

**DISTRIBUTED AND LEARNING-BASED MODEL
PREDICTIVE CONTROL FOR URBAN RAIL TRANSIT
NETWORKS**

DISTRIBUTED AND LEARNING-BASED MODEL PREDICTIVE CONTROL FOR URBAN RAIL TRANSIT NETWORKS

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus, prof. dr. ir. T.H.J.J. van der Hagen
chair of the Board of Doctorates,
to be defended publicly
on Wednesday, October 23, 2024 at 10:00 AM

by

Xiaoyu LIU

Master of Science in Traffic Information Engineering and Control,
Beijing Jiaotong University, Beijing, China,
born in Linyi, Shandong, China.

This dissertation has been approved by the promotor.

Promotor: Prof. dr. ir. B. De Schutter

Copromotor: Dr. A. Dabiri

Composition of the doctoral committee:

Rector Magnificus	chairperson
Prof. dr. ir. B. De Schutter	Delft University of Technology, promotor
Dr. A. Dabiri	Delft University of Technology, copromotor

Independent members:

Prof. dr. R.M.P. Goverde	Delft University of Technology
Prof. dr. ir. T. Keviczky	Delft University of Technology
Prof. dr. D. Huisman	Erasmus University Rotterdam
Prof. dr. D. Pacciarelli	Roma Tre University, Italy
Prof. dr. D. V. Dimarogonas	KTH Royal Institute of Technology, Sweden



This dissertation has been completed in fulfillment of the requirements of TU Delft Graduate School for the Doctoral Education Program and the Dutch Institute of Systems and Control (DISC) for graduate studies. The support of the Chinese Scholarship Council (CSC) is greatly acknowledged.

Cover designed by: X. Liu

Published and distributed by: X. Liu

E-mail: xiaoyuliu.lab@gmail.com

Keywords: Urban Rail Transit Network, Model Predictive Control, Distributed Control, Learning-based Control.

Copyright © 2024 by X. Liu

ISBN/EAN: 978-90-5584-350-3 (Paperback)

ISBN/EAN: 978-94-6384-659-2 (E-book)

An electronic version of this dissertation is available at
<http://repository.tudelft.nl/>.

Knowledge and Action

CONTENTS

Preface	xi
Summary	xiii
Samenvatting	xv
1 Introduction	1
1.1 Background	1
1.2 Motivation and challenges	3
1.3 Research questions	4
1.4 Contributions	5
1.5 Thesis outline	6
2 Integration of timetables, passenger flows, and train speed profiles	9
2.1 Introduction	10
2.2 State of the art	11
2.2.1 Passenger-oriented real-time timetable scheduling	11
2.2.2 Passenger-oriented train departure frequency optimization	12
2.2.3 MPC for real-time railway train scheduling	13
2.3 Problem statement and assumptions	16
2.3.1 Problem statement.	16
2.3.2 Explanations and assumptions.	17
2.4 Mathematical model	17
2.4.1 Notations	17
2.4.2 Passenger absorption model	18
2.4.3 Train scheduling model	24
2.5 Bi-level MPC for train scheduling	27
2.5.1 Bi-level MPC for the integrated problem	27
2.5.2 Higher-level MPC: departure frequency optimization	28
2.5.3 Lower-level MPC: train scheduling.	31
2.6 Case study	33
2.6.1 Basic setup.	34
2.6.2 Assessment of the absorption model.	36
2.6.3 Bi-level optimization based on the absorption model	38
2.6.4 Bi-level MPC for real-time train scheduling	39
2.7 Conclusions.	42

3	Train departure frequency optimization with uncertain passenger flows	43
3.1	Introduction	44
3.1.1	Passenger-oriented train scheduling for a single line.	44
3.1.2	Passenger-oriented train scheduling for networks	44
3.1.3	MPC for real-time railway traffic management.	45
3.1.4	Train scheduling under uncertainties	46
3.1.5	Chapter contributions and structure.	46
3.2	Mathematical model	47
3.2.1	Notations	47
3.2.2	Passenger absorption model	47
3.2.3	Constraints for the absorption model	50
3.3	Distributed knowledgeable-reduced-horizon algorithm for train schedul- ing	51
3.3.1	Problem formulation in MPC set-up	51
3.3.2	Knowledgeable-reduced-horizon algorithm for real-time train schedul- ing	52
3.3.3	Distributed KRH algorithm for train scheduling in urban rail transit network	54
3.4	Scenario-based DKRH algorithm	56
3.5	Case study	58
3.5.1	Setup.	59
3.5.2	Real-time train scheduling for the deterministic case	61
3.5.3	Real-time train scheduling with uncertain passenger flows	63
3.6	Conclusions.	65
4	MPC for passenger-oriented real-time train rescheduling	67
4.1	Introduction	68
4.2	State of the art	69
4.2.1	Models for timetable scheduling.	69
4.2.2	Passenger-oriented timetable scheduling	70
4.2.3	MPC for railway traffic management.	71
4.3	Passenger-oriented real-time timetable scheduling model	71
4.3.1	Notations	72
4.3.2	Simplified passenger flow model.	72
4.3.3	Train operation model	77
4.4	MPC for passenger-oriented timetable rescheduling	78
4.5	Solution approaches	80
4.5.1	Sequential quadratic programming approach	80
4.5.2	Mixed-integer linear programming approach	81
4.5.3	Simplified mixed-integer linear programming approach.	82
4.6	Case study	83
4.6.1	Assessment of the proposed model	83
4.6.2	Open-loop optimization based on the proposed model	86
4.6.3	Closed-loop control for real-time timetable scheduling	88
4.7	Conclusions.	91

5	Learning-based MPC for train rescheduling with flexible train composition	93
5.1	Introduction	94
5.1.1	Passenger-oriented train scheduling	94
5.1.2	MPC for real-time train scheduling	95
5.1.3	Learning-based train scheduling	96
5.1.4	Contributions and structure of the chapter	96
5.2	Problem description and explanations	97
5.3	Mathematical formulation for passenger-oriented train scheduling	98
5.3.1	Notations	98
5.3.2	Train operation constraints	98
5.3.3	Rolling stock circulation constraints	101
5.3.4	Passenger flow constraints	102
5.4	MPC for real-time train rescheduling	104
5.4.1	Problem formulation	104
5.4.2	MINLP-based MPC for real-time train rescheduling	105
5.4.3	MILP-based MPC for Real-Time Train Rescheduling	107
5.5	Learning-based MPC for real-time train rescheduling	107
5.5.1	Presolve techniques	108
5.5.2	Environment setting	109
5.5.3	Offline training for learning-based MPC algorithm	109
5.6	Case study	110
5.6.1	Basic Setting	110
5.6.2	Simulation and Results	112
5.7	Conclusions	115
6	Cooperative distributed MPC for virtually coupled trains	117
6.1	Introduction	118
6.2	Problem statement and preliminaries	121
6.2.1	Problem statement	121
6.2.2	Preliminaries	123
6.3	Mathematical model for virtually coupled trains	123
6.3.1	Train dynamic model	123
6.3.2	Dynamic model for virtually coupled trains	125
6.4	Distributed model predictive control for virtually coupled trains	125
6.4.1	General nonlinear MPC problem formulation	126
6.4.2	Nonconvex cooperative distributed MPC	127
6.4.3	Convex cooperative distributed MPC	129
6.4.4	Nonconvex serial distributed MPC	130
6.4.5	Convex serial distributed MPC	131
6.4.6	Nonconvex decentralized MPC	131
6.4.7	Convex decentralized MPC	132
6.5	Case study	132
6.5.1	General setup	132
6.5.2	Control performance with uniform train masses	134
6.5.3	Control performance with heterogeneous train mass	137
6.5.4	Highlights of results	140

6.6	Conclusions.	143
7	Conclusions and recommendations	145
7.1	Conclusions.	145
7.2	Impacts of this thesis	147
7.2.1	Social impacts	147
7.2.2	Scientific and technical impacts	148
7.3	Recommendations for future research	149
7.3.1	Recommendations for application topics	149
7.3.2	Recommendations for theory topics	149
	Bibliography	151
	Curriculum vitae	165
	List of publications	167
	TRAIL Thesis Series	169

PREFACE

Although four years may seem like a long time, my PhD journey is now truly coming to an end. As I look back, I realize these years have been filled with unforgettable and invaluable experiences. I would like to take this opportunity to express my gratitude to all who have offered me help and support along the way.

First, I would like to express my sincere appreciation to my supervisors, Prof. Bart De Schutter and Dr. Azita Dabiri. Thank you Bart for accepting me as your student, you have embodied everything I expected as an excellent supervisor. Your scientific thinking, rigorous research attitude, and unwavering assistance have been a constant source of inspiration, motivating me to become a better researcher. I truly cherish our weekly meetings, and even during your holidays, you always provided timely feedback and guidance. I will never forget the drafts that are full of your special handwriting. Thank you Azita for always being there to provide timely comments and suggestions on my research. Your insightful feedback ensured that my research stayed on the right track. The nice discussions in our biweekly joint meetings will always be in my memories.

I would also like to thank Prof. Dimos V. Dimarogonas for offering me the wonderful opportunity to conduct my research visit at KTH Royal Institute of Technology in Stockholm, Sweden. The discussions and your insightful suggestions during our meetings significantly broadened my research perspectives. My gratitude also extends to the friends I made at the Division of Decision and Control Systems in KTH for creating a warmly welcoming atmosphere and for the valuable discussions we shared.

I would like to express my gratitude to Prof. Rob Goverde for the opportunity to attend the bi-weekly meetings in your group, which significantly enhanced my understanding of railway operations. I am also thankful to the friends I made at Transport & Planning, especially in Digital Railway Traffic Lab, for their warm atmosphere and for engaging in enlightening discussions.

I want to express my appreciation to Prof. Jing Xun, the supervisor of my master's studies, for being supportive in both life and studies since my first year at the university. Thank you for encouraging me when I lacked confidence and for providing valuable support during the past eleven years. I would also like to thank Dr. Yihui Wang for her insightful suggestions and valuable help, both academically and personally. Deep gratitude also goes to our supervisor, Prof. Bin Ning, for his guidance and for bringing us into the same lab at Beijing Jiaotong University. The discussions, assistance, and suggestions from both Prof. Xun and Dr. Wang for my PhD research are also greatly appreciated.

Thanks to all my friends and colleagues at the Delft Center for Systems and Control. Thanks to Dingshan Sun, Yun Li, Kanghui He, Changrui Liu, Luyao Zhang, Yang Wang, Zhixin Feng, Ying Ma, Yixuan Liu, Diyou Liu, Shijie Huang, and Ruiyuan Li for your companionship and help. The lunches, dinners, parties, and trips we had during the past years will forever stay in my memories. Thanks to Dr. Meichen Guo for her valuable help and discussions. Thanks to Caio Fabio Oliveira da Silva for his significant efforts

and cooperation in our collaborative work. Maarten de Jong, Ioannis Panagopoulos, Filippo Airaldi, Emilio Benenati, Leila Gharavi, Athina Ilioudi, Alessandro Riccardi, Gianpietro Battocletti, Francesco Cordiano, Sam Mallick, Tolga Ok, Pedro Zattoni Scroccaro, Frederik Mathiesen, Giray Önür, and Roger Moens have shared excellent discussions and valuable times. I also would like to express my appreciation to the people who have left Delft: Dr. Maolong Lyu, Dr. Yongxia Shi, Dr. Jianfeng Fu, Dr. Ximan Wang, Dr. Tian Tao, Dr. Jingwei Dong, and Dr. Shengling Shi for the valuable suggestions and help. I truly cherish the time we spent together. Thanks to Berend Wentges, Alexander Daman, and Hengkai Zhang for working on their master's theses with me and for their valuable cooperation and engagement. Special thanks to our secretaries for their kind support. Thanks to the beach volleyball team for sharing a lot of exciting games. The same appreciation also goes to the friends in Transport & Planning: Dr. Egidio Quaglietta, Dr. Yongqiu Zhu, Dr. Joelle Aoun, Dr. Marko Kapetanović, Dr. Yimeng Zhang, Ziyulong Wang, Yuqing Ji, Weining Hao, Chao Yu, Tianqi Lu, Yiru Jiao, Alex Cunillera, Renate van der Knaap, Konstantinos Rigos, and Nina Versluis. Thank you!

I also want to express my gratitude to Cheng Chang, Desong Du, Tianlong Jia, Zichao Li, Yigu Liu, Dinghao Wu, Zhenzhen Wu, and Sen Yuan for the enjoyable times we shared and for your companionship, especially during the difficult period when none of us could return home in the first two years in the Netherlands. I still miss the happy times we spent together and the moments we supported each other through tough times. I am also grateful to Yifei Li, Jingyi Liu, Xiaohuan Lyu, Yujie Tang, Kai Wu, and Xinxin Zhang. I feel lucky to have met you at the start of my PhD and to have shared unforgettable moments during the past four years.

I would like to thank my parents and my sister for their unconditional love and support. No words can fully convey my love and gratitude to my parents. Last but not least, thanks to my fiancée Dr. Jiao Zhao for her endless love and unwavering support. Your love, understanding, and encouragement always bring warmth and ease to my life.

*Xiaoyu Liu
Delft, October 2024*

SUMMARY

Distributed and Learning-based Model Predictive Control for Urban Rail Transit Networks

Model predictive control (MPC) is an efficient optimization-based control methodology for real-time control of constrained systems. Urban rail transit networks are typical examples of constrained systems dedicated to providing safe, efficient, and eco-friendly services within infrastructure limitations. This thesis focuses on innovative MPC strategies for traffic management in urban rail transit networks, integrating three key elements in traffic management, i.e., passenger flows, timetables, and train speeds. The thesis includes four topics, where in the first topic, we develop an integrated model and its corresponding control approach. The second topic focuses on scenario-based distributed MPC intending to determine train departure frequency based on passenger flows in urban rail transit networks. The third topic includes efficient centralized and learning-based MPC for timetable rescheduling according to passenger demands. The fourth topic investigates cooperative distributed MPC for the speed control of trains.

In the first topic, we investigate the integration of passenger flows, timetables, and train speeds. A passenger absorption model that explicitly includes time-dependent origin–destination demands is developed, where the term “absorption” refers to passengers boarding trains. Then, the passenger absorption model is extended to a bi-level model, where passenger demands and rolling stock availability are considered at the higher level, and detailed timetables and train speed profiles are included at the lower level. A bi-level MPC approach is developed for the integrated problem. We show that the optimization problems of both levels of the bi-level MPC approach can be converted into mixed-integer linear programming (MILP) problems, which enables us to solve them with existing MILP solvers. In this way, we can achieve real-time train scheduling for urban rail transit networks. Simulation results show that the bi-level MPC approach outperforms the centralized MPC approach.

The second topic of the thesis focuses on train departure frequency optimization for urban rail transit networks while also considering uncertain time-dependent origin–destination demands. A distributed MPC approach is developed to deal with the computational burden and the communication restrictions of the train scheduling problem in urban rail transit networks. For the distributed MPC approach, a cost-to-go function is designed to reduce the prediction horizon of the original MPC approach while taking into account the control performance. By applying a scenario reduction approach, a scenario-based distributed MPC approach is proposed to handle the uncertain passenger flows with an acceptable increase in computation time. The simulation results indicate that distributed MPC can be used to achieve real-time train scheduling for the urban rail transit network, while scenario-based distributed MPC can handle the uncer-

tainty in the passenger flows with an acceptable sacrifice in computation time.

The third topic of the thesis focuses on efficient centralized MPC approaches for passenger-oriented real-time timetable rescheduling on an urban rail transit line. At each platform, we discretize the planning time window into several time intervals of equal length, where every time interval includes one departure of a train at the same platform. We adjust the train departure within the time interval to improve passenger satisfaction while ensuring regular departures. The resulting nonlinear non-convex MPC optimization problem is reformulated into an MILP problem that can be solved very efficiently by existing MILP solvers. The mixed-integer programming problem typically has computational issues due to the problem scale, and the computational complexity is significantly influenced by the number of integer variables. Therefore, we further develop a learning-based MPC approach, where the integer variables are obtained by deep learning, and then the MPC optimizer only needs to solve a continuous nonlinear optimization problem with fewer variables than the original problem at each time step. Simulation results show that compared to the original MPC approach, the learning-based MPC approach significantly reduces the computational time while achieving comparable performance.

