

In Chapter 4, we presented a robust mixed integer optimization model to increase energy-efficiency of railway networks. However, as described in Section 4.7, the model suffered from some drawbacks associated with computational complexity and energy saving strategy. To overcome those drawbacks, in this chapter we propose a novel two-stage linear optimization model to calculate energy-efficient timetables in electric railway networks. This chapter is organized as follows. The motivations behind this model is described in Section 6.1. The first stage of the optimization model is presented in Section 6.2. This stage minimizes the total energy consumed by the trains. Section 6.3 formulates the second stage of the optimization problem that additionally maximizes the utilization of regenerative braking energy. Both the stages of our optimization model are linear programs, whereas the optimization model in Chapter 4 and the optimization models in related works are \mathcal{NP} -hard. Section 6.4 describes the limitations of the optimization model.

A flow-chart of the two stages of the optimization model is shown in Figure 6.1.

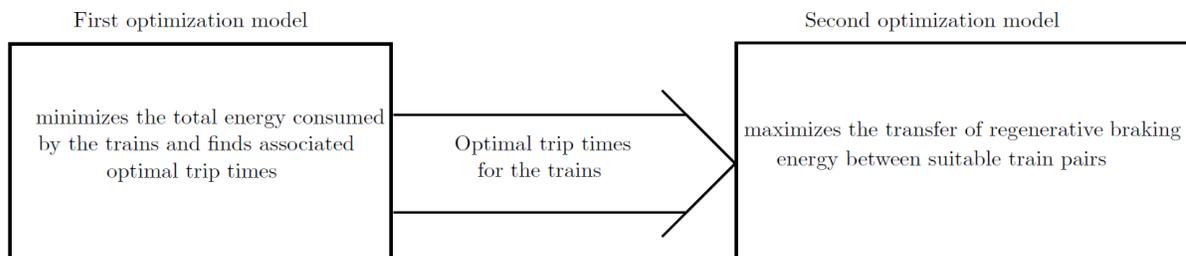


Figure 6.1: A flow-chart of the two stages of the optimization model

6.1 Motivation behind the model

Motivations behind the two-stage optimization model stem from the limitations of the robust mixed integer optimization model presented in Chapter 4, and are described as follows.

1. The robust mixed integer optimization model saves regenerative energy by maximizing the overlapping time between the temporal blocks associated with the acceleration and braking phases. This is a reasonable strategy, but it may not result in the the transfer of maximum amount regenerative energy. The transfer of regenerative energy between two trains depends on the nature of individual electrical power consumption/regeneration of the trains involved versus time graphs and to maximize the transfer of regenerative energy between these trains one should aim to schedule the trains such a way that the overlapped area between the two graphs is maximized.
2. Besides saving the regenerative energy, another way of increasing energy-efficiency is to reduce the energy consumption of trains itself. Trains consume most of the required electrical energy during the acceleration phases of making trips between platforms. So, one should consider an objective to minimize the energy consumption associated with these trips.
3. The robust mixed integer optimization model has \mathcal{NP} -hard computational complexity. Thus solving the underlying optimization problem will always require an exponential amount of time in the worst case even with state of the art commercial solvers. This worst-case behavior was indeed exhibited by the numerical study performed for the robust mixed integer optimization model for timetables spanning only 6 hours (instances 6 and 17 in Table 5.2). This can a be severe limitation in practice, as often we are interested in calculating energy-efficient timetables spanning a full service period of one day in a short period of time. In such a case, an optimization model that can be solved in polynomial time can be of great interest.

Motivated by the reasons above we propose the two-stage optimization model in this chapter. The rest of the chapter is devoted to its description.

6.2 Stage one of the linear programming model

In this section we formulate the first optimization model that minimizes the total energy consumed by all trains in the railway network. The organization of this section is

as follows. First, in order to keep the proofs less cluttered, we introduce an equivalent constraint graph notation. Then we formulate and justify the first optimization problem. Finally, we show that the nonlinear objective of the initial optimization model can be approximated as a linear one by applying least-squares. This results in a linear optimization problem, which has the interesting property that its optimal solution is attained by an integral vector.

6.2.1 Converting the initial notation into an equivalent constraint graph

Each of the constraints described by Equations (3.1)-(3.7) is associated with two event times (either arrival or departure time of trains at stations), where one of them precedes another by a time difference dictated by the time window of that constraint. This observation helps us to convert our initial notation into an equivalent constraint graph notation which we describe as follows.

Converting the initial notation into an equivalent constraint graph All event times in the original notation are treated as nodes in the constraint graph, the set of those nodes is denoted by $\bar{\mathcal{N}}$ and the value associated with a node $i \in \bar{\mathcal{N}}$ is denoted by x_i , which represents the arrival or departure time of some train from a platform.

- *Nodes of the constraint graph:* Consider any two nodes in the constraint graph; if there exists a constraint between the two in the original notation, then in the constraint graph we create a directed arc between them, the start node being the first event and the end node being the later one. The set of arcs thus created in the constraint graph is denoted by $\bar{\mathcal{A}}$. Note that there cannot be more than one arc between two nodes in the constraint graph.
- *Arcs of the constraint graph:* With each arc $(i, j) \in \bar{\mathcal{A}}$ we associate a time window $[l_{ij}, u_{ij}]$ with their values determined from the Equations (3.1)-(3.7). So, each arc $(i, j) \in \bar{\mathcal{A}}$ corresponds to a constraint of the form $l_{ij} \leq x_j - x_i \leq u_{ij}$. The set of all arcs associated with trip time constraints is expressed by $\bar{\mathcal{A}}_{\text{trip}} \subset \bar{\mathcal{A}}$.

Example Figure 6.2 represents a very simple network with 7 stations represented by black dots and 2 trains represented by the directed rectangles. The pointed edge of the symbol indicates the direction. In this network, $\mathcal{T} = \{1, 2\}$ and the stations are enumerated as $\{1, 2, 3, 4, 5, 6, 7\}$. There are two train lines as shown in the figure.

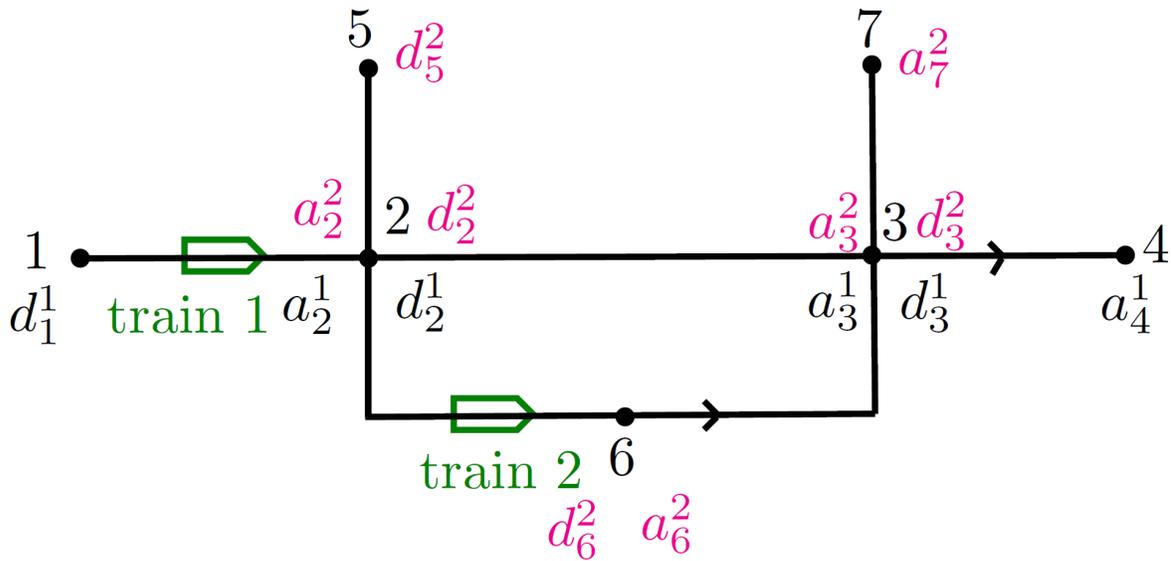


Figure 6.2: Example of a very simple railway network.

It should be noted that nodes represent stations, not platforms. Nodes 2 and 3 represent interchange stations, where there are two platforms for both lines on different levels. So, the event times at node 2 and node 3 are associated with different platforms and are differentiated using black and magenta colors. The set of tracks visited by train 1 is $\mathcal{A}^1 = \{(1, 2), (2, 3), (3, 4)\}$, and the set of tracks visited by train 2 is $\mathcal{A}^2 = \{(5, 2), (2, 6), (6, 3), (3, 7)\}$. The set of all tracks is then $\mathcal{A} = \mathcal{A}^1 \cup \mathcal{A}^2$. The event times corresponding to train 1 and train 2 are shown in black and magenta colours respectively in Figure 6.2. Applying the conversion process described above we can convert the initial notation in Figure 6.2 to the constraint graph shown in Figure 6.3. .

6.2.2 Description of the first stage of the linear programming model

Recall that, a train consumes most of its required electrical energy during the acceleration phase of making a trip from an origin platform to a destination platform (Section 1.1). Trip time constraints, described in Section 3.3.1, play the most important role in energy consumption and regenerative energy production of trains. Once the trip time for a trip is fixed, an energy optimal speed profile can be calculated efficiently by existing software [43], [20, page 285], such as *e.g.*, Thales **T**rain **K**inetics, **D**ynamics and **C**ontrol (TKDC) Simulator in our case. The TKDC simulator assumes maximum accelerating - speed holding - coasting - maximum braking strategy for calculation of speed profile.

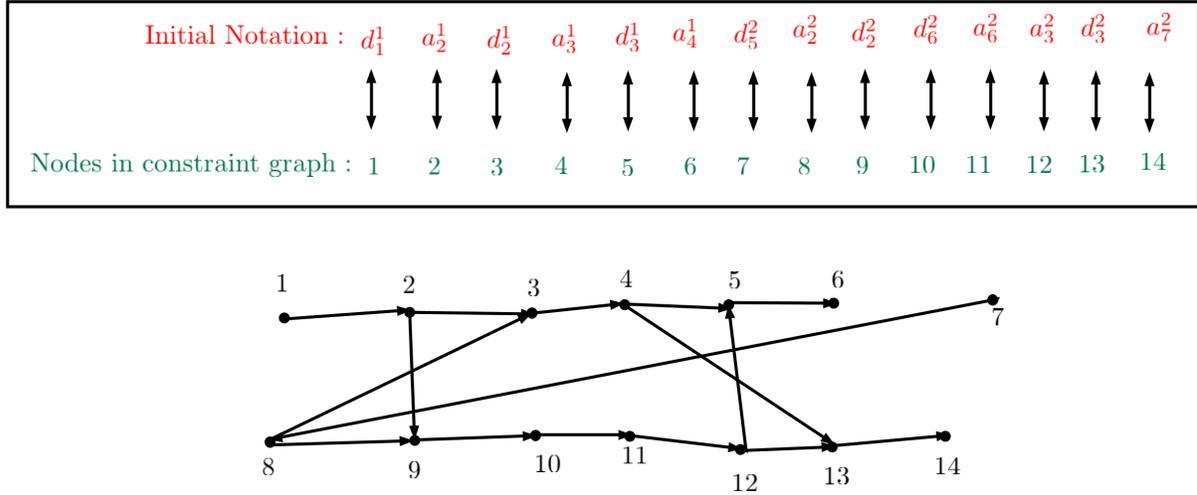


Figure 6.3: The constraint graph for the railway network of Figure 6.2.

Theoretically this is the optimal speed profile according to the monograph [20]. For calculation of the optimal speed profile of a train while making a trip, we refer the interested reader to the highly cited papers [22, 18, 19, 25, 31]. The electrical power consumption and regeneration of a train on a track is determined by its speed profile, so the optimal speed profile also gives the power versus time graph (*power graph* in short) for that trip. However, in the total railway service period there are many active trains, whose movements are coupled by the associated constraints. So, finding the energy-minimal trip time for a single trip in an isolated manner can result in a infeasible timetable. Consider an arc $(i, j) \in \bar{\mathcal{A}}_{\text{trip}}$ in the constraint graph, associated with some trip time constraint. Let us denote the energy consumption function for that trip $f_{ij} : \mathbf{R}_{++} \rightarrow \mathbf{R}_{++}$ with argument $(x_j - x_i)$. The first optimization problem with the objective to minimize the total energy consumption of the trains can be written as:

$$\begin{aligned}
 & \text{minimize} && \sum_{(i,j) \in \bar{\mathcal{A}}_{\text{trip}}} f_{ij}(x_j - x_i) \\
 & \text{subject to} && l_{ij} \leq x_j - x_i \leq u_{ij}, \quad \forall (i, j) \in \bar{\mathcal{A}} \\
 & && 0 \leq x_i \leq m, \quad \forall i \in \bar{\mathcal{N}},
 \end{aligned} \tag{6.1}$$

where the decision vector is $(x_i)_{i \in \bar{\mathcal{N}}} \in \mathbf{R}^{|\bar{\mathcal{N}}|}$.

The exact analytical form of every component of the objective function, *i.e.*, $f_{ij}(x_j - x_i)$ for $(i, j) \in \bar{\mathcal{A}}_{\text{trip}}$ is not known and may be intractable [19]. However, irrespective of the exact analytical form, the energy function can be shown to be monotonically decreasing in trip time, *i.e.*, it is *non-increasing* with the increase in trip time, if the optimal speed profile is followed [35]. Even when a train is manually driven with possibly suboptimal driving strategies, the average energy consumption of the train is found empirically to

be monotonically decreasing in the trip time [38].

Also, the energy function is relatively easy to measure in practice [20, Section 1.5]. For any $(i, j) \in \bar{A}_{\text{trip}}$, we denote the measured trip times $(x_j^{(1)} - x_i^{(1)}), \dots, (x_j^{(p)} - x_i^{(p)})$ and the corresponding energy consumption data $f_{ij}^{(1)}, \dots, f_{ij}^{(p)}$.

In any subway system, the amount by which the trip time is allowed to vary in Equations (3.1) and (3.2) is typically on the order of seconds [29], which motivates us to make the following assumption.

Assumption 6.1. *The amount by which the trip time is allowed to vary is on the order of seconds, i.e., for any the trip time window is on the order of seconds.*

The monotonically decreasing nature of the energy function together with Assumption 6.1 allows us to approximate the energy function $f_{ij}(x_j - x_i)$ as an affine function. Recall that in practice, we can measure the energy $f_{ij}^{(1)}, \dots, f_{ij}^{(p)}$ and associated trip times $(x_j^{(1)} - x_i^{(1)}), \dots, (x_j^{(p)} - x_i^{(p)})$, which is obtainable easily with present technology [20, Section 1.5].

Now we want to formulate an optimization problem which will provide us with the best possible affine approximation of the energy function $f_{ij}(x_j - x_i)$. We do so by applying least-squares and fit a straight line through measured energy versus trip time data. We seek an affine function $c_{ij}(x_j - x_i) + b_{ij} = (x_j - x_i, 1)^T(c_{ij}, b_{ij})$ where we want to determine c_{ij} and b_{ij} .