In the fourth topic, we investigate the speed control of virtually coupled trains, which is crucial for trains to adhere to a timetable. Virtual coupling is regarded as an efficient way to improve the line capacity of rail transportation systems by reducing the spacing between consecutive trains. In real life, masses of trains are different and can change at stations due to changes in passenger loads, which influence the dynamics and control of the virtually coupled trains. Taking into account the nonlinear train model and changes in masses of trains, cooperative distributed MPC, serial distributed MPC, and decentralized MPC are compared and assessed for controlling virtually coupled trains. To make a balanced trade-off between computational complexity and efficiency, we also propose and assess convex approximations of the above control approaches. We introduce relaxed dynamic programming into the train control field, and a distributed stopping criterion with a stability guarantee has been developed for the cooperative distributed MPC approach. Simulation results indicate that the cooperative distributed MPC approach has the best tracking performance, while the serial distributed MPC approach can reduce communication requirements and computation capabilities with sacrifices of tracking performance.

In summary, this thesis addresses the integration of passenger flows, timetables, and train speeds by developing several MPC frameworks, including bi-level MPC, scenario-based distributed MPC, learning-based MPC, and cooperative distributed MPC. These approaches are applied to traffic management of urban rail transit networks and yield improved performance compared to conventional methods.

SAMENVATTING

Gedistribueerde en Leer-gebaseerde Modelvoorspellende Besturing van Stedelijke Spoorwegnetwerken

Modelvoorspellende besturing (in het Engels: *Model Predictive Control*, MPC) is een efficiënte optimalisatie-gebaseerde regelmethodologie voor real-time besturing van systemen met beperkingen. Stedelijke spoorwegnetwerken zijn typische voorbeelden van dergelijke systemen, die zijn ontworpen om veilige, efficiënte en milieuvriendelijke diensten te leveren binnen de beperkingen van de infrastructuur. Dit proefschrift richt zich op innovatieve MPC-strategieën voor verkeersmanagement in stedelijke spoorwegnetwerken, waarbij drie belangrijke elementen van verkeersbeheer worden geïntegreerd, namelijk passagiersstromen, dienstregelingen en treinsnelheden. Het proefschrift omvat vier onderwerpen. In het eerste onderwerp ontwikkelen we een geïntegreerd model en de bijbehorende regelmethode. Het tweede onderwerp richt zich op scenario-gebaseerde gedistribueerde MPC met als doel de vertrekfrequentie van treinen te bepalen op basis van passagiersstromen. Het derde onderwerp omvat efficiënte gecentraliseerde en op leren gebaseerde MPC voor het herschikken van dienstregelingen op basis van passagiersbehoeften. Het vierde onderwerp onderzoekt coöperatieve gedistribueerde MPC voor de snelheidsregeling van virtuele gekoppelde treinen.

In het eerste onderwerp onderzoeken we de integratie van passagiersstromen, dienstregelingen en treinsnelheden. We ontwikkelen een passagiersabsorptiemodel dat expliciet de tijdafhankelijke herkomst-bestemmingsvraag omvat, waarbij de term “absorptie” verwijst naar passagiers die in treinen stappen. Vervolgens wordt het passagiersabsorptiemodel uitgebreid naar een tweeledig model, waarbij de passagiersvraag en de beschikbaarheid van rollend materieel op een hoger niveau worden beschouwd, en gedetailleerde dienstregelingen en treinsnelheidsprofielen worden bepaald op een lager niveau. Een tweeledige MPC-aanpak wordt ontwikkeld voor het geïntegreerde probleem. We tonen aan dat de optimaliseringsproblemen van beide niveaus van de tweeledige MPC-aanpak kunnen worden omgezet in gemeng-geheeltallige lineaire programmeringsproblemen (in het Engels: *Mixed-Integer Linear Programming*, MILP), wat ons in staat stelt ze op te lossen met bestaande MILP-oplossers. Op deze manier kunnen we real-time treinplanning bereiken voor stedelijke railtransitnetwerken. Simulatieresultaten tonen aan dat de tweeledige MPC-aanpak beter presteert dan de gecentraliseerde MPC-aanpak.

Het tweede onderwerp van het proefschrift richt zich op de optimalisatie van de vertrekfrequentie van treinen in stedelijke spoorwegnetwerken, waarbij ook rekening wordt gehouden met de onzekere tijdsafhankelijke oorsprong-bestemming svraag. Een gedistribueerde MPC-aanpak wordt ontwikkeld om de rekenlast en de communicatiebeperkingen van het treinplanningsprobleem in stedelijke spoorwegnetwerken aan te pak-

ken. Voor de gedistribueerde MPC-aanpak wordt een *cost-to-go* functie ontworpen om de voorspellingshorizon van de oorspronkelijke MPC-aanpak te verkorten, terwijl de regelprestaties worden meegenomen. Door een scenarioreductie-aanpak toe te passen, wordt een scenario gebaseerde gedistribueerde MPC-aanpak voorgesteld om de onzekere passagiersstromen te beheersen met een aanvaardbare toename van de rekestijd. De simulatieresultaten geven aan dat gedistribueerde MPC kan worden gebruikt voor real-time treinplanning in stedelijke spoorwegnetwerken, terwijl scenario gebaseerde gedistribueerde MPC de onzekerheid in passagiersstromen kan verwerken met een aanvaardbare in rekestijd.

Het derde onderwerp van het proefschrift richt zich op efficiënte gecentraliseerde MPC-aanpakken voor passagiersgerichte real-time herschikking van dienstregelingen op een stedelijke spoorlijn. Op elk perron discretiseren we het planningsvenster in verschillende tijdsintervallen van gelijke lengte, waarbij elk tijdsinterval één vertrek van een trein op hetzelfde perron omvat. We passen het treinvertrek binnen het tijdsinterval aan om de passagierstevredenheid te verbeteren, terwijl regelmatige vertrekken worden gewaarborgd. Het resulterende niet-lineaire niet-convexe MPC-optimalisatieprobleem wordt geherformuleerd tot een MILP-probleem dat zeer efficiënt kan worden opgelost met bestaande MILP-oplossers. Het gemengd-geheeltallige programmeerprobleem ondervindt doorgaans rekenproblemen vanwege de schaal van het probleem, en de rekencomplexiteit wordt sterk beïnvloed door het aantal gehele variabelen. Daarom ontwikkelen we een op leren gebaseerde MPC-aanpak, waarbij de gehele variabelen bij elke tijdstap worden verkregen via *deep learning*, en de MPC-probleem een wordt gereduceerd tot niet-lineair optimalisatieprobleem met alleen reële variabelen en met minder variabelen hoeft op te lossen dan het oorspronkelijke probleem. Simulatieresultaten tonen aan dat de op leren-gebaseerde MPC-aanpak, vergeleken met de oorspronkelijke MPC-aanpak, de rekestijd aanzienlijk vermindert, terwijl vergelijkbare prestaties worden behaald.

In het vierde onderwerp onderzoeken we de snelheidsregeling van virtueel gekoppelde treinen, wat cruciaal is voor treinen om zich aan de dienstregeling te houden. Virtuele koppeling wordt beschouwd als een efficiënte manier om de lijncapaciteit van spoorwegsystemen te verbeteren door de afstand tussen opeenvolgende treinen te verkleinen. In de praktijk zijn de massa's van treinen verschillend en kunnen ze op stations veranderen door variaties in de passagiersbelasting, wat de dynamiek en besturing van de virtueel gekoppelde treinen beïnvloedt. Rekening houdend met het niet-lineaire treinmodel en veranderingen in de massa's van treinen, worden coöperatieve gedistribueerde MPC, seriële gedistribueerde MPC en gedecentraliseerde MPC vergeleken en beoordeeld voor de besturing van virtueel gekoppelde treinen. Om een gebalanceerde afweging te maken tussen rekencomplexiteit en efficiëntie, stellen we ook convexe benaderingen voor en vergelijken we deze voor de bovengenoemde regelmethode. We introduceren gerelaxeerd dynamisch programmeren in het domein van treinbesturing, en we ontwikkelen een gedistribueerd stopcriterium met stabiliteitsgarantie voor de coöperatieve gedistribueerde MPC-aanpak. Simulatieresultaten geven aan dat de coöperatieve gedistribueerde MPC-aanpak de beste volgpresetaties oplevert, terwijl de seriële gedistribueerde MPC-aanpak de communicatie-eisen en rekenkost kan verminderen, maar met een afname in volgpresetaties.

Samenvattend behandelt dit proefschrift de integratie van passagiersstromen, dienstregelingen en treinsnelheden door verschillende MPC-methodes te ontwikkelen, waaronder twee-laags MPC, scenario gebaseerde gedistribueerde MPC, op leren gebaseerde MPC en coöperatieve gedistribueerde MPC. Deze benaderingen worden toegepast op het verkeersmanagement van stedelijke spoorwegnetwerken en leveren verbeterde prestaties op in vergelijking met traditionele methoden.

1

INTRODUCTION

1.1. BACKGROUND

Urban rail transit plays a prominent role in public transportation systems due to its safety, efficiency, high transport capacity, and environment-friendly characteristics. According to the data published by The International Association of Public Transport, as of 31 December 2020, 193 cities worldwide have implemented 731 urban rail lines, covering more than 17000 kilometers in total¹. These systems facilitate transportation and significantly reduce urban congestion and pollution, emphasizing their importance in sustainable urban development and economic prosperity. With the rapidly growing passenger demands and the increasing scale of urban rail networks, reliable urban rail transit systems have become increasingly important for the competitiveness of national and regional economies.

Urban rail transit systems primarily focus on ensuring that trains operate safely and efficiently to deliver high-quality service to passengers. Generally speaking, there are three key elements for train operations in urban rail transit networks, namely, passenger flows, timetables, and train speeds. In principle, the planning, management, and control of train operations in urban rail transit systems should incorporate these elements to achieve a high standard of reliability and efficiency, thereby ensuring that passengers receive satisfactory service while reducing operational costs.

The operation of trains in urban rail transit systems typically follows a hierarchical structure, as illustrated in Figure 1.1, consisting of three levels, i.e., strategic scheduling, timetable (re)scheduling, and train speed control [89; 132]. The strategic scheduling level involves planning according to the general key information in an urban rail transit network, such as passenger demands, rolling stock circulation, and crew scheduling, thereby determining a strategic timetable to minimize operational costs and total passenger travel times. In real-time operations of urban rail transit networks, i.e., at the timetable (re)scheduling level, this strategic timetable may require rescheduling to reflect real-time information, i.e., more elaborate passenger demands, disturbances,

¹[https://www.uitp.org/publications/metro-world-figures-2021/](https://www UITP.org/publications/metro-world-figures-2021/)

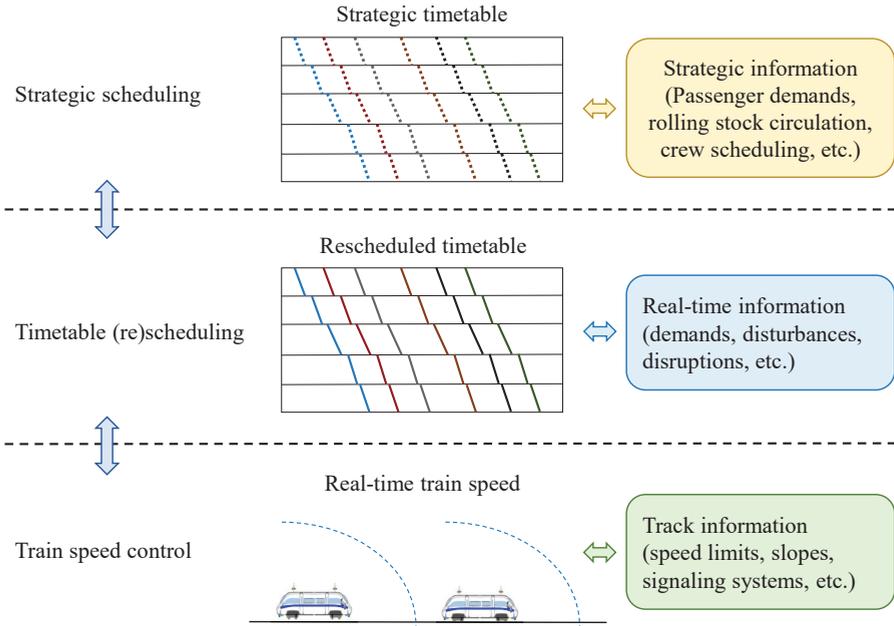


Figure 1.1: Hierarchical structure of train operations in urban rail transit systems (adopted from [89; 132]).

disruptions, etc. The objectives of the timetable (re)scheduling level are typically minimizing energy consumption, passenger delays, and train delays, and urban rail systems can therefore maintain efficient and reliable operations, ensuring they meet both immediate and future transportation needs effectively. Strategic scheduling and timetable rescheduling generally fall under the scope of traffic management, which is typically managed by dispatchers in traffic management centers. A comprehensive overview of railway traffic management approaches can be found in [13; 25].

Train speed control is responsible for implementing a given timetable by regulating trains running between two stations. Train speed control can be divided into train speed profile optimization and train speed control, wherein the train speed profile is generated first, and then the train tracks the profile in real time. The main focus of train speed control is on safe and efficient train movement on a track under restrictions of signaling systems, i.e., fixed block signaling systems, moving block signaling systems, and virtual coupling systems. Fixed block signaling systems divide the track into several fixed blocks, and each block can only be occupied by a single train at any given time, providing a safe distance of at least one block. In urban rail transit systems, trains operate with relatively high frequency, and moving block signaling systems are typically applied, which determine the distance between two consecutive trains based on the absolute braking distance, i.e., the distance a train needs to fully stop from its current speed. Recent advances in vehicle-to-vehicle communication and cooperative control have propelled virtual coupling to the forefront technology of further enhancing line capacity. In a platoon

of virtually coupled trains, the distance between two consecutive trains is determined based on the relative braking distance, which takes into account the braking characteristics of the predecessor train.

1.2. MOTIVATION AND CHALLENGES

Urban rail transit systems prioritize safe and efficient train operations while providing high-quality service to passengers. As passenger demands and the scale of urban rail networks continuously increase, how to realize flexible, highly efficient, and intelligent control of urban rail transit operations has become an emerging research topic in recent years. Urban rail transit systems operate under strict restrictions, such as infrastructure restrictions, timetabling rules, and train capacity limitations, which make real-time control problems in urban rail transit systems challenging.

Model predictive control (MPC) [46; 92; 104] has been widely adopted in the industry for its conceptual simplicity and its ability to handle multivariable constrained control problems. In the MPC scheme, the control action is obtained by solving an optimization problem over a finite-horizon window yielding a control sequence, and only the first control action is implemented in the real system. At the next control step, the optimization is conducted again using updated state information with a shifted finite-horizon window. This moving horizon optimization procedure is repeated until the end of the overall control period. The framework of MPC demonstrates its promising application prospects in urban rail transit systems. However, several issues exist, as outlined below. The first three items pertain to computational complexity, while the fourth item relates to communication limitations.

1. As a model-based approach, the performance of MPC significantly depends on the model accuracy and model complexity. There are three key elements for train operation in urban rail transit networks, i.e., passenger flows, timetables, and train speeds. In particular, passengers have different origins and destinations in urban rail transit networks, and passenger flows typically show time-varying characteristics. There is a trend to develop more elaborate models for traffic management to provide more effective control decisions. However, including too many details will significantly increase the computational complexity, thereby influencing the real-time feasibility of control approaches. Therefore, it is challenging to develop an efficient model for urban rail transit networks that can make a balanced trade-off between model accuracy and computational efficiency.
2. The traffic management and train control problem in urban rail transit networks is challenging because of its problem scale and network complexity. On the one hand, the online computation complexity of MPC is significantly influenced by the scale of the problem. For urban rail transit networks, the problem scale increases rapidly as the network expands, rendering centralized MPC difficult to implement in real-life networks due to its computational demands. On the other hand, the real-life communication topologies in urban rail transit networks may also pose challenges for implementing the centralized MPC approach. These factors highlight the increasing urgency of leveraging the inherent structure of the considered

traffic management problem and developing effective non-centralized MPC approaches.

3. Formulating the traffic management problem results in a mixed-integer programming problem, which is typically large-scale, NP-hard, and non-convex. This type of programming problem commonly encounters computational challenges, with complexity significantly influenced by the number of integer variables involved. As an optimization-based control approach, MPC should solve the mixed-integer programming problem at every control step, posing a notable challenge in managing the online computational burden. Despite efforts using various approaches, such as branch-and-bound [64] or Benders decomposition [8; 107], to relax integer variables and to reduce problem complexity, the enumeration or iteration process usually cannot guarantee a satisfactory solution within a given time. Therefore, efficiently obtaining a satisfactory solution for mixed-integer programming problems remains a challenge in the MPC scheme.
4. Virtual coupling is regarded as an efficient way to improve the line capacity of rail transportation systems by reducing the spacing between consecutive trains. Virtual coupling relies on vehicle-to-vehicle communication and cooperative train control schemes. However, as an emerging technology, a comprehensive comparison and assessment considering different models and different control schemes for virtually coupled trains is still unaddressed in the existing literature. Furthermore, due to communication and computational power limitations, the real-time application of distributed MPC may terminate its iteration before reaching the optimal solution, thereby resulting in inexact minimization. The inexact optimization can result in constraint violation issues and pose a risk to stability as feasibility and stability are typically defined considering the optimal solution. In this context, constraint satisfaction and stability of distributed MPC approaches for virtually coupled trains are still challenging issues.

1.3. RESEARCH QUESTIONS

According to the motivation and challenges, one main research question and six sub-questions will be answered in this thesis. The main research question is

How can model predictive control benefit flexible, highly efficient, and passenger-oriented urban rail transit network operations?

Six subquestions are given as follows:

1. How to realize the integration of passenger flows, timetables, and train speeds in urban rail transit networks?
2. How to apply model predictive control in passenger-oriented real-time train scheduling?
3. How to achieve efficient train rescheduling for large-scale urban rail transit networks?

4. What are efficient modeling and control methods for handling time-varying passenger demands in real-time train scheduling?
5. How to reduce the online computational burden of model predictive control in train rescheduling problems?
6. What are effective control approaches for virtually coupled trains considering communication topologies and heterogeneous trains?

1.4. CONTRIBUTIONS

The main contributions of the thesis are introduced next:

1. We develop a bi-level model for the integration of passenger flows, timetables, and train speeds. At the higher level, we develop a passenger absorption model to determine train departure frequencies that explicitly includes time-dependent passenger origin-destination demands in urban rail transit networks, where the term “absorption” refers to passengers boarding trains. The lower-level model is a timetable scheduling model including detailed timetables and train speed profiles. Then, a bi-level MPC approach is proposed to reduce the total passenger travel time and train energy consumption. Furthermore, we show that the optimization problems of both levels of the bi-level MPC approach can be converted into mixed-integer linear programming (MILP) problems, which enables us to solve them with existing MILP solvers.
2. We propose a distributed MPC approach for train departure frequency optimization in urban rail transit networks to deal with the computational burden and the communication restrictions. In the proposed distributed MPC approach, a cost-to-go function is designed to reduce the prediction horizon of the original MPC approach while taking into account the control performance. Furthermore, we incorporate a scenario-based distributed control scheme into the developed distributed MPC approach to handle uncertain passenger flows in large-scale urban rail transit networks.
3. We develop a passenger-oriented model for timetable rescheduling. By discretizing the planning time window into several time intervals of equal length, every time interval includes one departure of a train at the same platform. The time-varying passenger demands are approximated as piecewise constant functions in the model to achieve a trade-off between model accuracy and computation speed. A centralized MPC approach is then developed where a mixed-integer programming problem should be solved in each control step. Furthermore, we develop a learning-based MPC approach to reduce the online computational burden of MPC. In the proposed learning-based MPC approach, we obtain the integer variables by deep learning, and then the MPC optimizer only needs to solve a continuous nonlinear optimization problem with fewer variables than the original problem at each control step, thereby reducing computational complexity.
4. We compare and assess cooperative distributed MPC, serial distributed MPC, and decentralized MPC for virtually coupled trains. Relaxed dynamic programming

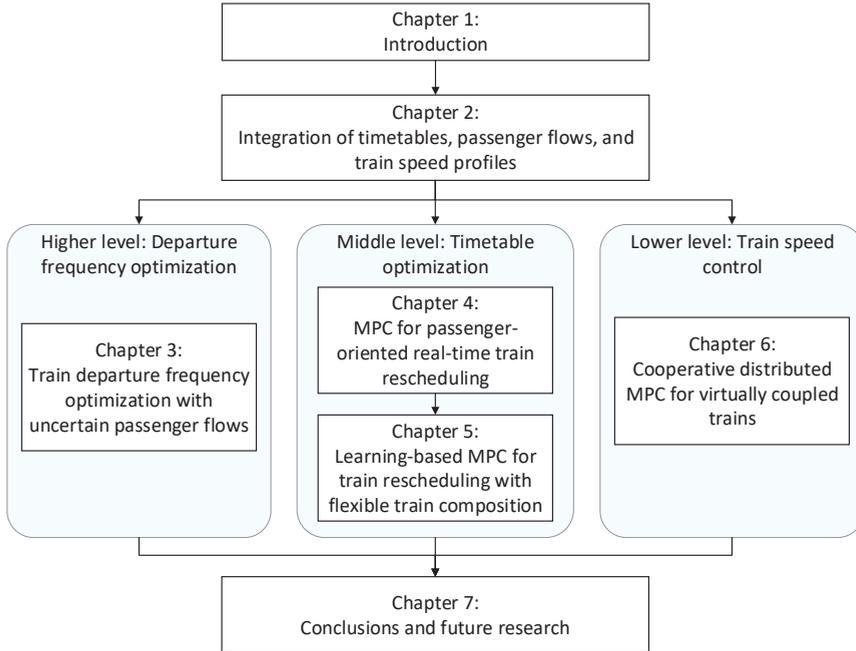


Figure 1.2: Outline of the thesis.

is a technique used to assess the stability of optimal control under suboptimality estimates. Furthermore, we are the first to incorporate the relaxed dynamic programming approach into the train control field. Moreover, a stopping criterion under the distributed MPC scheme with a stability guarantee is developed for the cooperative distributed MPC approach. Moreover, we explicitly consider the changes in train masses when designing MPC approaches, and we illustrate the impact of incorporating train masses in the control design through simulations.

1.5. THESIS OUTLINE

This thesis consists of seven chapters, and the thesis structure is illustrated in Figure 1.2. Chapter 2 to Chapter 6 are a collection of published or submitted journal papers. As different chapters focus on different aspects in urban rail transit networks, the mathematical notations are defined for each chapter separately.

The main contents from Chapter 1 to Chapter 7 are briefly introduced as follows. Chapter 1 provides a general introduction to urban rail transit networks and to the thesis. Chapter 2 develops a bi-level model and a bi-level MPC approach for the integration of passenger flows, timetables, and train speeds. In Chapter 3, a distributed MPC approach is presented to determine train departure frequencies in urban rail transit networks, and

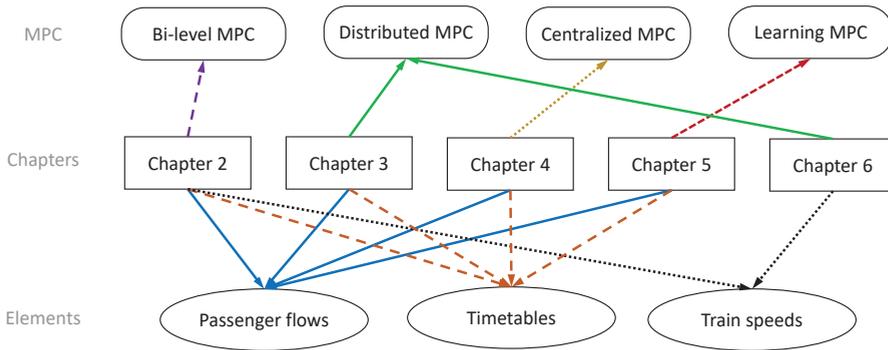


Figure 1.3: Categorization of the contents for the main chapters.

a scenario-based approach is applied to handle uncertain passenger demands. Chapter 4 develops a model with a centralized MPC approach to adjust timetables based on real-time passenger demands. In Chapter 5 a learning-based MPC approach is proposed for timetable rescheduling considering real-time passenger demands and train composition optimization. Chapter 6 investigates cooperative distributed MPC, serial distributed MPC, and decentralized MPC for virtually coupled trains. Chapter 7 concludes the whole thesis and provides an outlook for future work.

Apart from the outline figure, we also categorize the contents of the main chapters in Figure 1.3. The categorization is from both the MPC side (bi-level MPC/ distributed MPC/ centralized MPC/ learning-based MPC) and the elements side (passenger flows/ timetables/ train speeds). The arrows in Figure 1.3 represent the categorization of the chapters.

2

INTEGRATION OF TIMETABLES, PASSENGER FLOWS, AND TRAIN SPEED PROFILES

This chapter deals with the train scheduling problem for urban rail transit networks taking into account time-dependent passenger origin-destination demands and train speed profiles. The aim is to adjust train schedules online according to time-dependent passenger demands so that passenger satisfaction and operational costs are jointly optimized. A passenger absorption model that explicitly includes time-dependent passenger origin-destination demands is developed, where the term “absorption” refers to passengers boarding trains. Then, the passenger absorption model is extended to a bi-level framework, where passenger demands and rolling stock availability are considered at the higher level, and detailed timetables and train speed profiles are included at the lower level. A bi-level model predictive control (MPC) approach is developed for the integrated problem. The optimization problems of both levels of the bi-level MPC approach can be converted into mixed-integer linear programming (MILP) problems, which enables us to solve them with existing MILP solvers. We then show that the recursive feasibility of both the higher-level and the lower-level optimization problems can be guaranteed. In this way, we can achieve real-time train scheduling for the urban rail transit system. Numerical experiments, based on real-life data from the Beijing urban rail transit network, illustrate the effectiveness of the extended passenger absorption model and the proposed bi-level MPC approach.

This chapter is based on [74] and [79].

2.1. INTRODUCTION

As a safe, efficient, and eco-friendly transportation mode, the urban rail transit system plays a prominent role in public transportation. Real-time train scheduling is recognized as a valuable method for improving passenger satisfaction and energy efficiency under infrastructure limitations. As urban rail transit systems continue to expand to large-scale and networked systems, it becomes increasingly challenging to achieve real-time train scheduling while taking time-varying passenger flows and operational costs into account [51; 136].

Generally speaking, there are three key elements for train operation in urban rail transit networks, i.e., passenger flows, timetables, and train speeds. Some data-driven approaches can be applied to predict the near future passenger flow information in real time, which can be represented by time-dependent origin-destination (OD) matrices, thereby facilitating timetable scheduling [98]. An efficient passenger-oriented timetable should properly address time-dependent passenger OD demands [136]. Train speeds are closely related to operation time and energy consumption [86; 152]. As train speed control between two stations is usually conducted under the guidance of a recommended train speed profile, a well-designed speed profile is crucial for effective train speed control [51]. The integration of timetables, passenger flows, and train speed profiles is desired to generate efficient timetables that can jointly consider passenger satisfaction and operational costs in urban rail transit networks.

Real-time train scheduling considering passenger flows and train speed profiles is challenging due to its complexity and scale. Many studies include passenger flows in train scheduling problems while also considering stopping patterns of trains [14], short-turning [158], and rolling stock circulation [47; 156], but without time-dependent passenger origin-destination demands. Furthermore, train speed profiles are not included in these studies, and thus train speed-related objectives, e.g., the energy consumption of trains, cannot be directly included in the passenger-oriented train scheduling problem. Several papers consider the integration of timetables, passenger flows, and train speeds [93; 135; 136; 152]. However, most existing studies that consider both passenger OD demands and train speed profiles, are limited to a single line because of the computational complexity issues arising from the integrated problem. This chapter therefore focuses on the integration of timetables, passenger flows, and train speed problems for urban rail transit networks.

To reduce the computational burden of including many microscopic details of the network, some studies develop macroscopic models to handle passenger OD demands by optimizing departure frequencies [19; 49; 66]. The train departure frequency (i.e., the number of trains departing from a platform per time unit) is crucial for passenger satisfaction since it determines the maximum transport capacity of each line. The departure frequency should be adjusted properly to match time-varying passenger flows, e.g., compared with off-peak hours, higher departure frequencies are required during peak hours to address the large passenger demands. Furthermore, the departure frequency should be linked with specific departure and arrival times for a practically implementable timetable. Therefore, effective model formulations and control approaches are required to integrate train departure frequencies and train timetables in urban rail transit networks.

This chapter contributes to the state of the art as follows.

1. A passenger absorption model is developed that determines train departure frequencies in urban rail transit networks by explicitly including time-dependent passenger origin–destination demands.
2. A bi-level model predictive control (MPC) approach is proposed for real-time train scheduling considering passenger flows, rolling stock circulations, and train speed profiles. Passenger flows are included at the higher level based on the novel extended passenger absorption model, and detailed timetables and train speed profiles are incorporated at the lower level taking into account the detailed rolling stock circulation. The MPC optimization problems of both levels are exactly converted to mixed-integer linear programming problems, and we show that the recursive feasibility of both levels can be guaranteed.

The remaining part of the chapter is arranged as follows: Section 2.2 reviews the related works. Section 2.3 provides the problem statement and assumptions for this chapter. Section 2.4 introduces the developed passenger absorption model and the corresponding bi-level modeling framework. Section 2.5 introduces the developed bi-level MPC approach. Section 2.6 shows the effectiveness of the developed approach through numerical experiments, and conclusions are provided in Section 2.7.

2.2. STATE OF THE ART

2.2.1. PASSENGER-ORIENTED REAL-TIME TIMETABLE SCHEDULING

There exists a considerable body of research on passenger-oriented timetable scheduling problems. Cury et al. [27] presented an analytical model to describe the movement of trains and passengers; then, the optimal schedule is generated considering operational costs and the average delay of passengers. Wang et al. [135] developed an iterative algorithm to reduce the total passenger travel time on an urban rail transit line while considering the energy efficiency of trains, where train speeds in each segment were simplified via three stages, i.e., acceleration stage, cruising stage, and deceleration stage. Wang et al. [133] realized real-time train scheduling for an urban rail transit line by integrating passenger demands and rolling stock circulation, and the aim is to ensure service quality while reducing operational costs. Hou et al. [51] considered unexpected disturbances in an urban rail transit system and solved a mixed-integer linear programming (MILP) problem to reduce train delays, energy consumption, and the number of stranded passengers, where train speeds were also limited to a finite set of different speed levels. Considering train loading capacity constraints, Mo et al. [93] formulated an MILP problem to maximize the utilization of regenerative energy, where rolling stock circulation was also incorporated into the resulting train scheduling problem. However, these studies do not include passenger origin–destination (OD) demands, indicating the possibility of further improving passenger satisfaction.

Real-time train scheduling with detailed passenger OD demands has received much attention in recent years. Niu et al. [97] formulated a mixed-integer nonlinear programming (MINLP) problem for train scheduling in a rail corridor to reduce passenger waiting time taking into account time-dependent passenger demands. A space-time network

was used in [152] to describe the movement of trains on an urban rail transit line, where the train operation in a segment is considered for different speed levels; a Lagrangian relaxation-based method was then presented to optimize the total passenger waiting time and operational costs. Qi et al. [103] optimized train stopping plans and timetables of a high-speed railway line considering time-dependent passenger OD demands. The aim of [103] is to find a solution that satisfies passenger preferences for departure times and that ensures trains operate within capacity limits; the given problem was formulated as an MILP problem. Bešinović et al. [9] integrated passenger flow control and train rescheduling under disruptions, and applied an iterative matheuristic approach to reduce the passenger waiting time and the time of recovering from disruptions. Nevertheless, these papers only include passenger OD demands on a single railway line, and further research is still required for the railway network.