The affine function approximates the measured energy in the least-squares sense as follows:

$$\begin{aligned} (c_{ij}, b_{ij}) &= \operatorname{argmin}_{(\tilde{c}_{ij}, \tilde{b}_{ij})} \sum_{k=1}^p \left(\tilde{c}_{ij}(x_j^{(k)} - x_i^{(k)}) + \tilde{b}_{ij} - f_{ij}^{(k)} \right)^2 \\ &= \operatorname{argmin}_{(\tilde{c}_{ij}, \tilde{b}_{ij})} \left\| \begin{bmatrix} (x_j^{(1)} - x_i^{(1)}, 1)^T \\ \vdots \\ (x_j^{(p)} - x_i^{(p)}, 1)^T \end{bmatrix} \begin{bmatrix} \tilde{c}_{ij} \\ \tilde{b}_{ij} \end{bmatrix} - \begin{bmatrix} f_{ij}^{(1)} \\ \vdots \\ f_{ij}^{(p)} \end{bmatrix} \right\|_2^2 \end{aligned} \quad (6.2)$$

The problem above is an unconstrained optimization problem with convex quadratic differentiable objective. So, as described in Section 2.3.3, it can be solved by taking the gradient with respect to $(\tilde{c}_{ij}, \tilde{b}_{ij})$, setting the result equal to zero vector and then solving for $(\tilde{c}_{ij}, \tilde{b}_{ij})$. This yields the following closed form solution:

$$\begin{bmatrix} c_{ij} \\ b_{ij} \end{bmatrix} = \left(\begin{bmatrix} (x_j^{(1)} - x_i^{(1)}, 1)^T \\ \vdots \\ (x_j^{(p)} - x_i^{(p)}, 1)^T \end{bmatrix}^T \begin{bmatrix} (x_j^{(1)} - x_i^{(1)}, 1)^T \\ \vdots \\ (x_j^{(p)} - x_i^{(p)}, 1)^T \end{bmatrix} \right)^{-1} \begin{bmatrix} (x_j^{(1)} - x_i^{(1)}, 1)^T \\ \vdots \\ (x_j^{(p)} - x_i^{(p)}, 1)^T \end{bmatrix}^T \begin{bmatrix} f_{ij}^{(1)} \\ \vdots \\ f_{ij}^{(p)} \end{bmatrix} \quad (6.3)$$

Using Equation (6.3), we can approximate the nonlinear objective of the optimization problem (6.1) as an affine one: $\sum_{(i,j) \in \bar{\mathcal{A}}_{\text{trip}}} c_{ij}(x_i - x_j) + b_{ij}$. A measurement of the quality of such fittings is given by the *coefficient of determination*, which can vary between 0 to 1, with 0 being the worst and 1 being the best [24, page 518]. In our numerical studies the mean coefficient of determination of the energy fittings over all the different trips of all the trains is found to be 0.9483 with a standard deviation of 0.05, which justifies our approach. We can also discard the b_{ij} s from the objective, as it has no impact on the minimizer. Thus we arrive at the following linear optimization problem to minimize the total energy consumption of the trains:

$$\begin{aligned} & \text{minimize} && \sum_{(i,j) \in \bar{\mathcal{A}}_{\text{trip}}} c_{ij}(x_j - x_i) \\ & \text{subject to} && l_{ij} \leq x_j - x_i \leq u_{ij}, \quad \forall (i,j) \in \bar{\mathcal{A}} \\ & && 0 \leq x_i \leq m, \quad \forall i \in \bar{\mathcal{N}}. \end{aligned} \tag{6.4}$$

Note that, we have not used the same cost-time curve for all the trips. Each of the constituent parts $c_{ij}(x_j - x_i)$ of the objective function $\sum_{(i,j) \in \bar{\mathcal{A}}_{\text{trip}}} c_{ij}(x_j - x_i)$ in Problem (12) represents approximated affine function for each of the trips considered in the optimization problem. If optimal speed profiles for the trips are available to the railway management, from the first optimization model the optimal trip times for those optimal speed profiles can be found. If available speed profiles are suboptimal, then the first optimization model would still produce an energy-efficient timetable with best trip times subject to the available speed profiles.

An important property of this optimization model is that the polyhedron associated with optimization problem has only integer vertices, so the optimal value is attained by an integral vector. A necessary and sufficient condition of integrality of the vertices of a polyhedron is given by the following theorem [40, page 269, Theorem 19.3], which we will use to prove the subsequent proposition.

Theorem 6.1. *Let A be a matrix with entries 0, +1, or -1 . For all integral vectors a, b, c, d the polyhedron $\{x \in \mathbf{R}^n \mid c \preceq x \preceq d, a \preceq Ax \preceq b\}$ has only integral vertices if and only if for each nonempty collection of columns of A , denoted by C , there exist two subsets, C_1 and C_2 such that $C_1 \cup C_2 = C, C_1 \cap C_2 = \emptyset$, and the sum of the columns in C_1 minus the sum of the columns in C_2 is a vector with entries 0, 1 and -1 .*

Proposition 6.2. *The optimization problem (6.4) has an integral optimal solution.*

Proof. We write the problem (6.4) in vector form. We construct a cost vector c , such that a component of that vector is c_{ij} if it is associated with a trip time constraint

in the original notation, and zero otherwise. Construct integral vectors $l = (l_{ij})_{(i,j) \in \bar{\mathcal{A}}}$, $u = (u_{ij})_{(i,j) \in \bar{\mathcal{A}}}$ and matrix $A \in \{-1, 0, 1\}^{|\bar{\mathcal{A}}| \times |\bar{\mathcal{N}}|}$ such that the (k, i) th entry of the matrix A , denoted by a_{ki} , is associated with the k th hyperarc and i th node of the constraint graph as follows:

$$a_{ki} = \begin{cases} 1 & \text{if node } i \text{ is the end node of hyperarc } k, \\ -1 & \text{if node } i \text{ is the start node of hyperarc } k, \\ 0 & \text{otherwise.} \end{cases}$$

So, the vector form of the optimization problem (6.4) is:

$$\begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && l \preceq Ax \preceq u, \\ & && 0 \preceq x \preceq m\mathbf{1}. \end{aligned} \tag{6.5}$$

Consider any nonempty collection of columns of A denoted by C . Take $C_1 = C$ and $C_2 = \emptyset$. Then the sum of the columns in C_1 minus the sum of the columns in C_2 will be a vector with entries 0, 1 and -1 , because in A there cannot exist more than one row corresponding to an arc between two nodes of the constraint graph and each such row has exactly two nonzero entries, a $+1$ and a -1 . So, by Theorem 6.1 the polyhedron $\{x \in \mathbf{R}^{|\bar{\mathcal{N}}|} : l \preceq Ax \preceq u, 0 \preceq x \preceq m\mathbf{1}\}$ has only integral vertices and optimizing the linear objective in problem (6.5) over this polyhedron will result in an integral solution. \square

After solving the linear programming problem (6.4), we obtain an integral timetable, which we will call the energy minimizing timetable (**EMT**). We denote the optimal decision vector of this timetable by \bar{x} in the constraint graph notation and $((\bar{a}_i^t, \bar{d}_i^t)_{i \in \mathcal{N}})_{t \in \mathcal{T}}$ in the original notation.

6.3 Stage two of the linear programming model

In this section we modify the trip time constraints such that the total energy consumption of the final timetable is kept at the same minimum as the EMT. Then, we describe our optimization strategy aimed to maximize the utilization of regenerative energy of braking trains, and we present the second optimization model.

6.3.1 Keeping the total energy consumption at minimum

In any feasible timetable, if the trip times are kept to be the same as the ones obtained from the EMT, then the energy optimal speed profiles for all trains will be the same. As a result, the energy consumption associated with that timetable will remain at the same minimum as found in the EMT. So, in the second optimization problem, instead of using the trip time constraint, for every trip we fix the trip time to the value in the EMT, *i.e.*,

$$\forall t \in \mathcal{T}, \quad \forall (i, j) \in \mathcal{A}^t, \quad a_j^t - d_i^t = \bar{a}_j^t - \bar{d}_i^t, \quad (6.6)$$

and

$$\forall (i, j) \in \varphi, \quad \forall (t, t') \in \mathcal{B}_{ij}, \quad a_j^{t'} - d_i^{t'} = \bar{a}_j^{t'} - \bar{d}_i^{t'}. \quad (6.7)$$

For all other constraints, bounds are allowed to vary as described by Equations (3.3)-(3.7). As a consequence of fixing all trip times, the power graph of every trip made by any train becomes known to us, since it depends on the corresponding optimal speed profile calculated in real-time by existing software [43], [20, page 285].