Considering passenger OD demands in railway networks, Wang et al. [136] presented an event-based model that explicitly includes time-dependent passenger OD demands intending to minimize the total passenger travel time and the energy consumption of trains. Train arrival, train departure, and passenger arrival rate changes were formulated as three different classes of events in [136] to describe the movement of passengers and trains. Yin et al. [150] formulated a graph-based model to describe feasible passenger travel paths in an urban rail transit network; then, a decomposition-based adaptive large-neighborhood search approach is presented to minimize station crowdedness. Zhu & Goverde [158] developed a timetable rescheduling approach for disruptions in a railway network based on an event-activity model, where passenger OD demands and passenger paths are included and used to determine weights of different objectives. Corman [23] investigated the interactions between train schedules and passenger route choices, and presented a game theory-based approach to investigate the equilibrium point between them. Luan & Corman [84] formulated the train schedules and passenger routing process in an integrated model, and the resulting MINLP formulation is reformulated as an MILP formulation to minimize passenger disutility (i.e., the number of stranded passengers, the passenger delays, and the passenger travel time) and the total train delay. However, these studies typically encounter computational issues because more details about passenger demands and railway networks should be included. Therefore, efficient model and solution approaches are required for passenger-oriented train scheduling.

2.2.2. PASSENGER-ORIENTED TRAIN DEPARTURE FREQUENCY OPTIMIZATION

The studies introduced in Section 2.2.1 aim to build elaborate models for detailed passenger dynamics and infrastructure information. These studies can generate directly implementable arrival and departure times of trains; however, the computational burden increases as many details related to passenger dynamics are included using such detailed microscopic models. In order to obtain a balanced trade-off between model accuracy and computational efficiency, another research direction develops macroscopic models to handle passenger OD demands by optimizing departure frequencies [19; 66; 74], considering the periodic characteristic of train departures.

Optimizing the departure frequency determines the maximum transport capacity and is essential for handling passenger demands in urban public transport systems, e.g.,

city bus systems [65] and urban rail transit systems [49]. In general, higher departure frequencies typically result in higher operational costs while providing a better chance of boarding trains for passengers. The urban rail transit system, however, is quite different from other urban public transport systems, e.g., the braking distance of trains is relatively long, and the signaling system imposes an upper bound on the line frequency. Thus, effective departure frequency control approaches are required for urban rail transit networks to address time-dependent passenger OD demands considering operational costs and infrastructure constraints. Canca et al. [19] solved a MINLP problem to optimize train capacities and line frequencies for each line of urban rail transit networks considering both track allocations and passenger assignments, where train capacities were considered as soft constraints. Li et al. [66] developed a bi-level strategy to optimize the train departure frequencies at the upper level while a passenger assignment problem was considered at the lower level to balance operational cost and service quality. These studies aim to generate static and published train departure frequencies and schedules at the tactical planning stage based on periodic passenger flows, leaving an open gap in optimizing departure frequencies online based on real-time observed passenger demand.

Adjusting departure frequency online is also regarded as an effective way to accommodate time-dependent passenger demand [42]. Pu & Zhan [102] developed a two-stage method for railway line planning problems where the first stage generates a line plan with deterministic passenger demands and the second stage adjusts the line plan to accommodate real-life passenger demands. Liu et al. [74] presented a passenger flow model to determine departure frequencies of urban rail transit systems in real time. However, that paper does not lead to a directly implementable timetable, i.e., specific arrival and departure times are not considered, and the case when different lines use the same physical track and/or physical platforms is also not involved. In summary, the above-mentioned studies only optimize the departure frequency of trains, which does not directly lead to practically executable timetables. Moreover, more detailed passenger flows, rolling stock circulation plans, and operational costs can be included to further improve operational performance.

2.2.3. MPC FOR REAL-TIME RAILWAY TRAIN SCHEDULING

The studies introduced in Section 2.2.1 and Section 2.2.2 are summarized in Table 2.1 based on the railway network details, passenger demands, and objectives. The train scheduling problem is a typical control problem with input and state constraints. From Section 2.2.1 and Section 2.2.2, we can conclude that efficient modeling frameworks and control approaches for the integration of timetables, passenger flows, and train speeds in urban rail transit networks are urgently needed to achieve passenger-oriented train scheduling.

Model predictive control (MPC) is regarded as an efficient control methodology for real-time control of constrained systems [92]. MPC has also been implemented in real-time train scheduling problems. Van den Boom et al. [125] applied MPC to minimize the delay of trains and the costs of changing train orders and braking connections by developing a switching max-plus-linear model. Caimi et al. [15] applied the MPC framework and proposed a scheduling assistant method for complex station areas considering in-

infrastructure constraints and passenger satisfaction. Li et al. [69] proposed a state space model to represent the dynamics of the train capacity and departure times on an urban rail transit line and an MPC approach was then developed to minimize the headway and timetable deviations by adjusting timetables and train capacity. Liu et al. [76] applied MPC to passenger-oriented urban rail transit networks to adjust a given timetable based on real-time passenger demands with the aim of minimizing the total travel time of passengers.

In addressing large-scale systems, many studies have focused on the development of non-centralized MPC methodologies, solving the problem in decentralized, distributed, or hierarchical manners to obtain efficient solutions for the overall system [56; 62; 90]. Such approaches have found application in railway train scheduling problems as well. Kersbergen et al. [55] introduced various distributed MPC techniques for optimizing railway traffic management, encompassing the collective optimization of arrival and departure times, infrastructure connections, and train sequencing within the railway network. Luan et al. [85] employed three distributed optimization methodologies, i.e., the alternating direction method of multipliers, a priority-rule-based approach, and the cooperative distributed robust safe but knowledgeable (CDRSBK) algorithm, for real-time traffic management in railway networks. Through numerical simulations, it was demonstrated that the CDRSBK approach, utilizing train-based decomposition, outperforms the other approaches in terms of feasibility, optimality, and computational efficiency. Wang et al. [138] introduced a hierarchical MPC framework to integrate railway delay management and train control, which can realize real-time control and reduce delays effectively. Cavone et al. [22] applied MPC to address disruptions and disturbances in railway networks, where an MILP problem is formulated under a bi-level structure by dividing the model into macroscopic and mesoscopic levels. The successful applications of the aforementioned centralized and non-centralized MPC approaches have motivated us to design an efficient MPC approach to realize real-time train scheduling.

We therefore develop a bi-level MPC approach for real-time train scheduling while considering time-dependent passenger OD demands and train speed profiles in urban rail transit networks. A bi-level model is developed to reduce the computational complexity of the integrated problem, and then the corresponding bi-level MPC approach is proposed. The higher-level controller is conducted with relatively slow dynamics to optimize departure frequencies (i.e., the number of trains departing from a platform per time unit), while the lower-level controller calculates detailed timetables with fast dynamics considering train scheduling constraints. The MPC optimization problems of both levels are transformed exactly into MILP problems, which enables us to solve them with existing MILP solvers.

Publications	Infrastructure	Passenger demands	Train capacity	Rolling stock circulation	Train order	Train speed	Objective (s)
Cury et al. (1980) [27]	bi-directional line	OD-independent	no	no	no	no	minimize passenger delays and total number of trains
Niu et al. (2015) [97]	uni-directional line	OD-dependent	hard constraint	no	no	no	minimize the total passenger waiting time at stations
Wang et al. (2015) [135]	uni-directional line	OD-independent	hard constraint	no	no	continuous speed	minimize train energy consumption and total passenger travel time
Wang et al. (2015) [136]	network	OD-dependent	hard constraint	no	no	continuous speed	minimize total passenger travel time and train energy consumption
Canca et al. (2016) [19]	network	OD-dependent	soft constraint	yes	no	no	minimize total passenger travel time and operational costs
Yin et al. (2017) [152]	bi-directional line	OD-dependent	hard constraint	no	no	speed levels	minimize total passenger waiting time and train energy consumption
Li et al. (2018) [66]	uni-directional line	OD-dependent	hard constraint	no	no	no	optimizing departure frequency to balance operational cost and service quality
Wang et al. (2018) [133]	bi-directional line	OD-independent	soft constraint	yes	no	no	minimize load factor variation, headway variation, and entering depot operations
Hou et al. (2019) [51]	uni-directional line	OD-independent	hard constraint	no	no	speed levels	minimize train delays, energy consumption, and number of stranded passengers
Zhu and Goverde (2019) [158]	network	OD-dependent	no	yes	yes	no	minimize passenger delays and impacts of cancelling trains and skipping stops
Mo et al. (2020) [93]	bi-directional line	OD-independent	hard constraint	yes	no	no	maximize utilization of regenerative energy
Corman (2020) [23]	network	OD-dependent	no	no	yes	no	analyse equilibrium point between train schedules and passenger route choices
Pu and Zhan (2021) [102]	uni-directional line	OD-dependent	hard constraint	no	no	no	minimize operational costs and total passenger travel time
Yin et al. (2021) [150]	network	OD-dependent	hard constraint	no	no	no	minimize station crowdedness
Qi et al. (2021) [103]	uni-directional line	OD-dependent	hard constraint	no	yes	no	minimize passenger waiting time and deviation from original timetable
Bešinović et al. (2022) [9]	bi-directional line	OD-dependent	hard constraint	yes	no	no	minimize the running time of trains from origin station the terminal station
Luan and Corman (2022) [84]	network	OD-dependent	hard constraint	no	yes	no	minimize passenger disutility and total train delay
Chapter 2	network	OD-dependent	hard constraint	yes	yes	speed levels	minimize total passenger travel time and train energy consumption

Table 2.1: Summary of the relevant studies on passenger-oriented timetable scheduling.

2.3. PROBLEM STATEMENT AND ASSUMPTIONS

2.3.1. PROBLEM STATEMENT

In urban rail transit systems, train schedules should be adjusted throughout the day to accommodate time-varying passenger flows while taking operational costs into account. A pre-determined timetable cannot include time-dependent passenger demands information and, in general, may be far from optimal. This chapter focuses on adjusting train schedules online based on time-dependent passenger origin-destination demands while taking into account train capacity, rolling stock circulation, train speed profiles, and train orders. As discussed in Section 2.2, the time-dependent passenger-oriented train scheduling problem typically has computational issues. We therefore handle the problem in a bi-level framework to achieve a balanced trade-off between model accuracy and computational burden.

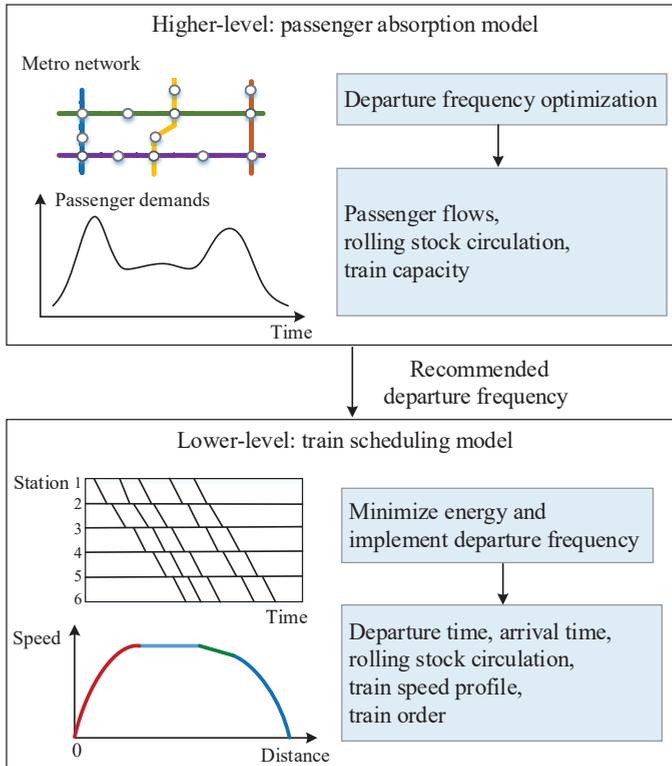


Figure 2.1: Illustration of the bi-level framework.

The general idea of the bi-level framework is illustrated in Fig. 2.1. The train departure frequency determines the upper bound of the transport capacity and is included at the higher level to address the time-dependent passenger OD demands based on the developed passenger absorption model. As the departure frequency is restricted by the

availability of rolling stock, the rolling stock circulation is also considered at the higher level. The lower level focuses on generating a practically implementable timetable to fulfill the departure frequency while considering detailed rolling stock circulation, train speed profiles, and train orders.

2.3.2. EXPLANATIONS AND ASSUMPTIONS

Some general explanations and assumptions about the problem are listed as follows:

(1) A line in the urban rail transit network is typically defined as the route of one certain class of train services; these train services thus visit identical stations in each run. The assigned platforms for trains of each line are fixed.

(2) Passenger OD demands (i.e., the number of passengers choosing the urban rail transit for their travel, their origins, and their destinations) are not influenced by the departure frequencies. Time-dependent passenger OD demands are approximated as piece-wise constant functions.

(3) As passenger route choices observed from urban rail transit data collection systems typically exhibit consistent patterns [98], we assume that the fractions of passengers choosing each route are given a priori, and that passengers do not change their route once they have entered the urban rail transit network.

(4) As we assume that passengers do not change their routes once they have entered the urban rail transit network, we define a lower bound for the departure frequency, so that the time interval between the departures of two consecutive trains is always shorter than a given threshold. In this way, the maximum waiting time for passengers should still be acceptable in case the departure frequency and/or departure times change with respect to the original timetable.

2.4. MATHEMATICAL MODEL

Based on the bi-level framework, a bi-level model is presented for the passenger-oriented train scheduling problem, where (1) a macroscopic model, i.e., passenger absorption model, is included at the higher level considering time-independent passenger OD demands, rolling stock circulation, and train departure frequencies, and (2) a train scheduling model is included at the lower level considering the detailed timetable, detailed rolling stock circulation, train speed profiles, and train orders. In this section, we first provide the notations for the mathematical models. Then, the passenger absorption model and the train scheduling model are introduced respectively.

2.4.1. NOTATIONS

Tables 2.2, 2.3, and 2.4 respectively list the indices and input parameters, decision variables, and output variables for the model formulations. Noting that in Table 2.3 the decision variables for the higher level are the departure frequency $u_\ell(k)$ for all lines while the arrival time $a_{i,p}$, departure times $d_{i,p}$, $a_{i,\ell}^{\text{depot}}$, and speed profile option $x_{i,p,b}$ for all trains at all line platforms are the decision variables for the lower level.

Notations	Definition
i, j	Index of trains
p, q	Index of line platforms, $p \in \mathcal{P}, q \in \mathcal{P}$, \mathcal{P} is the set of line platforms
ℓ	Index of lines, $\ell \in \mathcal{L}$, \mathcal{L} is the set of lines
s, e	Index of stations, $s, e \in \mathcal{S}$, \mathcal{S} denotes the set of stations, s_p is the station corresponding to line platform p
z	Index of depots, $z \in \mathcal{Z}$, \mathcal{Z} denotes the set of depots
k	Index of phases
T	Length of a phase
$p_\ell^{\text{tra}}(i)$	Preceding train of train i at line ℓ
$p^{\text{pla}}(p)$	Preceding line platform of line platform p
$\rho_{s,e}^{\text{station}}(k)$	Passenger arrival rate at station s with destination e during phase k
$\rho_{p,e}(k)$	Passenger arrival rate at line platform p with destination e during phase k
$\lambda_{s,p,e}(k)$	Proportion of passengers at station s that are assigned to line platform p for their travel to destination e during phase k
$\alpha_{p,e}(k)$	Proportion of passengers absorbed by trains at line platform p with destination e during phase k
C_{train}	Maximum capacity of a train
$\chi_{p,q,e}$	Proportion of passengers transferring from line platform p to q with destination e
$\text{cop}(p)$	The set of line platforms located at the identical station as line platform p
$\text{in}(z)$	The set of platforms related to the entering link of depot z
$\text{out}(z)$	The set of lines corresponding to the output link of depot z
N_z^{train}	The number of available trains at depot z
$t_{p,q}^{\text{transfer}}$	Average time for passengers transferring from line platform p to line platform q
h_p^{min}	Minimum departure-arrival headway at line platform p
τ_p^{min}	Minimum dwell time of train at line platform p
τ_p^{max}	Maximum dwell time of train at line platform p
r_p^{min}	Minimum running time of train from line platform p to its succeeding line platform
r_p^{max}	Maximum running time of train from line platform p to its succeeding line platform
$\mathcal{B}_{i,p}$	Speed profile options set for train i from line platform p to its succeeding line platform
$t_{i,p,b}$	Running time of train i from line platform p to its succeeding line platform with speed profile b , $b \in \mathcal{B}_{i,p}$
$\sigma_{p,p'}$	Binary parameter; if line platforms p and p' correspond to the same physical platform, $\sigma_{p,p'} = 1$; otherwise, $\sigma_{p,p'} = 0$

Table 2.2: Indices and input parameters.