6.3.2 Maximizing the utilization of regenerative energy of braking trains

In this subsection we describe our strategy to maximize the utilization of the regenerative energy produced by the braking trains. Strategies based on transfer of regenerative braking energy back to the electrical grid requires specialized technology such as reversible electrical substations [15]. A strategy based on storing is not feasible with present technology, because storage options such as super-capacitors, fly-wheels, *etc.*, have drastic discharge rates besides being too expensive [5, page 66], [13, page 92] (see Section 4.2 for details). As mentioned in Chapter 4, a better strategy that can be used with existing technology [10] is to transfer the regenerative energy of a braking train to a nearby and simultaneously accelerating train, if both of them operate under the same electrical substation. We call such pairs of trains *suitable train pairs*. So our objective is to maximize the total overlapped area between the graphs of power consumption and regeneration of all suitable train pairs. To model this mathematically, we are faced with the following tasks:

- Define suitable train pairs, which have already been described in details in Section 4.2.,

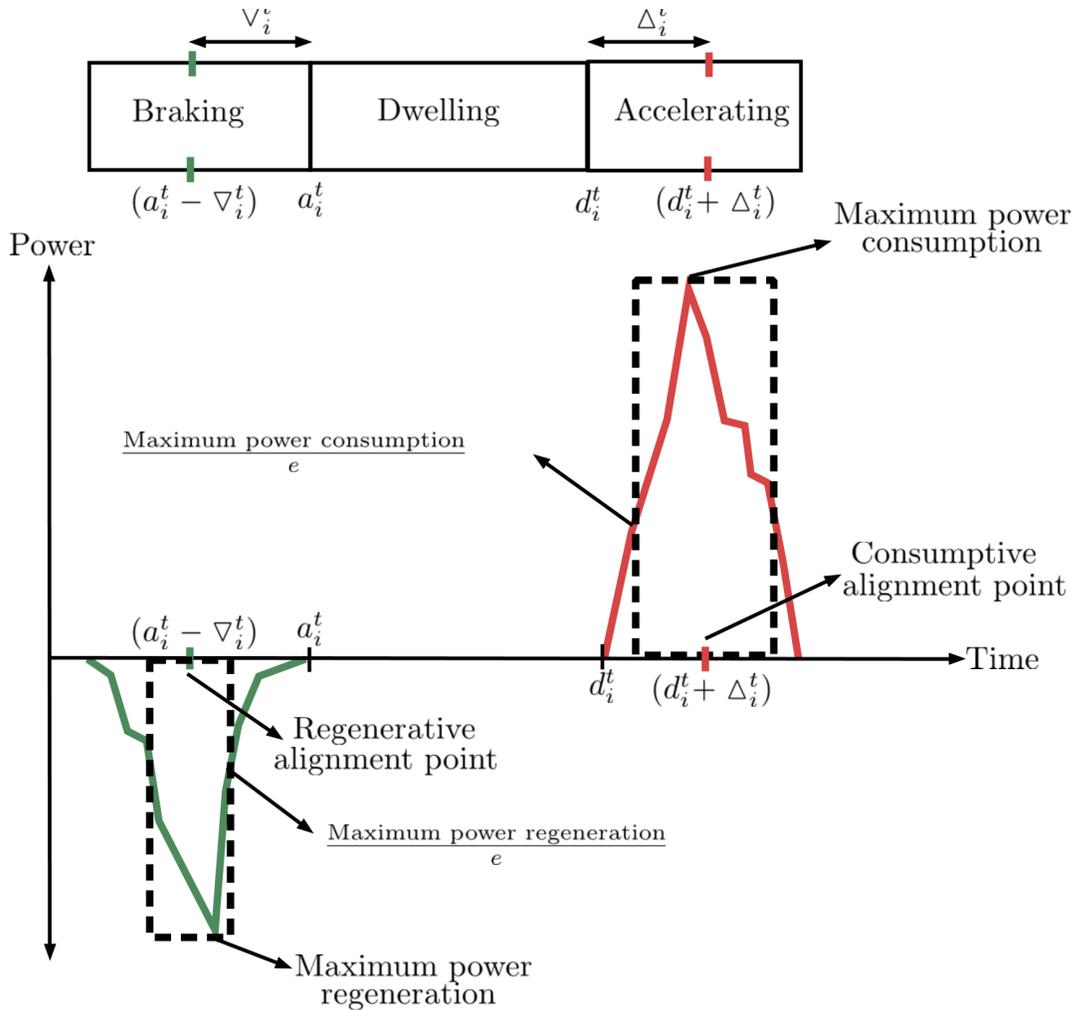


Figure 6.4: Applying $\frac{1}{e}$ heuristic to power graphs

- Provide a tractable description of the overlapped area between power graphs of such a pair, which is described in the next subsection.

6.3.3 Description of the overlapped area between power graphs

The power graph during accelerating and braking is highly nonlinear in nature with no analytic form, as shown in Figure 6.4. So, maximizing the exact overlapped area will lead to an intractable optimization problem. However, as discussed in Section 2.4, the existence of dominant peaks with sharp falls in the power graph allows us to apply $\frac{1}{e}$ heuristic to approximate the power graphs as rectangles. The $\frac{1}{e}$ heuristic is applied as follows (see Figure 6.4). Recall that, the height of the rectangle is the maximum power, and the width is the interval with extreme points corresponding to power dropped at

$1/e$ of the maximum. For the sharp drop from the peak, such rectangles are very robust approximations to the original power graph containing the most concentrated part of the energy, *e.g.*, if the drop were exponential, then the energy contained by the rectangle would have been exactly equal to that of the original curve [33, page 33-34]. After converting both the power graphs to rectangles, maximizing the overlapped area under those rectangles is equivalent to aligning the midpoint of the width of the rectangles; we call such a midpoint **regenerative or consumptive alignment point**. These alignment points act as virtual peaks of the approximated power graphs. As shown in Figure 6.4, for a train t in its braking phase prior to its arrival at platform i , the relative distance of a_i^t from the regenerative alignment point is denoted by ∇_i^t , while during acceleration the relative distance of the consumptive alignment point from d_i^t is denoted by Δ_j^t . Note that both relative distances are known parameters for the current optimization problem.

6.3.4 Description of the second stage of the linear programming model

Consider an element $(i, j, t, \bar{t}) \in \bar{\mathcal{E}}$. To ensure the transfer of maximum possible regenerative energy from the braking train \bar{t} to the accelerating train t , we aim to align both their alignment points such that $d_i^t + \Delta_i^t = a_j^{\bar{t}} - \nabla_j^{\bar{t}}$, or keep them as close as possible otherwise. Similarly, for any $(i, j, t, \bar{t}) \in \bar{\mathcal{E}}$, our objective is $d_j^{\bar{t}} + \Delta_j^{\bar{t}} = a_i^t - \nabla_i^t$, or as close as possible. Let a decision vector y be defined as

$$y = \left((d_i^t + \Delta_i^t - a_j^{\bar{t}} + \nabla_j^{\bar{t}})_{(i,j,t,\bar{t}) \in \bar{\mathcal{E}}}, (d_j^{\bar{t}} + \Delta_j^{\bar{t}} - a_i^t + \nabla_i^t)_{(i,j,t,\bar{t}) \in \bar{\mathcal{E}}} \right). \quad (6.8)$$