Notations	Definition
$u_\ell(k)$	The departure frequency from the depot corresponding to line ℓ during period k
$a_{i,p}$	Arrival time of train i at line platform p
$d_{i,p}$	Departure time of train i at line platform p
$d_{i,\ell}^{\text{depot}}$	Departure time of train i from the depot corresponding to line ℓ
$x_{i,p,b}$	Binary variable indicating whether train i from line platform p selects speed profile b

Table 2.3: Decision variables.

2.4.2. PASSENGER ABSORPTION MODEL

This section presents a macroscopic model to determine train departure frequencies based on the time-dependent passenger OD demands. In the passenger absorption model, the planning time window is divided into several phases, and in each phase, the time-dependent passenger demands at each platform are considered to be constant. The train departure frequency during each phase can be optimized while taking into ac-

Notations	Definition
$\tau_{i,p}$	Dwell time of train i at line platform p
$r_{i,p}$	Running time of train i from line platform p to its succeeding line platform
\bar{r}_p	Average running time of trains from line platform p to its succeeding line platform
$\gamma_p(k)$	Average time for a train from the first line platform to line platform p at phase k
$\beta_p(k)$	The largest integer less than or equal to $\frac{\gamma_p(k)}{T}$
$\phi_p(k)$	The remainder of $\frac{\gamma_p(k)}{T}$
$n_{p,e}(k)$	Number of passengers at line platform p with destination station e at the start of phase k
$n_{p,e}^{\text{absorb}}(k)$	Number of passengers absorbed by trains at line platform p with destination station e during phase k
$C_p(k)$	Total remaining capacity of trains visiting line platform p during phase k
$n_p^{\text{want}}(k)$	Total number of passengers who want to board trains at line platform p during phase k
$n_{p,e}^{\text{on-board}}(k)$	Number of passengers on board when trains arrive at line platform p with destination e during phase k
$n_{p,e}^{\text{alight}}(k)$	Number of passengers alighting from trains at line platform p with destination station e during phase k
$n_{p,q,e}^{\text{transfer}}(k)$	Number of passengers transferring from line platform p to line platform q with destination e during phase k
$n_{p,e}^{\text{trans,arrive}}(k)$	Number of transfer passengers arriving at line platform p with destination station e during phase k
$n_{p,e}^{\text{depart}}(k)$	Number of passengers departing from line platform p with destination station e during phase k
$f_p(k)$	Number of trains departing from line platform p during phase k
$\theta_z(k)$	The total number of trains available at depot z at the end of phase k
$y_{i,j,\ell,p}$	Binary variable; if train j departs from line platform p before train i departs from the depot related to line ℓ , $y_{i,j,\ell,p} = 1$; otherwise, $y_{i,j,\ell,p} = 0$
$\xi_{i,i',p,p'}$	Binary variable; if train i arrives at line platform p earlier than train i' at line platform p' , $\xi_{i,i',p,p'} = 1$; otherwise, $\xi_{i,i',p,p'} = 0$

Table 2.4: Output variables.

count passenger OD demands. The variables and parameters related to the number of passengers for the passenger absorption model are listed in Table 2.4. To illustrate the above variables, a general overview of these variables is presented in Fig. 2.2, which features a station with two line platforms, i.e., line platform p and line platform q . More details about the variables are introduced below.

A matrix is typically used to describe time-dependent passenger OD demands. Each entry of the matrix is represented by $\rho_{s,e}^{\text{station}}(t)$ where s and e are the origin and destination stations, respectively, and t represents time. Generally, $\rho_{s,e}^{\text{station}}(t)$ is a nonlinear time-varying function, and it would significantly increase the computational complexity of including passenger flows in train scheduling problems. Considering the periodic characteristic of passenger flows in urban rail transit systems, the planning time window is divided into a sequence of phases with length T , and each phase has constant passenger demands. The illustration for approximating time-dependent passenger arrival rates in the passenger absorption model is given in Fig. 2.3.

In urban rail transit networks (especially in large cities, such as London, Barcelona), different lines may use the same physical track and/or the same physical platforms to maximize the utilization of the infrastructure. To distinguish platforms for different lines and different directions, we introduce the definition of “(virtual) line platform”, where

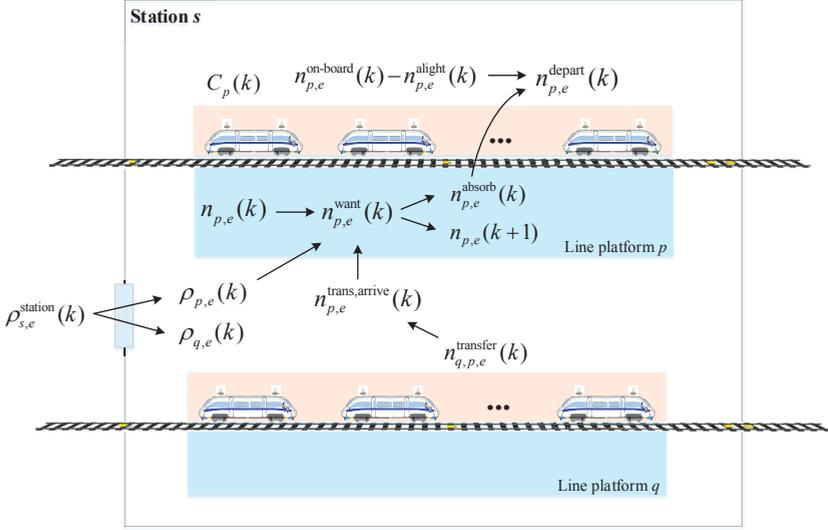


Figure 2.2: Variables for the model during phase k .

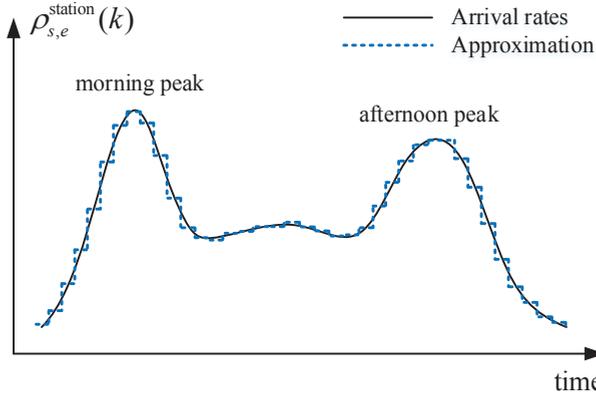


Figure 2.3: Illustration for approximating time-dependent passenger arrival rates.

each line platform is exclusively linked with one direction of one line. For example, in Fig. 2.4, Line 1 and Line 2 share the same physical platform B, and we regard platform B as two different line platforms. The safe operation at the line platforms is ensured by constraints (2.17), (2.28)-(2.31) below.

The arrival rate $\rho_{p,e}(k)$ for passengers at line platform $p \in \mathcal{P}$ with destination station $e \in \mathcal{S}$ in phase k is computed by

$$\rho_{p,e}(k) = \lambda_{s_p,p,e}(k) \rho_{s_p,e}^{\text{station}}(k), \quad (2.1)$$

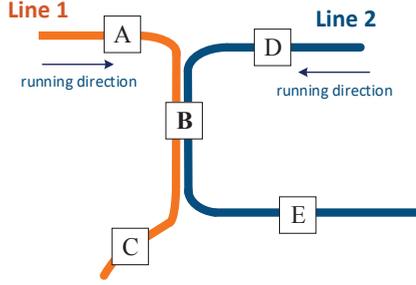


Figure 2.4: Illustration for the line platform concept.

where s_p represents the station corresponding to line platform p ; note that each line platform p is corresponding to only one station s_p ; $\lambda_{s_p,p,e}(k)$ denotes the splitting rate of passengers at station s_p who choose line platform p for their travel to destination e ; $\rho_{s_p,e}^{\text{station}}(k)$ denotes passenger origin-destination demand at phase k with s_p and e as the origin and destination stations, respectively; \mathcal{P} represents the set collecting all line platforms; \mathcal{S} is the set collecting all stations in the network.

At each line platform, the number of passengers evolves as:

$$n_{p,e}(k+1) = n_{p,e}(k) + \rho_{p,e}(k)T + n_{p,e}^{\text{trans,arrive}}(k) - n_{p,e}^{\text{absorb}}(k), \quad (2.2)$$

where $n_{p,e}(k)$ denotes the number of passengers stranded at line platform p with destination e at the start of phase k ; $n_{p,e}^{\text{trans,arrive}}(k)$ is the number of transfer passengers arriving at line platform p with destination e during phase k ; $n_{p,e}^{\text{absorb}}(k)$ denotes the number of passengers absorbed by trains at line platform p with destination e during phase k .

The variable $n_{p,e}^{\text{absorb}}(k)$ is estimated by

$$n_{p,e}^{\text{absorb}}(k) = \alpha_{p,e}(k)n_p^{\text{absorb}}(k), \quad (2.3)$$

where $\alpha_{p,e}(k)$ is the relative fraction of passengers boarding trains at line platform p during phase k in order to reach their destination station e , and $\alpha_{p,e}(k)$ can be estimated through the historical data; $n_p^{\text{absorb}}(k)$ denotes the total number of passengers boarding trains at line platform p during phase k , and we have

$$n_p^{\text{absorb}}(k) = \min\left(C_p(k), n_p^{\text{want}}(k)\right), \quad (2.4)$$

where $C_p(k)$ denotes the total remaining capacity provided by trains that visit line platform p during phase k , $n_p^{\text{want}}(k)$ is the total number of passengers that want to board trains at line platform p during phase k . Thus, we have

$$n_p^{\text{want}}(k) = n_p(k) + \rho_p(k)T + n_p^{\text{trans,arrive}}(k), \quad (2.5)$$

with

$$n_p(k) = \sum_{e \in \mathcal{S}} n_{p,e}(k), \quad \rho_p(k) = \sum_{e \in \mathcal{S}} \rho_{p,e}(k), \quad n_p^{\text{trans,arrive}}(k) = \sum_{e \in \mathcal{S}} n_{p,e}^{\text{trans,arrive}}(k). \quad (2.6)$$

The total remaining capacity of trains $C_p(k)$ at line platform p during phase k is determined by the maximum capacity of the trains, the number of passengers already on board the train, and the number of passengers alighting from the trains:

$$C_p(k) = f_p(k) \cdot C_{\text{train}} - \sum_{e \in \mathcal{S}} n_{p,e}^{\text{on-board}}(k) + \sum_{e \in \mathcal{S}} n_{p,e}^{\text{alight}}(k), \quad (2.7)$$

where $f_p(k)$ denotes the number of trains departing from line platform p during phase k , and $f_p(k)$ is the decision variable of the absorption model; C_{train} represents the maximum capacity of a train; $n_{p,e}^{\text{on-board}}(k)$ denotes the number of passengers with destination station e already on board the train when trains arrive at line platform p during phase k ; $n_{p,e}^{\text{alight}}(k)$ represents the number of passengers with destination station e alighting from trains at line platform p during phase k .

We define $p^{\text{pla}}(p)$ as the preceding line platform of line platform p , and $\bar{r}_{p^{\text{pla}}(p)}$ as the mean running time for trains from line platform $p^{\text{pla}}(p)$ to p . Then, the variable $n_{p,e}^{\text{on-board}}(k)$ in (2.7) is the number of passengers transported by trains from line platform $p^{\text{pla}}(p)$ to p during phase k with destination station e . As the length of the time step for the absorption model is T , and passengers departing from line platform $p^{\text{pla}}(p)$ require time $\bar{r}_{p^{\text{pla}}(p)}$ to reach line platform p , we have

$$n_{p,e}^{\text{on-board}}(k) = \frac{T - \bar{r}_{p^{\text{pla}}(p)}}{T} n_{p^{\text{pla}}(p),e}^{\text{depart}}(k) + \frac{\bar{r}_{p^{\text{pla}}(p)}}{T} n_{p^{\text{pla}}(p),e}^{\text{depart}}(k-1), \quad (2.8)$$

where $n_{p^{\text{pla}}(p),e}^{\text{depart}}(k)$ represents the number of passengers departing from line platform $p^{\text{pla}}(p)$ with destination e during phase k , and T and $\bar{r}_{p^{\text{pla}}(p)}$ are parameters of the model. As the developed model aims to address passenger demands within a relatively long time, we typically set $T \gg \bar{r}_{p^{\text{pla}}(p)}$. Note that if p is the first line platform of the line, we set $n_{p,e}^{\text{on-board}}(k) = 0$, which means the train is empty when arriving at the first line platform of a line.

The number of passengers $n_{p,q,e}^{\text{transfer}}(k)$ transferring from line platform p to line platform q with destination e during phase k , is calculated by

$$n_{p,q,e}^{\text{transfer}}(k) = \chi_{p,q,e} n_{p,e}^{\text{on-board}}(k), \forall q \in \text{cop}(p) \setminus \{p\}, \quad (2.9)$$

where $\text{cop}(p)$ defines a set collecting all line platforms located at the identical station as line platform p , $\chi_{p,q,e}$ refers to the proportion of passengers¹ transferring from line platform p to line platform q with destination e , which can be estimated according to the historical data, and $\sum_{q \in \text{cop}(p)} \chi_{p,q,e} = 1$.

At each line platform, passengers that either have transfer connections or have reached their destinations will alight from trains. Thus, the number of alighting passengers $n_{p,e}^{\text{alight}}(k)$ is computed by

$$n_{p,e}^{\text{alight}}(k) = \begin{cases} \sum_{q \in \text{cop}(p)} n_{p,q,e}^{\text{transfer}}(k), & \text{if } e \in \mathcal{S} \setminus \{\text{sta}(p)\}, \\ n_{p,e}^{\text{on-board}}(k), & \text{if } e = \text{sta}(p), \end{cases} \quad (2.10)$$

¹ $\chi_{p,p,e}$ represents the proportion of passengers remaining on trains at platform p .

where $\text{sta}(p)$ refers to the station corresponding to line platform p .

The number of departing passengers $n_{p,e}^{\text{depart}}(k)$ is computed by

$$n_{p,e}^{\text{depart}}(k) = n_{p,e}^{\text{on-board}}(k) - n_{p,e}^{\text{alight}}(k) + n_{p,e}^{\text{absorb}}(k), \quad (2.11)$$

which means that, at each line platform, some passengers will alight from trains while passengers waiting at the platform will board the trains before the trains depart from the platform.

As the basic time unit of the model is T , and the transfer passengers require time $t_{q,p}^{\text{transfer}}$ to reach line platform p , the number of transfer passengers arriving at line platform p . Then, $n_{p,e}^{\text{trans,arrive}}(k)$ can be computed by

$$n_{p,e}^{\text{trans,arrive}}(k) = \sum_{q \in \text{cop}(p) \setminus \{p\}} \left(\frac{T - t_{q,p}^{\text{transfer}}}{T} n_{q,p,e}^{\text{transfer}}(k) + \frac{t_{q,p}^{\text{transfer}}}{T} n_{q,p,e}^{\text{transfer}}(k-1) \right), \quad (2.12)$$

where $t_{q,p}^{\text{transfer}}$ denotes the mean time of transferring from line platform q to line platform p .

In this chapter, we address the train scheduling problem without disruptions. Thus, for each line, all trains will visit every pre-determined station along the line with the same stopping pattern. Let's define $\gamma_p(k)$ as the mean time of trains from a depot to line platform p . Define $\lfloor x \rfloor$ as the greatest integer less than or equal to x ; then, we define

$$\beta_p(k) = \left\lfloor \frac{\gamma_p(k)}{T} \right\rfloor, \quad (2.13)$$

$$\phi_p(k) = \gamma_p(k) - \beta_p(k)T, \quad (2.14)$$

where $\phi_p(k)$ denotes the remainder of $\frac{\gamma_p(k)}{T}$ with $0 \leq \phi_p(k) < T$. In this context, $\beta_p(k) \geq 0$ determines the number of phases required for trains from the depot to line platform p .

The departure frequency $f_p(k)$ of line platform p is determined by the departure frequency from the output link of the depot. As trains typically depart from depot and require $\gamma_p(k)$ to reach line platform p , $f_p(k)$ is determined by

$$f_p(k) = \frac{T - \phi_p(k)}{T} u_\ell(k - \beta_p(k)) + \frac{\phi_p(k)}{T} u_\ell(k - \beta_p(k) - 1), \quad p \in \mathcal{P}_\ell, \quad (2.15)$$

where $u_\ell(k)$ defines the departure frequency from the depot corresponding to line ℓ during period k ; \mathcal{P}_ℓ denotes set of line platforms of line ℓ .

The departure frequency determines the time interval between the departure times of two consecutive trains, thereby influencing the maximum waiting time of passengers. We define a lower bound for the departure frequency:

$$f_p(k) \geq f_{\min}, \quad (2.16)$$

where f_{\min} represents the minimum departure frequency. In this way, the time interval between the departures of two consecutive trains is always shorter than a given threshold. Thus, the maximum waiting time for passengers should still be acceptable in case the departure frequency and/or departure time change.

Remark 2.1. We assume that rolling stock resource is such that the minimum departure frequency constraint can always be satisfied. However, in case this assumption is dropped and the rolling stock resource is so limited that the minimum departure frequency constraint can be violated, then we can turn the minimum departure frequency constraint into a soft constraint.

To ensure safe operation, the number of trains departing from line platform p during phase k is constrained by

$$\sum_{p' \in \text{phy}(p)} f_{p'}(k)(h_p^{\min} + \tau_p^{\min}) \leq T, \quad (2.17)$$

where $\text{phy}(p)$ represents the set of line platforms using the same physical platform as line platform p ; h_p^{\min} and τ_p^{\min} are the minimum departure-arrival headway and the minimum dwell time at line platform p , respectively.

The rolling stock circulation determines the availability of trains for each line, which should be included in the optimization of train departure frequencies. In this chapter, we only consider the case that the depot is located at the end of each line, and the constraints for rolling stock circulation are

$$\theta_z(k) = \theta_z(k-1) + \sum_{p \in \text{in}(z)} f_p(k) - \sum_{\ell \in \text{out}(z)} u_\ell(k), \forall z \in \mathcal{Z} \quad (2.18)$$

$$\theta_z(k) \geq 0, \forall z \in \mathcal{Z}, \quad (2.19)$$

where z is the depot index, \mathcal{Z} is the set of depots, $\theta_z(k)$ represents the total number of trains available at depot z at the end of phase k , $\sum_{p \in \text{in}(z)} f_p(k)$ calculates the total number of trains entering depot z during phase k , $\text{in}(z)$ defines the set of line platforms corresponding to the entering link of depot z , $\sum_{\ell \in \text{out}(z)} u_\ell(k)$ calculates the total number of trains leaving depot k during phase k , and $\text{out}(z)$ defines the set of lines corresponding to the output link of depot z , $\theta_z(0) = N_z^{\text{train}}$ is a parameter representing the number of trains available at depot z .

Remark 2.2. Note that if $\theta_z(k) = 0$, depot z may need to wait for new arrivals. This effect is not included in the higher-level problem and may thus result in suboptimality for the final solution produced by the lower-level optimization problem.

2.4.3. TRAIN SCHEDULING MODEL

As indicated before, the upper level of the proposed bi-level framework determines the number of trains departing from the lines in the urban rail transit network. However, the exact departure and arrival times should be determined to obtain a practically implementable timetable. Therefore, a train scheduling model is introduced for the detailed timetable (including departure/arrival time and train orders), detailed rolling stock circulation, and train speed profiles.

There are typically three groups of constraints corresponding to the train operation, i.e., departure/arrival constraints, rolling stock circulation constraints, running time constraints, and headway constraints.

DEPARTURE/ARRIVAL CONSTRAINTS

The departure time $d_{i,p}$ of train i at line platform p is determined by:

$$d_{i,p} = a_{i,p} + \tau_{i,p}, \quad (2.20)$$

where $a_{i,p}$ and $\tau_{i,p}$ respectively denote arrival time and dwell time of train i at line platform p , and $\tau_{i,p}$ should satisfy

$$\tau_p^{\min} \leq \tau_{i,p} \leq \tau_p^{\max}, \quad (2.21)$$

where τ_p^{\min} and τ_p^{\max} denote the minimum and the maximum dwell times for trains at line platform p , respectively.

Define $p^{\text{pla}}(p)$ as the preceding line platform of line platform p , the arrival time $a_{i,p}$ of train i at line platform p is determined by:

$$a_{i,p} = d_{i,p^{\text{pla}}(p)} + r_{i,p^{\text{pla}}(p)}, \quad (2.22)$$

where $d_{i,p^{\text{pla}}(p)}$ denotes the departure time of train i at line platform $p^{\text{pla}}(p)$, $r_{i,p^{\text{pla}}(p)}$ is the running time of train i from line platform $p^{\text{pla}}(p)$ to line platform p .

Remark 2.3. If p is the first line platform of the line, for completeness, we set $d_{i,p^{\text{pla}}(p)} = d_{i,\ell}^{\text{depot}}$, where $d_{i,\ell}^{\text{depot}}$ represents the departure time of train i from the depot corresponding to line ℓ , and $r_{i,\ell}^{\text{depot}}$ is the running time of train i from the depot to the first line platform of the line, $p \in \mathcal{P}_\ell$.

ROLLING STOCK CIRCULATION CONSTRAINTS

Before sending a train from a depot, the availability of trains in the depot should be taken into account. Let us define a binary variable $y_{i,j,\ell,p}$ based on the departure time $d_{i,\ell}^{\text{depot}}$ of train i from the depot corresponding to line ℓ :

$$y_{i,j,\ell,p} = \begin{cases} 1, & \text{if } d_{j,p} \leq d_{i,\ell}^{\text{depot}}; \\ 0, & \text{otherwise.} \end{cases} \quad (2.23)$$

Then, the rolling stock circulation constraint at the lower level is

$$\sum_{\ell' \in \text{out}(z)} \sum_{j \in \mathcal{J}_{\ell'}} y_{i,j,\ell,p_{\ell'}} - \sum_{p \in \text{in}(z)} \sum_{j \in \mathcal{J}_p} y_{i,j,\ell,p} \leq N_z^{\text{train}}, \quad (2.24)$$

where $\mathcal{J}_{\ell'}$ defines the set of trains departing from the output link of the depot corresponding to line ℓ' with $p_{\ell'}$ being the corresponding departure platform, and \mathcal{J}_p defines the set of trains that depart from line platform p . In (2.24), the first term represents the total number of trains that have left depot z before train i departs, while the second term accounts for the total number of trains that have entered depot z prior to the departure of train i from the same depot.

RUNNING TIME CONSTRAINTS

Considering the operational requirement and speed limits, the running time constraint is

$$r_p^{\min} \leq r_{i,p} \leq r_p^{\max}, \quad (2.25)$$

where r_p^{\min} and r_p^{\max} are the minimum and maximum running times from line platform p to its succeeding line platform, respectively.

In general, $r_{i,p}$ is determined by train running speeds. In real life, train speeds and train running time between two stations are usually adjusted through an on-board train operation system, where different operation levels are defined, and each level corresponds to one speed profile option [152]. Therefore, we consider different train speed profile options for trains between two stations, and each option is related to a specific running time and a value of energy cost. In this context, the running time $r_{i,p}$ for train i is determined by

$$r_{i,p} = \sum_{b \in \mathcal{B}_{i,p}} x_{i,p,b} r_{i,p,b}, \quad (2.26)$$

where b denotes the train speed profile option index, $\mathcal{B}_{i,p}$ represents the set of speed profile options for train i at line platform p (for example, speed profile options in Fig. 2.5); $r_{i,p,b}$ denotes the running time corresponding to speed profile option b ; $x_{i,p,b}$ represents a binary variable indicating whether a speed profile is selected, i.e., $x_{i,p,b} = 1$ if speed profile option b is selected for train i at line platform p , otherwise, $x_{i,p,b} = 0$.

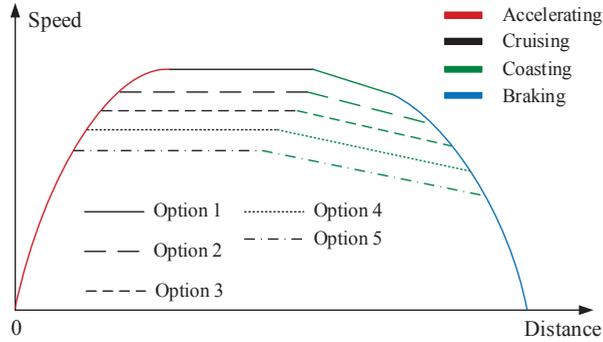


Figure 2.5: Illustration of different train speed profile options in a segment.

In order to ensure only one option can be selected, $x_{i,p,b}$ should satisfy

$$\sum_{b \in \mathcal{B}_{i,p}} x_{i,p,b} = 1. \quad (2.27)$$

In this chapter, different speed profiles can be calculated offline, and we only need to select one speed profile among different speed profile options in real time.

HEADWAY CONSTRAINTS

Headway is crucial for the safety of two consecutive trains, and for trains in the same line (see Fig. 2.6 (a)) we have:

$$a_{i,p} \geq d_{p^{\text{tra}}(i),p} + h_p^{\min}, \quad (2.28)$$

where $p^{\text{tra}}(i)$ represents the preceding train of train i at line ℓ , and h_p^{\min} represents the minimum departure-arrival headway at line platform p .

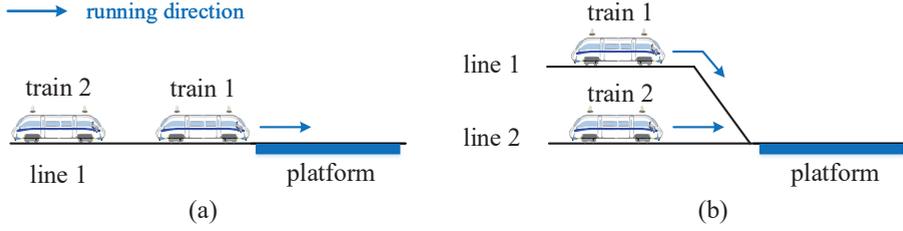


Figure 2.6: Different lines may use the same physical platform.

In urban rail transit networks (especially in large cities, such as London, Barcelona), different lines may use the same physical track and/or physical platforms to maximize the utilization of infrastructure (see Fig. 2.6 (b)). In this context, headway constraints for trains on different lines are required. We use a binary variable $\xi_{i,i',p,p'}$ to represent the order of trains from different lines:

$$\xi_{i,i',p,p'} = \begin{cases} 1, & \text{if } a_{i,p} \leq a_{i',p'}; \\ 0, & \text{otherwise.} \end{cases} \quad (2.29)$$

Then, the headway constraint for train i and train i' can be represented as

$$a_{i,p} - d_{i',p'} \geq h_p^{\min} - M_a(1 - \sigma_{p,p'} + \xi_{i,i',p,p'}), \quad (2.30)$$

where M_a represents a sufficiently large positive value. Eq. (2.30) represents the headway constraint of trains i and i' when line platforms p and p' are associated with the same physical platform, i.e., $\sigma_{p,p'} = 1$; otherwise, $\sigma_{p,p'} = 0$, then (2.30) holds automatically.

Furthermore, the order of trains should also satisfy

$$\xi_{i,i',p,p'} + \xi_{i',i,p',p} = 1, \quad (2.31)$$

which is employed to keep train order variables consistent, i.e., either $\xi_{i,i',p,p'} = 1$ or $\xi_{i',i,p',p} = 1$.

2.5. BI-LEVEL MPC FOR TRAIN SCHEDULING

MPC is an efficient real-time model-based control approach where finite-horizon optimization procedures are conducted repeatedly in a receding horizon scheme [91]. By dividing the long planning time window into several short time windows, MPC solves the problem with a short time window in a receding horizon manner to reduce the computational burden, while taking into account the real-time information of the urban rail transit network. A bi-level MPC approach is proposed to achieve real-time timetable scheduling in this section. The general introduction and the bi-level structure are introduced in Section 2.5.1. Then, the MPC approaches for both levels are presented in Section 2.5.2 and Section 2.5.3, respectively.

2.5.1. BI-LEVEL MPC FOR THE INTEGRATED PROBLEM

The bi-level control scheme is illustrated in Fig. 2.7 where passenger flow control and train scheduling are addressed at two different levels.

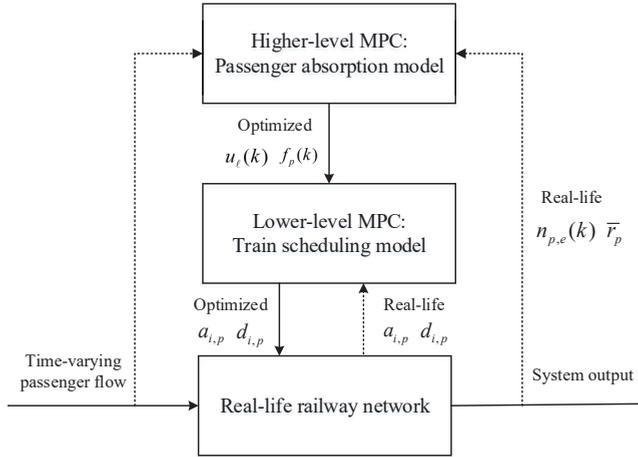


Figure 2.7: Bi-level control structure for the integrated problem.

As shown in Fig. 2.7, the higher level aims to address time-dependent passenger origin-destination (OD) demands by determining the number of trains departing from each line platform during each phase. The higher-level controller uses the passenger absorption model of Section 2.4.2. As we approximate time-dependent passenger OD demands as piecewise constants, the higher-level controller can be handled at every phase. Therefore, the higher-level controller can be conducted in relatively slow dynamics. Once the higher-level MPC optimization problem is solved, the optimized decision variables $f_p^*(k)$ are sent to the lower level. At the lower level, the train scheduling problem is solved to obtain the optimized arrival time $a_{i,p}^*$ and departure time $d_{i,p}^*$ for each train taking train speed profiles into account. The lower-level controller should be addressed with fast dynamics for real-time train scheduling so that the obtained arrival times, departure times, and train speed profiles can be implemented into the practical urban rail transit network.

In the bi-level MPC scheme, at the end of the control interval of the lower-level controller, the planning time window at the lower level will be moved for one step, and the train scheduling problem is resolved for the next step according to the collected real-life arrival and departure times ($a_{i,p}$ and $d_{i,p}$). At the end of the control interval of the higher-level controller (i.e., one phase), the planning time span for the higher level will be shifted for one phase, and the control problem will be solved again for the next phase based on the realized \bar{r}_p and $n_{p,e}(k)$.

2.5.2. HIGHER-LEVEL MPC: DEPARTURE FREQUENCY OPTIMIZATION

The time-dependent passenger OD demands can be addressed by a centralized MPC approach based on the model presented in Section 2.4.2. As passenger flows usually change periodically, the control time interval of the higher-level controller is equal to the length of a phase. The decision variable at the higher level will be the number of

trains departing from each line platform during each phase.