Then our goal comprises of two parts: 1) maximize the number of zero components of y which corresponds to minimizing $\mathbf{card}(y)$, and 2) keep the nonzero components as close to zero as possible which corresponds to minimizing the ℓ_1 norm of y , $\|y\|_1$. Combining these two we can write the exact optimization problem as follows:

$$\begin{aligned} & \text{minimize} \quad \mathbf{card}(y) + \gamma \|y\|_1 \\ & \text{subject to} \\ & \text{Equations (3.3) – (3.7), (6.6), (6.7), (6.8),} \\ & 0 \leq a_i^t \leq m, 0 \leq d_i^t \leq m, \quad \forall i \in \mathcal{N}^t, \quad \forall t \in \mathcal{T}, \end{aligned} \quad (6.9)$$

where γ is a positive weight, and decision variables are a , d and y . The objective function is nonconvex as shown next. Take the convex combination of the vectors $2e_1/\gamma$ and 0

with convex coefficients $1/2$. Then,

$$\mathbf{card}\left(\frac{e_1}{\gamma}\right) + \gamma \left\| \frac{e_1}{\gamma} \right\|_1 = 2 > \frac{1}{2} \left(\mathbf{card}\left(\frac{2e_1}{\gamma}\right) + \gamma \left\| \frac{2e_1}{\gamma} \right\|_1 \right) + \frac{1}{2} (\mathbf{card}(0) + \gamma \|0\|_1) = 1.5,$$

and thus violates definition of a convex function. As a result, problem (6.9) is a non-convex problem. Note that if we remove the cardinality part from the objective, then it reduces to a convex optimization problem because the constraints are affine and the objective is the ℓ_1 norm of an affine transformation of the decision variables [8, pages 72, 79, 136-137]. Such problems are often called convex-cardinality problem and are of \mathcal{NP} -hard computational complexity in general [9]. An effective yet tractable numerical scheme to achieve a low-cardinality solution in a convex-cardinality problem is the ℓ_1 norm heuristic, where $\mathbf{card}(y)$ is replaced by $\|y\|_1$, thus converting problem (6.9) into a convex optimization problem. This is described by problem (6.10) below. The ℓ_1 norm heuristic is supported by extensive numerical evidence with successful applications to many fields, *e.g.*, robust estimation in statistics, support vector machine in machine learning, total variation reconstruction in signal processing, compressed sensing *etc.* In the next section we show that in our problem too, the ℓ_1 norm heuristic produces excellent results. Intuitively, the ℓ_1 norm heuristic works well, because it encourages sparsity in its arguments by incentivizing exact alignment between regenerative alignment points with the associated consumptive ones [8, pages 300-301]. We provide a theoretical justification for the use of ℓ_1 norm in our case as follows.

Proposition 6.3. *The convex optimization problem described by*

$$\begin{aligned} & \text{minimize} && \|y\|_1 \\ & \text{subject to} && \\ & && \text{Equations(3.3) – (3.7), (6.6), (6.7), (6.8),} \\ & && 0 \leq a_i^t \leq m, 0 \leq d_i^t \leq m, \quad \forall i \in \mathcal{N}^t, \quad \forall t \in \mathcal{T}, \end{aligned} \tag{6.10}$$

is the best convex approximation of the nonconvex problem (6.9) from below.

Proof. Both problems (6.10) and (6.9) have the same constraint set, so we need to focus on the objective only. Recall from Section 2.3.2 that the best convex approximation of a nonconvex function $f : C \rightarrow \mathbf{R}$ (where C is any set) from below is given by its convex envelope $\mathbf{env} f$ on C . The function $\mathbf{env} f$ is the largest convex function that is an under estimator of f on C , *i.e.*,

$$\mathbf{env} f = \sup\{\tilde{f} : C \rightarrow \mathbf{R} \mid \tilde{f} \text{ is convex and } \tilde{f} \leq f\},$$

where \sup stands for the supremum, *i.e.*, the least upper bound of the set. The definition implies, $\mathbf{epi\ env\ } f = \mathbf{conv\ epi\ } f$.

From Equation (6.8) we see that y is an affine transformation of a and d , and from the last constraints of problem (6.9) we see that both a and d are upper bounded by m , *i.e.*, $\|a\|_\infty \leq m$ and $\|d\|_\infty \leq m$. So there exists a positive number P such that $\|y\|_\infty \leq P$. The domain of y is bounded in an ℓ_∞ ball with radius P . So, as discussed in Section 2.3.2, we have $\mathbf{env\ card\ } (y) = \frac{1}{P}\|y\|_1$. As a result, the best convex approximation of the objective from below is $\frac{1}{P}\|y\|_1 + \gamma\|y\|_1 = (\frac{1}{P} + \gamma)\|y\|_1$. As the coefficient $(\frac{1}{P} + \gamma)$ is a constant for a particular optimization problem, it can be omitted, and thus we arrive at the claim. \square

Using the epigraph approach (See Section 2.3.6), we can transform the convex problem (6.10) into a linear program as follows. For each $(i, j, t, \vec{t}) \in \vec{\mathcal{E}}$ and each $(i, j, t, \overleftarrow{t}) \in \overleftarrow{\mathcal{E}}$, we introduce new decision variables $\theta_{ij}^{\vec{t}, \vec{t}}$ and $\theta_{ij}^{\overleftarrow{t}, \overleftarrow{t}}$ respectively, such that $\theta_{ij}^{\vec{t}, \vec{t}} \geq |d_i^{\vec{t}} + \Delta_i^{\vec{t}} - a_j^{\vec{t}} + \nabla_j^{\vec{t}}|$ and $\theta_{ij}^{\overleftarrow{t}, \overleftarrow{t}} \geq |d_j^{\overleftarrow{t}} + \Delta_j^{\overleftarrow{t}} - a_i^{\overleftarrow{t}} + \nabla_i^{\overleftarrow{t}}|$. Then, the convex optimization problem can be converted into the following linear problem:

$$\begin{aligned}
& \text{minimize} && \sum_{(i,j,t,\vec{t}) \in \vec{\mathcal{E}}} \theta_{ij}^{\vec{t}, \vec{t}} + \sum_{(i,j,t,\overleftarrow{t}) \in \overleftarrow{\mathcal{E}}} \theta_{ij}^{\overleftarrow{t}, \overleftarrow{t}} \\
& \text{subject to} && \\
& \theta_{ij}^{\vec{t}, \vec{t}} \geq d_i^{\vec{t}} + \Delta_i^{\vec{t}} - a_j^{\vec{t}} + \nabla_j^{\vec{t}}, && \forall (i, j, t, \vec{t}) \in \vec{\mathcal{E}} \\
& \theta_{ij}^{\overleftarrow{t}, \overleftarrow{t}} \geq -d_i^{\overleftarrow{t}} - \Delta_i^{\overleftarrow{t}} + a_j^{\overleftarrow{t}} - \nabla_j^{\overleftarrow{t}}, && \forall (i, j, t, \overleftarrow{t}) \in \overleftarrow{\mathcal{E}} \\
& \theta_{ij}^{\vec{t}, \overleftarrow{t}} \geq d_j^{\vec{t}} + \Delta_j^{\vec{t}} - a_i^{\overleftarrow{t}} + \nabla_i^{\overleftarrow{t}}, && \forall (i, j, t, \vec{t}) \in \vec{\mathcal{E}} \\
& \theta_{ij}^{\overleftarrow{t}, \vec{t}} \geq -d_j^{\overleftarrow{t}} - \Delta_j^{\overleftarrow{t}} + a_i^{\vec{t}} - \nabla_i^{\vec{t}}, && (i, j, t, \overleftarrow{t}) \in \overleftarrow{\mathcal{E}} \\
& \text{Equations (3.3) - (3.7), (6.6), (6.7),} && \\
& 0 \leq a_i^{\vec{t}} \leq m, 0 \leq d_i^{\vec{t}} \leq m && \forall t \in \mathcal{T}, \quad \forall i \in \mathcal{N}^t,
\end{aligned} \tag{6.11}$$

where the decision variables are $a_i^{\vec{t}}, d_i^{\vec{t}}, \theta_{ij}^{\vec{t}, \vec{t}}$ and $\theta_{ij}^{\overleftarrow{t}, \overleftarrow{t}}$.

6.4 Limitations of the optimization model

The model presented in this chapter does not suffer from the drawbacks of the robust mixed integer programming model presented in Chapter 4. However, like any optimization model, this model too has some limitations, which stem from the assumptions and approximations we made to construct the model. In this section, we discuss the limitations of the model and possible workarounds when available. The limitations are as follows.