The total travel time for passengers during phase k is represented by

$$J^{\text{pass}}(k) = \sum_{p \in \mathcal{P}} \left(n_p(k)T + n_p^{\text{depart}}(k)\bar{r}_p + n_p^{\text{trans,arrive}}(k)t_p^{\text{transfer}} \right), \quad (2.32)$$

where \mathcal{P} defines a set collecting all line platforms in the urban rail transit network; $n_p(k)T$ denotes the passenger waiting time at line platform p during phase k ; $n_p^{\text{depart}}(k)\bar{r}_p$ denotes the total running time for passengers departing from line platform p during phase k ; $n_p^{\text{trans,arrive}}(k)t_p^{\text{transfer}}$ is the total transfer time for passengers at line platform p during phase k , and t_p^{transfer} denotes the average time for passengers transferring to line platform p .

Although scheduling more trains, running with the minimum headway, can help to minimize $J^{\text{pass}}(k)$, it is typically not acceptable to use too many trains in real life, as it would significantly increase the total energy consumption. Thus, a penalty term corresponding to train energy consumption is included in the cost function. Then, the MPC optimization problem for passenger flow control at phase k_0 can be represented by

$$\begin{aligned} \min_{\mathbf{u}(k)} J^{\text{high}} &= \sum_{k=k_0}^{k_0+N-1} \left(J^{\text{pass}}(k) + \eta \sum_{p \in \mathcal{P}} f_p(k)\bar{E}_p \right) + L_N(k_0) \\ \text{s.t.} \quad & (2.1) - (2.12), (2.15) - (2.19), \end{aligned} \quad (2.33)$$

where N denotes the number of phases in the prediction time span; η represents a weight; \bar{E}_p denotes the average energy consumption for a train running from the line platform p to its succeeding line platform, since the higher level does not know which speed profile will be selected at the lower level when solving the high-level optimization problem, we use the average value among all speed profile options in the high-level optimization problem; and $\mathbf{u}(k)$ collects the independent decision variables, i.e., the departure frequency at the depot corresponding to each line $u_\ell(k)$; $L_N(k_0)$ is a penalty term for the passengers that can not board trains at the end of the prediction window, and in this chapter we set $L_N(k_0) = \sum_{p \in \mathcal{P}} n_p(k_0 + N)T$. As stated in (2.15), the departure frequencies of other line platforms are determined by $u_\ell(k)$.

In each MPC step, problem (2.33) is a nonlinear nonconvex optimization problem. By using the properties in [139], we can convert the nonconvex term (2.4) into linear constraints.

Transformation property 2.4. If we introduce a binary variable $\delta_{k,p}^{\text{absorb}}$ and an auxiliary real variable $f_{k,p}^{\text{absorb}}$ with $f_{k,p}^{\text{absorb}} = n_p^{\text{want}}(k) - C_p(k)$. Then, if we define m_p and M_p as the minimum and the maximum values of $f_{k,p}^{\text{absorb}}$, respectively, the expression $\delta_{k,p}^{\text{absorb}} = 1 \Leftrightarrow f_{k,p}^{\text{absorb}} \leq 0$ is equivalent to

$$\begin{cases} f_{k,p}^{\text{absorb}} \leq M_p (1 - \delta_{k,p}^{\text{absorb}}), \\ f_{k,p}^{\text{absorb}} \geq \varepsilon + (m_p - \varepsilon) \delta_{k,p}^{\text{absorb}}, \end{cases} \quad (2.34)$$

where ε represents a sufficiently small number. Then, (2.4) can be replaced by

$$n_p^{\text{absorb}}(k) = \delta_{k,p}^{\text{absorb}} n_p^{\text{want}}(k) + (1 - \delta_{k,p}^{\text{absorb}}) C_p(k). \quad (2.35)$$

Transformation property 2.5. The multiplication of real variable \bar{y} and logical variable $\bar{\delta}$ can be replaced by an auxiliary real variable, with

$$\bar{g} = \bar{y} \cdot \bar{\delta}. \quad (2.36)$$

Then, $\bar{g} = \bar{y} \cdot \bar{\delta}$ can be exactly transformed into

$$\begin{cases} \bar{g} \leq M_{\bar{y}} \bar{\delta}, \\ \bar{g} \geq m_{\bar{y}} \bar{\delta}, \\ \bar{g} \leq \bar{y} - m_{\bar{y}}(1 - \bar{\delta}), \\ \bar{g} \geq \bar{y} - M_{\bar{y}}(1 - \bar{\delta}), \end{cases} \quad (2.37)$$

where $M_{\bar{y}}$ and $m_{\bar{y}}$ respectively represent the maximum and minimum values of \bar{y} .

By using the above transformations, problem (2.33) can be exactly converted to an MILP problem with the following form:

$$\begin{aligned} \min_{\substack{\mathbf{x}(k), \mathbf{u}(k) \\ \boldsymbol{\delta}(k), \mathbf{z}(k)}}} J^{\text{high}} &:= \sum_{k=k_0}^{k_0+N-1} \left(J^{\text{pass}}(k) + \eta \sum_{p \in \mathcal{P}} f_p(k) \bar{E}_p \right) + L_N(k_0) \\ \text{s.t. } \mathbf{x}(k+1) &= A_k \mathbf{x}(k) + B_{1,k} \mathbf{u}(k) + B_{2,k} \boldsymbol{\delta}(k) + B_{3,k} \mathbf{z}(k), \\ D_{2,k} \boldsymbol{\delta}(k) + D_{3,k} \mathbf{z}(k) &\leq D_{1,k} \mathbf{u}(k) + D_{4,k} \mathbf{x}(k) + D_{5,k}, \\ k &= k_0, \dots, k_0 + N - 1, \end{aligned} \quad (2.38)$$

where $\mathbf{x}(k)$ collects the output variables in phase k ; $\boldsymbol{\delta}(k)$ and $\mathbf{z}(k)$ collect the auxiliary binary and auxiliary continuous variables in phase k , respectively; $\mathbf{x}(k+1) = A_k \mathbf{x}(k) + B_{1,k} \mathbf{u}(k) + B_{2,k} \boldsymbol{\delta}(k) + B_{3,k} \mathbf{z}(k)$ includes all equality constraints in (2.1)-(2.12), (2.15), and (2.18); $D_{2,k} \boldsymbol{\delta}(k) + D_{3,k} \mathbf{z}(k) \leq D_{1,k} \mathbf{u}(k) + D_{4,k} \mathbf{x}(k) + D_{5,k}$ includes all inequality constraints.

Remark 2.6 (Complexity Analysis). There are three categories of variables in (2.38), i.e., continuous variables, binary variables, and auxiliary continuous variables. The constraints include linear and nonlinear constraints. The total numbers of variables and constraints are listed in Table 2.5, where \mathcal{S} , \mathcal{P} , and \mathcal{L} are the set of stations, line platforms, and lines, respectively, and $|\cdot|$ denotes the cardinality of a set.

Table 2.5: Numbers of variables and constraints in problem (2.38)

Variables or constraints	Maximal possible total number
Continuous variables	$(7 \cdot \mathcal{S} + 6) \cdot N \cdot \mathcal{P} $
Binary variables	$N \cdot \mathcal{P} $
Auxiliary continuous variables	$3 \cdot N \cdot \mathcal{P} $
Constraints	$(8 \cdot \mathcal{S} + 16) \cdot N \cdot \mathcal{P} $

It can be observed from Table 2.5 that the number of variables depends on the scale of the considered urban rail transit network and the prediction horizon N . The MILP problem is an NP-hard problem, and the computation time for solving the problem typically increases rapidly when the number of integer variables increases [40]. In this problem, the number of binary variables is determined by the number of lines $|\mathcal{L}|$, the number of line platforms $|\mathcal{P}|$, and the prediction horizon N . A large prediction horizon N can include more information in the train departure frequency optimization, while the computational burden increases. Therefore, for a given urban rail transit network, choosing

a proper prediction horizon is important to balance the computation time versus the performance.

Solving problem (2.38) results in a series of decision variables from phase k_0 to $k_0 + N - 1$, and according to the MPC paradigm, only the variables for phase k_0 are applied. In the next phase, the prediction time span is shifted for one phase, and a new optimization problem can be obtained.

Lemma 2.7 (*Recursive Feasibility*). If problem (2.38) is feasible at phase k_0 with initial state $\mathbf{x}(k_0)$, then the feasibility of problem (2.38) at phase $k_0 + 1$ can also be ensured.

Proof. The proof is based on finding a feasible solution for phase $k_0 + 1$. At phase k_0 with initial state $\mathbf{x}(k_0)$, problem (2.38) can be solved and the optimized decision variables are collected in $\mathbf{U}(k_0)$ with

$$\mathbf{U}(k_0) = [(\mathbf{u}^*(k_0))^\top, (\mathbf{u}^*(k_0 + 1))^\top, \dots, (\mathbf{u}^*(k_0 + N - 1))^\top]^\top, \quad (2.39)$$

where $\mathbf{u}^*(k_0)$ is the optimized value of $\mathbf{u}(k_0)$ for solving problem (2.38). By implementing the first decision variable $\mathbf{u}^*(k_0)$, we get

$$\mathbf{x}^*(k_0 + 1) = A_{k_0} \mathbf{x}(k_0) + B_{1,k_0} \mathbf{u}^*(k_0) + B_{2,k_0} \boldsymbol{\delta}^*(k_0) + B_{3,k_0} \mathbf{z}^*(k_0). \quad (2.40)$$

As we only have input constraint (2.17) at the higher level, and the inequalities constraints introduced in *Transformation property 2.4* and *Transformation property 2.5* are equivalent transformations for the mixed logical dynamical (MLD) model, a feasible solution for phase $k_0 + 1$ can always be found as

$$\mathbf{U}(k_0 + 1) = [(\mathbf{u}^*(k_0 + 1))^\top, \dots, (\mathbf{u}^*(k_0 + N - 1))^\top, (\mathbf{u}(k_0 + N))^\top]^\top, \quad (2.41)$$

where $\mathbf{u}^*(k_0 + 1), \dots, \mathbf{u}^*(k_0 + N - 1)$ are from solution $\mathbf{U}(k_0)$ at phase k_0 , and $\mathbf{u}(k_0 + N)$ can be any solution that satisfies (2.17), e.g., the corresponding value of the regular timetable. Hence, the recursive feasibility of the higher-level MPC problem is guaranteed. \square

2.5.3. LOWER-LEVEL MPC: TRAIN SCHEDULING

Based on the number of trains departing from each line platform obtained from the higher-level controller, the detailed timetable considering the energy consumption can be generated at the lower level. The lower level uses the train scheduling model introduced in Section 2.4.3, and the decision variables are departure/arrival times and train speed profile options of trains. As the lower-level controller aims to generate a practically implementable timetable considering real-time information of the network, the lower-level controller should be addressed with relatively fast dynamics.

According to Section 2.4.3, the energy consumption $E_i(p)$ for train i from line platform p to its succeeding line platform is determined by

$$E_i(p) = \sum_{b \in \mathcal{B}_{i,p}} x_{i,p,b} E_{i,b}(p), \quad (2.42)$$

where $E_{i,b}(p)$ denotes the energy consumption of speed profile option b for train i from line platform p to its succeeding line platform.

Generally, the energy consumption of a train in a segment is highly related to the running time, i.e., a longer running time (and thus a lower speed) typically leads to lower energy consumption. Furthermore, a penalty term has been to ensure consistency between the desired departure frequency and the departure times of trains, promoting an even spread of departures as much as possible. We define ϑ as the index for the control step of the lower level, where the time interval of each step is R . Then, the objective function for the lower-level controller is defined as

$$J^{\text{low}} = \sum_{i \in \mathcal{S}(k, \vartheta)} \sum_{p \in \mathcal{V}_i} \left(E_i(p) + \zeta \left| \frac{T}{u_\ell(k)} - (d_{i,p} - d_{i-1,p}) \right| \right), \quad (2.43)$$

where $\mathcal{S}(k, \vartheta)$ denotes the set of indices for trains leaving their first line platforms before the end of phase k but have not yet reached their destination at time step ϑ , \mathcal{V}_i denotes the set of line platforms that train i will visit, and ζ is a weighting factor.

The optimization problem for train scheduling at the lower level is

$$\begin{aligned} \min_{\mathbf{g}(k, \vartheta)} J^{\text{low}} := & \sum_{i \in \mathcal{S}(k, \vartheta)} \sum_{p \in \mathcal{V}_i} \left(E_i(p) + \zeta \left| \frac{T}{u_\ell(k)} - (d_{i,p} - d_{i-1,p}) \right| \right), \\ \text{s.t. } & (2.20) - (2.31), (2.42), \end{aligned} \quad (2.44)$$

where $\mathbf{g}(k, \vartheta)$ collects the decision variables for trains in set $\mathcal{S}(k, \vartheta)$, i.e., $a_{i,p}$, $d_{i,p}$, and $x_{i,p,b}$, $\forall i \in \mathcal{S}(k, \vartheta)$, $p \in \mathcal{V}_i$, $b \in \mathcal{B}_{i,p}$. Problem (2.44) contains piecewise constant (“if-then”) constraints in (2.29), which can be reformulated by using the property developed in [7] (see *Transformation property 2.8* below). Therefore, Problem (2.44) can also be transformed into an MILP problem.

Transformation property 2.8. If we define m_a and M_a as the minimum and maximum values of $a_{i,p}$, respectively, then (2.29) is equivalent to the following inequalities

$$\begin{cases} a_{i,p} - a_{i',p'} \leq (1 - \xi_{i,i',p,p'}) (M_a - a_{i',p'}), \\ a_{i,p} - a_{i',p'} \geq \varepsilon + \xi_{i,i',p,p'} (m_a - a_{i',p'} - \varepsilon). \end{cases} \quad (2.45)$$

In the MPC scheme, we solve the optimization problem (2.44) in a receding horizon way, which enables the decision-making process to include real-time information from the urban rail transit network. Solving problem (2.44) leads to a series of decision variables for all trains $i \in \mathcal{S}(k, \vartheta)$ from their current line platforms to their terminal line platforms. Only the decision variables pertaining to the first interval are executed, following which the prediction window is shortened by one step, and a new problem is formulated considering the newly collected information. The procedure is repeated until the last train in set $\mathcal{S}(k, \vartheta)$ arrives at its terminal line platform.

In this chapter, the lower-level controller optimizes the timetable of trains that have not yet reached their destination at phase k . As each train operates from its starting line platform to its terminal line platform, the MPC optimization is terminated until the last planned train arrives at its terminal platform. Therefore, the lower-level controller can be solved in a shrinking horizon manner, i.e., the end of the prediction horizon is fixed and equal to the arrival time of the last train in set $\mathcal{S}(k, \vartheta)$ at its terminal line platform.

Lemma 2.9 (Recursive Feasibility). Given a feasible solution of problem (2.44) at time step ϑ for trains in the set $\mathcal{S}(k, \vartheta)$ and line platforms in the set \mathcal{V}_i , a feasible solution for time step $\vartheta + 1$ can always be found.

Proof. For trains that have not departed from their depot at the current phase, a feasible solution of problem (2.44) can always be found by keeping trains at the depot. For trains that have already departed from their first line platform, a feasible solution for time step $\vartheta + 1$ can be found by keeping the solutions (i.e., $a_{i,p}$, $d_{i,p}$, $r_{i,p}$, $\forall i \in \mathcal{I}(k, \vartheta)$, $\forall p \in \mathcal{V}_i$) of the time step ϑ unchanged. In this context, the recursive feasibility of lower-level MPC can be guaranteed. \square

In the proposed method, both the higher level and the lower level use centralized MPC. We define the first step of the lower-level controller is indexed by $\vartheta_0(k)$ and the procedure of bi-level MPC for the integration of passenger flows, timetables, and train speed profiles is shown in *Algorithm 1*.