- We have assumed that the amount by which the trip time is allowed to vary is on the order of seconds (Assumption 6.1). Though this is true for most of the subway systems, there are exceptions where this assumption may not hold. For example, when a trip between two cities is considered (especially involving different countries), the trip is on the order of hours with the acceptable trip time bound often being on the order of 5-10 minutes and even more in some cases. In such a scenario, an affine approximation of the energy with respect to trip time would not be very efficient any more, and our model would not be suitable for such a case.
- In Section 6.3 we have applied two different heuristics to arrive at a convex optimization problem. At first we have used $\frac{1}{\epsilon}$ heuristic to come up with a tractable description of the overlapped area between power graphs, and then we have used the ℓ_1 norm heuristic to approximate a nonconvex objective function with its convex envelope. So, it is quite likely that the timetable obtained by solving the convex optimization problem (Problem 6.11) would have a worse objective value compared to the original intractable optimization problem. For this reason our model is energy-efficient, but not necessarily energy-optimal.
- The model does not directly address the case of significant delay. However, we have considered two indirect ways of dealing with it in practice.
 - In any automatic train supervision system, which has the responsibility of implementing the timetable, dwell and velocity regulation are performed to maintain trains on their proper time. If there is a deviation from the optimal timetable because of some delay, the ATS performs regulation to move delayed trains back to the planned optimal timetable. Thus the system will typically return to a normal state in less than half an hour after a delay of one minute.
 - Another way is incorporating the change into the system (due to the delay) as an input data and solving a new but shorter optimization problem with a time horizon of 1-2 hours which can be done in real time using our model. While the shorter model is being implemented we solve the larger optimization problem spanning the rest of the day.

Chapter 7

Numerical experiments for the two-stage LP model

In this chapter we apply our model to different problem instances spanning full service period of one day to service PES2-SFM2 of line 8 of Shanghai Metro network (see Figure 5.1). Detailed description on this railway network has been provided in Section 5.1.

7.1 Shanghai Metro network

The numerical study was executed on a Intel Core i7-46400 CPU with 8 GB RAM running Windows 8.1 Pro operating system. For modelling the problem, we have used JuMP - an open source algebraic modelling language embedded in programming language Julia [32]. Within our JuMP code we have called academic version of Gurobi Optimizer 6.0 as the solver. We have implemented an interior point algorithm because of the underlying sparsity in the data structure. As mentioned before, a measure of the quality of affine fittings using least-squares approach is given by the *coefficient of determination*, which can vary between 0 to 1, with 0 being the worst and 1 being the best [34, page 518]. In our numerical study, the average coefficient of determination of the affine fittings for energy versus trip times over all different trips and all trains is found to be 0.9483 with a standard deviation of 0.05, which justifies our approach.

The duration of the timetables is eighteen hours which is the full service period of the railway network. We have considered eleven different instances with varying average headway times and number of trains. The number of trains increases as the average headway time decreases, where the relation between them can be determined from Equation 3.9. The results of the numerical study are shown in Table 7.1.

We can see that, in all of the cases our model has found the optimal timetables very

quickly, the largest runtime being 12.58s. To the best of our knowledge, this model is the only one to calculate energy-efficient railway timetable spanning an entire day, the next largest being 6 hours only [11] with a much larger computation time for smaller sized problems. After we get the final timetable, we calculate the total *effective energy consumption* by all trains involved in SPSTPs and compare it with the original timetables. The effective energy consumption of a train during a trip is defined as the difference between the total energy required to make a trip and the amount of energy that is being supplied by a braking train during synchronization process. So, the effective energy consumption is the energy that will be consumed from the electrical substations.

The original timetables, which we compare the final timetables with, are provided by Thales Canada Inc. It should be noted that, the number of trains \mathcal{T} is fixed for each of the instances. The energy calculation is done using `SPSIM`, which is a proprietary software owned by Thales Canada Inc [43], and `Cubature`, which is an open-source `Julia` package written by Steven G. Johnson that uses an adaptive algorithm for the approximate calculation of multiple integrals[23]. `SPSIM` calculates the power versus time graphs of all the active trains for the original and optimal timetables. `Cubature` is used to calculate the effective area under the power versus time graphs to determine 1) the total energy required by the trains during the trips, 2) the total transferred regenerative energy during the SPSTPs, and 3) the effective energy consumption as the difference of the first two quantities. The effective energy consumption of the optimal timetables in comparison with the original ones is reduced quite significantly - even in the worst case, the reduction in effective energy consumption is 19.27%, with the best case corresponding to 21.61%.

7.2 Comparison with the robust mixed integer programming model

Now we compare the performance of the two-stage linear model with the robust mixed integer programming model.

1. *Reduction in effective energy consumption.* We can clearly see from Tables 7.1 and 5.1 that the the two-stage model reduces the effective energy consumption more than the robust mixed integer programming model. The mean reductions in energy consumption for the two stage model and the robust mixed integer programming model are 20.47% and 11.2%, respectively. The two-stage model is also more consistent than the robust model in reducing energy consumption. The standard deviation in effective energy reduction is 0.65 for the two-stage model and is 4.72

Table 7.1: Results of the numerical study performed to line 8 of Shanghai Metro network

Instance number	Number of trains	Number of constraints Stage 1	Number of variables Stage 1	Stage 1 CPU time (s)	Number of constraints Stage 2	Number of variables Stage 2	Stage 2 CPU Time (s)	Initial effective energy consumption (kWh)	Final effective energy consumption (kWh)	Reduction in effective energy consumption
1	1000	91998	30060	3.24	116558	34871	6.03	250951.3	201658.7	19.64 %
2	1032	94944	31022	3.03	120394	36038	5.45	261994.5	208558.1	20.40 %
3	1066	98074	32044	3.96	124494	37290	5.62	272486.7	215896.7	20.77 %
4	1100	101204	33066	3.47	129284	38887	5.39	288677.5	229091.8	20.64 %
5	1132	104150	34028	3.15	133354	40171	6.69	308924.4	243672	21.12 %
6	1166	107280	35050	2.84	137322	41357	6.67	322612.7	256288.7	20.56 %
7	1198	110226	36012	2.96	141322	42606	7.16	329388.2	262205.6	20.40%
8	1232	113356	37034	4.04	145756	44025	7.61	354050.2	277536.7	21.61%
9	1266	116486	38056	3.84	149868	45283	8.74	368901.4	297815	19.27 %
10	1298	119432	39018	3.93	153480	46338	7.62	366488.4	293068.8	20.03 %
11	1332	122562	40040	4.22	157752	47676	8.02	379700.8	300910.1	20.75 %

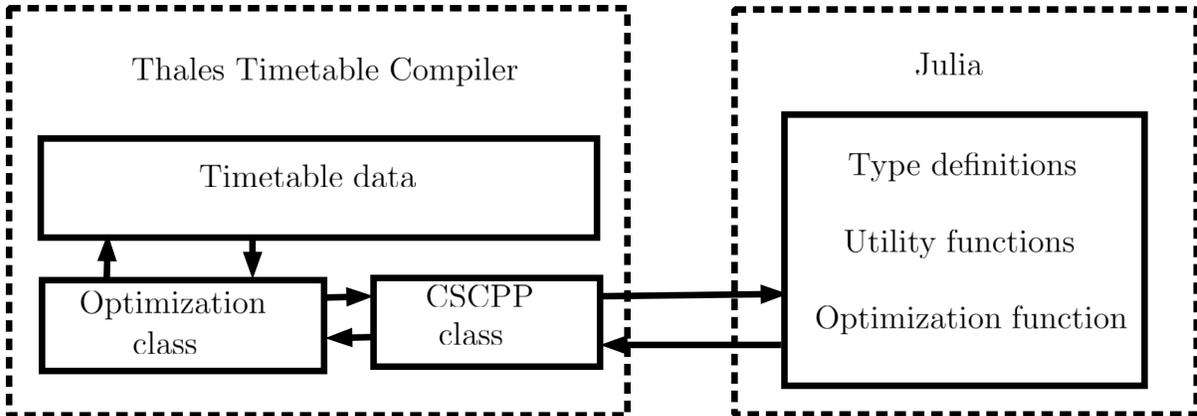


Figure 7.1: The integration architecture of the two stage model with `Thales timetable compiler`

for the robust mixed integer programming model. This significant difference in the performance can be attributed to the difference in modeling. *First*, unlike the two-stage model, the robust mixed integer optimization model does not attempt to reduce the energy consumed by the trains. *Second*, the regenerative energy saving strategy is more realistic in the two-stage model. The numerical experiments support that the strategy to maximize the overlapping time between the temporal blocks may not result in the the transfer of maximum amount regenerative energy.

2. *CPU time*. From Tables 5.1, 5.2 and 7.1, we can see that in terms of CPU time the two-stage model is significantly faster than mixed integer programming model. In fact, for the same railway network the two stage model calculates 18 hours timetables in far less CPU time (max CPU time less than 13s) than the mixed integer programming model which calculates timetables spanning only 6 hours (max CPU time 86.64s). This comes as no surprise, as the two stage model is a linear program having polynomial time complexity, whereas the mixed integer programming model is of \mathcal{NP} -hard complexity.

From the above discussions, we arrive at the conclusion that the two-stage optimization model is significantly better than the robust mixed integer optimization model.

7.3 Integration with Thales timetable compiler

In this section we describe how the codes for the two-stage model has been integrated with `Thales timetable compiler`. The overall integration architecture is illustrated in Figure 7.1. The descriptions of various blocks in the figure are given below.

- `Thales timetable compiler`
 - *Timetable data*. This block contains the required data for solving the optimization problem. The data structure in this block is based on tracks, and holds more data than necessary. The data structure in the Julia code is based on suitable pairs and platforms.
 - *Optimization class*. The optimization class converts the data structure in the timetable data block in a form compatible for the optimization algorithm.
 - *CSCPP class*. The CSCPP class maintains the communication between the Thales timetable compiler and the Julia code. It inputs the data into the Julia code and collects the optimized timetable.

- `Julia code`
 - *Type definitions*. This part defines the data structures necessary to describe a railway network.
 - *Utility functions*. This part contains necessary functions that are required to run the optimization algorithm, and maintains the communication between the two stages of the optimization model.
 - *Optimization function*. This part calculates the energy efficient timetable and sends the result to the CSCPP class.

Chapter 8

Conclusion and future works

In this thesis, we have proposed two optimization models to calculate energy-efficient timetables in railway networks. The first optimization model is a robust mixed integer optimization model. It provides accurate and relatively tractable modeling of the optimization problem in order to maximize the total overlapping time between suitable train pairs. The second optimization model is proposed to overcome the limitations of the first model. Unlike the first model and other relevant works, the second model 1) is more realistic in addressing the problem of increasing energy-efficiency of railway timetables, and yet 2) has a polynomial time computational complexity. In practice, the second model has proven to be very efficient in calculating energy-efficient timetables and has been integrated with an industrial railway timetable compiler.

8.1 Conclusion

After providing the necessary background for the thesis in Chapter 2, we modeled all the constraints needed by a feasible railway timetable in Chapter 3, including safety regulations, service levels and restrictions that consider the operational feasibility of the railway management. The constraints form the feasible set for the optimization models described in the later chapters.

In Chapter 4 we presented a robust mixed integer optimization model to utilize regenerative braking energy produced by trains in a railway network. Using this model saves regenerative energy of braking trains by transferring it to suitable accelerating trains in need of energy. We modeled the objective function of the optimization problem using hypograph approach and interval algebra.

Then in Chapter 5, we applied the optimization model presented in Chapter 4 to different instances of two railway networks (Shanghai Metro Network and Dockland Light

Railway) for time horizon spanning six hours. Compared to the original timetables, the overlapping time increases significantly in the optimal timetables. In the first railway network we have access to relevant energy information to calculate the relative reduction in effective energy consumption and we find that there is significant increase in utilization of regenerative energy for every instance compared to the existing timetables.

However, the robust mixed integer programming model has some limitations associated with \mathcal{NP} -hard computational complexity, maximization in overlapping time not being equal to maximization in regenerative energy transfer and not considering minimizing the energy expenditure by moving trains. To overcome the limitations of the first model, in Chapter 6 we proposed a novel two-stage linear optimization model to calculate energy-efficient timetables in electric railway networks. Both stages of this optimization model are linear programs, hence solvable in polynomial time.

Application of the two stage model to different instances of a real life railway network spanning a full service period of 18 hours with thousands of active trains shows that the model finds an optimal timetable very quickly with significant reduction in effective energy consumption. A comparison between the two models indicate that the two-stage optimization model is significantly better than the robust mixed integer optimization model.

8.2 Future works

Now we discuss possible future works.

- One of the assumptions of the two stage model is that the amount by which the trip time is allowed to vary is on the order of seconds. There are exceptions where this assumption may not hold. For example, when a trip between two cities is considered, an affine approximation of the energy with respect to trip time is not very efficient. In such a case, a non-affine but polynomial approximation of the energy can still be found. Replacing the affine energy functions with polynomials will convert the first-stage of the optimization problem into a polynomial optimization problem. Solving such polynomial optimization problems can be of significant interest.
- In the second stage of the optimization model we used $\frac{1}{\epsilon}$ heuristic to come up with a tractable description of the overlapped area between power graphs, and then we used the ℓ_1 norm heuristic to approximate a nonconvex objective function with its convex envelope. A natural research direction is to investigate how to model

the optimization problem without these two layers of approximations, which would provide an energy-optimal timetable, rather than an energy-efficient one.

- When a significant delay occurs in the railway network, the models provide indirect ways of dealing with it without any direct resolution. An interesting research direction can be how to model a delay resilient optimization model to calculate energy-efficient timetables.

Appendix

Speed limit for the railway network considered

Table 8.1: Speed limit for line 8 of Shanghai Metro network

Origin-Destination	Start (m)	End (m)	Speed limit (km/h)
CSR1-YSS1	0.0	143.5	60
	143.5	1004.6	70
	1004.6	1138.2	60
CSR2-YHR2	0.0	910.0	60
GRW1-LXM1	0.0	153.3	60
	153.3	870.1	70
	870.1	1006.9	60
GRW2-PES2	0.0	173.1	60
	173.1	636.4	70
	636.4	769.5	60
JYR1-LZV1	0.0	1366.7	60
	1366.7	2220.6	65
	2220.6	2357.3	60
JYR2-YSS2	0.0	143.4	60
	143.4	1388.9	70
	1388.9	1522.3	60
JYS1-LHS1	0.0	140.0	60
	140.0	829.2	75
	829.2	1202.3	60
JYS2-PJT2	0.0	140.0	60

	140.0	371.1	70
	371.1	1081.9	75
	1081.9	1249.8	70
	1249.8	1386.2	60
LHR1-PJT1	0.0	140.1	60
	140.1	766.2	70
	766.2	1623.4	75
	1623.4	1805.9	70
	1805.9	2374.4	75
	2374.4	2487.8	70
	2487.8	2624.3	60
LHR2-LZV2	0.0	139.8	60
	139.8	2457.3	70
	2457.3	2594.1	60
LHS1-SFM1	0.0	140.0	60
	140.0	1220.1	70
	1220.1	1357.4	60
LHS2-JYS2	0.0	186.7	60
	186.7	853.8	75
	853.8	1064.4	70
	1064.4	1200.8	60
LJB1-SXZ1	0.0	140.1	60
	140.1	1027.4	70
	1027.4	1167.2	60
LJB2-LXM2	0.0	140.1	60
	140.1	693.3	70
	693.3	830.2	60
LXM1-LJB1	0.0	140.0	60
	140.0	689.3	70
	689.3	826.1	60
LXM2-GRW2	0.0	140.0	60
	140.0	855.5	70
	855.5	1005.2	60

LZV1-LHR1	0.0	140.1	60
	140.1	1901.1	70
	1901.1	2199.1	75
	2199.1	2456.3	70
	2456.3	2592.8	60
LZV2-JYR2	0.0	143.3	60
	143.3	1007.8	65
	1007.8	2338.1	60
PES1-GRW1	0.0	140.0	60
	140.0	561.7	70
	561.7	761.5	60
PJT1-JYS1	0.0	143.3	60
	143.3	374.6	70
	374.6	1089.8	75
	1089.8	1250.2	70
	1250.2	1386.7	60
PJT2-LHR2	0.0	143.4	60
	143.4	355.8	70
	355.8	829.3	75
	829.3	1039.2	70
	1039.2	1858.3	75
	1858.3	2488.9	70
	2488.9	2622.1	60
SFM2-LHS2	0.0	140.3	60
	140.3	373.0	70
	373.0	742.3	75
	742.3	1225.1	70
	1225.1	1358.3	60
SXZ1-ZJD1	0.0	140.1	60
	140.1	647.2	65
	647.2	1699.3	70
	1699.3	2039.3	60
SXZ2-LJB2	0.0	160.1	60

	160.1	1027.5	70
	1027.5	1164.2	60
YHR1-CSR1	0.0	143.4	60
	143.4	773.7	70
	773.7	910.3	60
YHR2-ZJD2	0.0	140.0	60
	140.0	601.3	70
	601.3	738.0	60
YSS1-JYR1	0.0	140.2	60
	140.2	664.7	70
	664.7	987.7	75
	987.7	1389.2	70
	1389.2	1525.9	60
YSS2-CSR2	0.0	430.4	60
	430.4	1014.8	70
	1014.8	1151.6	54
ZJD1-YHR1	0.0	140.1	60
	140.1	605.9	70
	605.9	742.5	60
ZJD2-SXZ2	0.0	353.8	60
	353.8	1393.7	70
	1393.7	1910.0	65
	1910.0	2043.3	60

Bibliography

- [1] *UNESCO science report: towards 2030*. UNESCO Publishing, 2015.
- [2] M Abril, F Barber, L Ingolotti, MA Salido, P Tormos, and A Lova. An assessment of railway capacity. *Transportation Research Part E: Logistics and Transportation Review*, 44(5):774–806, 2008.
- [3] James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
- [4] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.
- [5] K.R.V. Subramanian Avinash Balakrishnan, editor. *Nanostructured Ceramic Oxides for Supercapacitor Applications*. CRC Press, 2014.
- [6] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.
- [7] Dimitris Bertsimas and Robert Weismantel. *Optimization over integers*, volume 13. Dynamic Ideas Belmont, 2005.
- [8] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2009.
- [9] Giuseppe Calafiore and Laurent El Ghaoui. *Optimization Models*. Cambridge University Press, 2014.
- [10] Flavio Ciccarelli, Andrea Del Pizzo, and Diego Iannuzzi. Improvement of energy efficiency in light railway vehicles based on power management control of wayside lithium-ion capacitor storage. *IEEE Transactions on Power Electronics*, 29(1):275–286, 2014.

- [11] Shuvomoy Das Gupta, Lăcră Pavel, and James Kevin Tobin. An optimization model to utilize regenerative braking energy in a railway network. In *American Control Conference (ACC)*, 2015.
- [12] Oxford English Dictionary. Oxford: Oxford university press, 2004.
- [13] Bert Droste-Franke, Boris Paal, Christian Rehtanz, Dirk Uwe Sauer, Jens-Peter Schneider, Miranda Schreurs, and Thomas Ziesemer. *Balancing Renewable Electricity: Energy Storage, Demand Side Management, and Network Extension from an Interdisciplinary Perspective*, volume 40. Springer Science & Business Media, 2012.
- [14] George Gamow. *Thirty years that shook physics: The story of quantum theory*. Courier Corporation, 1966.
- [15] Arturo González-Gil, Roberto Palacin, and Paul Batty. Sustainable urban rail systems: Strategies and technologies for optimal management of regenerative braking energy. *Energy conversion and management*, 75:374–388, 2013.
- [16] Shanghai Shentong Metro Group. Description of line 8, shanghai metro, 2015.
- [17] Steven S Harrod. A tutorial on fundamental model structures for railway timetable optimization. *Surveys in Operations Research and Management Science*, 17(2):85–96, 2012.
- [18] Phil Howlett. The optimal control of a train. *Annals of Operations Research*, 98(1-4):65–87, 2000.
- [19] Phil G Howlett, Peter J Pudney, and Xuan Vu. Local energy minimization in optimal train control. *Automatica*, 45(11):2692–2698, 2009.
- [20] Philip G Howlett and Peter J Pudney. *Energy-efficient train control*. Springer Science & Business Media, 1995.
- [21] IBM. Ilog cplex optimizer 12.6.
- [22] Cheng Jiabin and Phil Howlett. A note on the calculation of optimal strategies for the minimization of fuel consumption in the control of trains. *IEEE Transactions on Automatic Control*, 38(11):1730–1734, 1993.
- [23] Steven G. Johnson. The cubature module for julia.
- [24] Asha Seth Kapadia, Wenyaw Chan, and Lemuel A Moyé. *Mathematical statistics with applications*. CRC Press, 2005.

- [25] Eugene Khmelnitsky. On an optimal control problem of train operation. *IEEE Transactions on Automatic Control*, 45(7):1257–1266, 2000.
- [26] Zhao Le, Keping Li, Jingjing Ye, and Xiaoming Xu. Optimizing the train timetable for a subway system. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 2014.
- [27] Xiang Li and Hong K Lo. An energy-efficient scheduling and speed control approach for metro rail operations. *Transportation Research Part B: Methodological*, 64:73–89, 2014.
- [28] Xiang Li and Hong K Lo. Energy minimization in dynamic train scheduling and control for metro rail operations. *Transportation Research Part B: Methodological*, 70:269–284, 2014.
- [29] Christian Liebchen. *Periodic timetable optimization in public transport*. Springer, 2007.
- [30] Thomas Lindner. Train schedule optimization in public rail transport. *Mathematics Key Technology for the Future: Joint Projects Between Universities and Industry*, pages 703–716, 2000.
- [31] Rongfang Rachel Liu and Iakov M Golovitcher. Energy-efficient operation of rail vehicles. *Transportation Research Part A: Policy and Practice*, 37(10):917–932, 2003.
- [32] Miles Lubin and Iain Dunning. Computing in operations research using julia. *INFORMS Journal on Computing*, 27(2):238–248, 2015.
- [33] Sanjoy Mahajan. *Street-fighting Mathematics*. MIT Press Cambridge, 2010.
- [34] William Mendenhall, Robert Beaver, and Barbara Beaver. *Introduction to probability and statistics*. Cengage Learning, 2009.
- [35] Ian P Milroy. *Aspects of automatic train control, PhD Thesis*. PhD thesis, Loughborough University of Technology, 1980.
- [36] Leon Peeters. *Cyclic railway timetable optimization, PhD Thesis*. PhD thesis, Erasmus University Rotterdam, 2003.
- [37] Maite Peña-Alcaraz, Antonio Fernández, Asuncion Paloma Cucala, Andres Ramos, and Ramon R Pecharromán. Optimal underground timetable design based on power flow for maximizing the use of regenerative-braking energy. *Proceedings of the*

- Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 226(4):397–408, 2012.
- [38] Maite Peña-Alcaraz, Mort Webster, and Andrés Ramos. An approximate dynamic programming approach for designing train timetables. *ESD Working Paper Series*, 2012.
- [39] Eduardo Pilo. *Power supply, energy management and catenary problems*. WIT Press, 2010.
- [40] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
- [41] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2012.
- [42] S Sivanagaraju. *Power system operation and control*. Pearson Education India, 2009.
- [43] Emil P Vlad and Valeriy Tatarnikov. R&M&A&S of communication based train control systems applied to urban rail transportation: a way to improve city sustainability. In *Proceedings of IEEE Reliability and Maintainability Symposium (RAMS)*, pages 1–6, 2011.
- [44] Wikipedia. List of current systems for electric rail traction. 2015.
- [45] H Paul Williams. *Model building in mathematical programming*. John Wiley & Sons, 2013.
- [46] Xin Yang, Xiang Li, Ziyou Gao, Hongwei Wang, and Tao Tang. A cooperative scheduling model for timetable optimization in subway systems. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):438–447, 2013.