Algorithm 1 Bi-level MPC for the integrated problem

Input: k_{\max} , $\vartheta_{\max}(k)$; initial estimate for the variables γ_p , \bar{r}_p ;

Output: optimized values $a_{i,p}$, $d_{i,p}$

- 1: $k \leftarrow k_0$
 - 2: **repeat**
 - 3: $\vartheta \leftarrow \vartheta_0(k)$
 - 4: solve the higher-level problem (2.38), get $u_\ell(k)$ and $f_p(k)$
 - 5: **repeat**
 - 6: solve problem (2.44), get $a_{i,p}$ and $d_{i,p}$
 - 7: implement $a_{i,p}$ and $d_{i,p}$ to real-life network
 - 8: $\vartheta \leftarrow \vartheta + 1$
 - 9: collect real-life value of $a_{i,p}$, $d_{i,p}$, and $n_{p,e}(k)$
 - 10: **until** $\vartheta = \vartheta_{\max}(k)$
 - 11: $k \leftarrow k + 1$
 - 12: calculate real-life values of γ_p , \bar{r}_p
 - 13: **until** $k = k_{\max}$
-

In the developed bi-level MPC approach, the MPC optimization problems of both levels can be transformed into MILP problems by using the methods introduced in [7] and [139]. Therefore, we can derive an MILP problem at each level that is an exact equivalence of the original optimization problem. Furthermore, with existing MILP solvers, the resulting optimization problems can be solved.

2.6. CASE STUDY

This section involves conducting simulations to demonstrate the efficacy of the proposed passenger absorption model and bi-level control approach. Firstly, we introduce the urban rail transit network and the basic setup utilized in the case study. Subsequently, we evaluate the passenger absorption model based on real-life data from the Beijing urban rail transit network. Finally, simulations are conducted to assess the performance of the developed bi-level framework and bi-level MPC approach.

2.6.1. BASIC SETUP

In this chapter, we carry out the case study based on the real-life passenger flow data from the Beijing urban rail transit network. The network is displayed in Fig. 2.8, which is generated according to the northern part of the Beijing urban rail transit network. The network includes six bidirectional lines and 54 stations. Moreover, the network contains seven transfer stations, i.e., Station ZXZ, Station XEQ, Station HY, Station OP, Station WJX, Station LSQ, and Station DD, where passengers can transfer from one line to another to reach their destinations. Transfer passengers are defined as passengers whose route consists of more than one line.

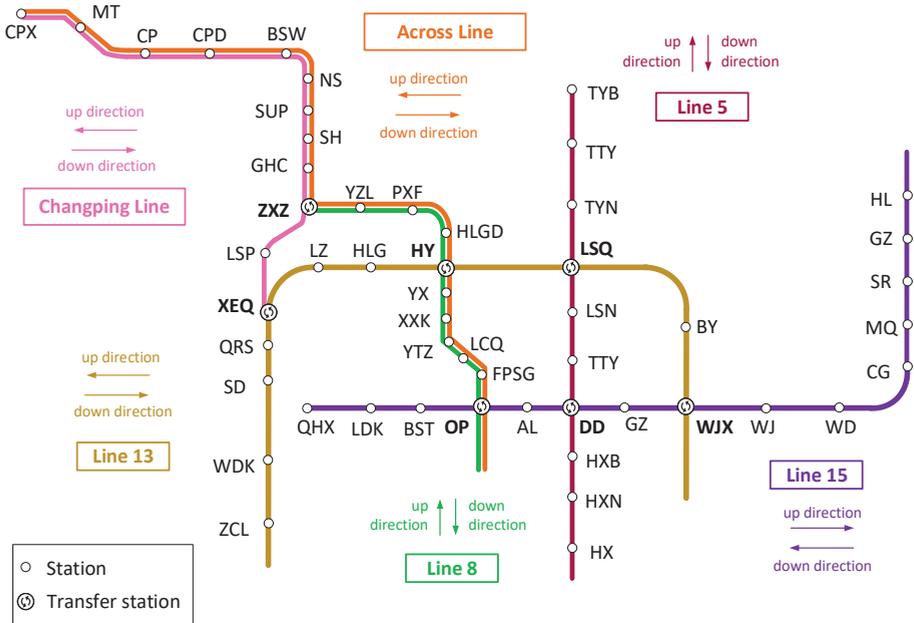


Figure 2.8: Layout of the considered urban rail transit network.

The across-line operation is one important way to maximize the utilization of infrastructure and to improve passenger satisfaction by reducing the number of transfer activities in the network (especially in big cities like London, Barcelona, and Beijing²). Therefore, we add an “Across Line” for the case study to meet the case when different lines use the same physical track and/or platforms. Some passengers at the Across Line (e.g. from CPD to PXF) can use the Across Line to reach their destination and transfer actions are not required anymore, so they are not considered to be transfer passengers. There are five lines in Fig. 2.8, where Changing Line, Line 8, Line 13, and Line 15 are the real-life lines, and the Across Line is added in this chapter to simulate the case of

²Beijing Subway plans to achieve the across-line operation among several lines in recent years, including the across-line operation of Changing Line and Line 8 in Fig. 2.8; see also <http://bj.people.com.cn/n2/2022/0126/c233088-35113072.html>

cross-line operation. The Across Line uses the same physical platforms as Changping Line from Station CPX to Station GHC, and the same physical platforms as Line 8 from Station ZXZ to Station OP.

Parameters	Line 8	Line 13	Line 15
	Changping Line Across Line		
Minimum departure-arrival headway	120 s	120 s	120 s
Regular departure-arrival headway	480 s	180 s	240 s
Maximum dwell time τ_p^{\max}	360 s	360 s	360 s
Minimum dwell time τ_p^{\min}	30 s	30 s	30 s
Regular dwell time $\tau_{i,p}$	60 s	60 s	60 s
Maximum capacity of a train C_{train}	2400 persons	2400 persons	2400 persons
Average transfer time t_p^{transfer}	60 s	60 s	60 s
Phase time T	1800 s	1800 s	1800 s
Number of speed profile options	8 options	8 options	8 options

Table 2.6: Parameters for the simulations

The passenger OD data are generated according to the real-life passenger flow data, i.e., the entering and exiting flow data of the Beijing urban rail transit network. This information is updated every 30 minutes. The data we use is for the morning peak hours from 7:00AM. The prediction time window is 1 hour. In the case study, we include the case when different lines use the same physical platforms, and the order of trains from different lines at the same physical platform can be adjusted. Table 2.6 presents the main parameters for the simulation. The parameters are generated based on the real-life timetable of the Beijing urban rail transit network. For the different speed profile options, we calculate the speed profile according to the method in [135] with the maximum acceleration of 0.8 m/s^2 , the maximum deceleration of 0.75 m/s^2 , and cruising speeds as 8 equidistant values in [65, 79] km/h, i.e., 65 km/h, 67 km/h, 69 km/h, 71 km/h, 73 km/h, 75 km/h, 77 km/h, 79 km/h, respectively. The length between every two consecutive stations is openly accessible on the website of Beijing Subway³. As the Across Line is not yet included in the historical data, in the basic timetable, trains of the Across Line and trains of Changping Line (or Line 8) depart alternately, which means part of the transport capacity that was originally performed by Changping Line (or Line 8) is taken over by the Across Line to reduce the number of transfer actions of passengers, and that change does not affect the total transport capacity or the number of trains needed for the basic timetable. Thus, the original OD demand is divided equally over two lines, so for the basic timetable, half of the departures of the original timetable is then arranged to Changping Line (or Line 8), while the other half is arranged to Across Line. This also means that the total number of trains in the network and the depot does not have to be changed compared with the original timetable. The simulation is coded using MATLAB (R2019b) on an Intel Xeon W2223 CPU (3.60 GHz) with 8GB RAM. In this chapter, we assume passengers' route choices are given a priori, and we consider passengers will choose the route with the shortest travel time for their travel.

³<https://www.bjsubway.com/station/zjgls/>

As far as we know, there is no well-recognized micro-simulator currently available that includes timetables, passenger OD demands, and train speeds. The model developed by [136] is the most elaborate model we noticed in the literature; thus, we use the model of [136] as the “accurate model” of the practical passenger dynamics in the railway network. The passenger absorption model combined with the train scheduling model presented in Section 2.4 are used as prediction models for the train scheduling problem. The basic timetable is generated by using the regular headway and the regular dwell time given in Table 2.6.

2.6.2. ASSESSMENT OF THE ABSORPTION MODEL

As mentioned in Section 2.6.1, we select the “accurate model” developed by [136] as the benchmark to assess the passenger absorption model. Instead of focusing on the specific times of train arrivals and departures, the passenger absorption model deals with the train departure frequencies in each phase. Thus, we regard the number of passengers as a function of the phase index rather than as a function of time.

The accumulated number of waiting passengers (AWP) and the accumulated number of boarding passengers (ABP) in each line are two main variables in passenger-oriented urban rail transit networks. In particular, AWP reflects whether passengers can board trains in time, since if passengers are unable to board trains in the current phase, they should wait for trains in the next phase. ABP reflects the transport capacity of trains.

The simulations are conducted on the network in Fig. 2.8 based on both the developed model and the “accurate model” of [136]. We perform the simulation from 7:00 to 15:00 which includes both peak and off-peak hours. We collect the AWP and ABP values in each phase. The required simulation time for the developed model and the accurate model are 2.10 s and 84.24 s, respectively. The relative differences between the absorption model and the “accurate model” for AWP and ABP of each line are displayed in Table 2.7. The simulation contains 16 phases, and we select the minimum, maximum, and final values of the relative difference among the phases at each line.

Table 2.7: Relative differences of variables for each line

	min		max		average	
	AWP	ABP	AWP	ABP	AWP	ABP
Changping Line	1.91 %	5.12%	7.74 %	19.38%	4.76 %	7.69%
Line 13	0.50 %	0.13%	22.15 %	25.79%	7.81 %	3.78%
Line 8	1.24 %	4.28%	20.48 %	33.55%	8.73 %	8.66%
Across Line	0.09 %	6.32%	17.82 %	17.54%	3.53 %	9.14%
Line 15	0.61 %	0.16%	27.71 %	32.19%	5.61 %	5.12%
Line 5	0.07 %	6.01%	17.12 %	19.76%	2.99 %	8.96%

It can be observed from Table 2.7 that Line 8 has the largest average relative difference for AWP, while the largest average relative difference for the ABP value occurs in Line 15. We select Line 8 and Line 15 for visualization, and the corresponding values for AWP and ABP at each time step are respectively depicted in Fig. 2.9 and Fig. 2.10.

Compared with the accurate model, the passenger flows can be modeled by the absorption model with the largest final error of about 10% and the required simulation

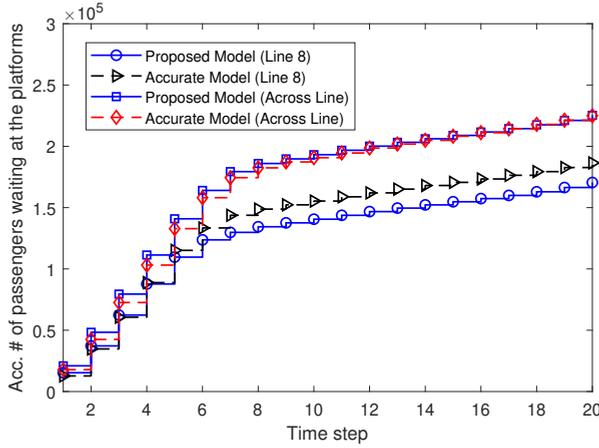


Figure 2.9: Accumulated number of passengers waiting at the platforms in each phase (AWP).

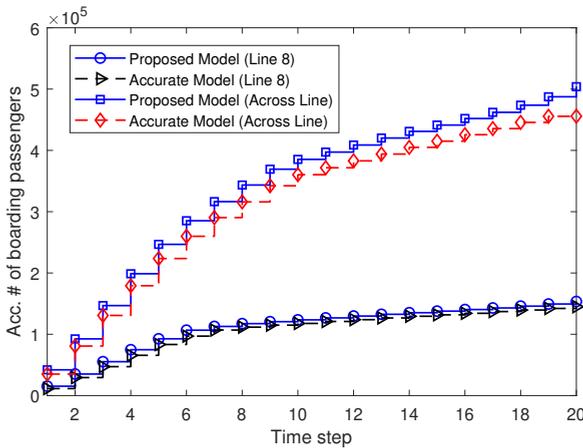


Figure 2.10: Accumulated number of boarding passengers in each phase (ABP).

time drops with a factor of around 40. Hence, we can conclude that with an acceptable accuracy loss, the absorption model can simulate passenger flows much more efficiently with time-dependent passenger OD demands, which allows more efficient methods for passenger-oriented train scheduling problems. The major loss is that the developed model does not include detailed arrival and departure times of trains, and thus a train scheduling model in the lower level is required to determine the specific departure and arrival times of trains.

2.6.3. BI-LEVEL OPTIMIZATION BASED ON THE ABSORPTION MODEL

We first perform simulations of sequentially solving optimization problems at both levels based on the developed model. We also use the single-level optimization approach to solve the integrated problem in a centralized manner. Then, we compare the single-level approach with the proposed bi-level approach based on solution quality and solution time. The single-level optimization problem is a nonlinear nonconvex problem containing integer variables. Compared with the bi-level optimization problem, the single-level counterpart introduces an additional nonlinear term, namely $\frac{T}{u_\ell(k)}$, in (2.43). The single-level optimization problem can also be converted to an MILP problem by using the method in [139]. We use the `gurobi` to solve all MILP problems. For the single-level approach, the nonlinear function $\frac{T}{u_\ell(k)}$ is approximated as a piecewise linear function by setting several breakpoints. However, setting more breakpoints can lead to a more accurate approximation of the nonlinear term, while more computation time is required for solving the resulting MILP problem. Therefore, in the case study, we use both one breakpoint and four breakpoints for the approximation of the nonlinear term in the single-level optimization problem, and for simplicity, the corresponding approaches are called single-level-1-brk and single-level-4-brk, respectively.

Table 2.8: Simulation results of different approaches in two cases

Case	Method	Objective function	CPU time (s)
Unsaturated case	Basic timetable	$8.3925 \cdot 10^3$	-
	Single-level-1-brk	$7.5520 \cdot 10^3$	3106.1
	Single-level-4-brk	$7.5339 \cdot 10^3$	7200.0
	Bi-level approach	$7.5903 \cdot 10^3$	40.5
Over-saturated case	Basic timetable	$9.5186 \cdot 10^3$	-
	Single-level-1-brk	$9.1386 \cdot 10^3$	5250.7
	Single-level-4-brk	$9.1027 \cdot 10^3$	7200.0
	Bi-level approach	$9.1119 \cdot 10^3$	87.0

We evaluate the developed approach in both the over-saturated (i.e., peak hours) and the unsaturated (i.e., off-peak hours) cases. For comparison, both single-level-1-brk and single-level-4-brk are also applied to solve the optimization problem. As our aim is to generate a timetable online, it is required to check whether an approach is real-time implementable. In the case study, the time limit for each method is set to be 7200 s, which is larger than the length of a step (1800 s) because we want each method to have sufficient time to find its solution, and we can compare the relative time of different methods. By using the regular dwell time and departure-arrival headway in Table 2.6, we can obtain a basic timetable.

The simulation results and CPU times of solving the problem for one step are presented in Table 2.8. The objective for comparison is the weighted sum of the total passenger travel time and the total energy consumption based on the simulation model. In both the unsaturated case and the over-saturated case, the simulation results indicate that single-level-4-brk performs slightly better than single-level-1-brk with regard to the objective function value. However, the CPU time of single-level-4-brk increases significantly as more integer variables are introduced when adding more breakpoints.

As real-time feasibility is important for real-time train scheduling, single-level-1-brk is more suitable for real-life applications than single-level-4-brk.

Compared to the basic timetable, the single-level-1-brk approach, single-level-4-brk approach, and bi-level approach exhibit a performance improvement of 10.01%, 10.23%, and 9.56%, respectively, in the unsaturated case, while the improvement for the over-saturated case is 3.99%, 4.37%, and 4.27%, respectively. The bi-level approach can find its optimal solution very quickly. The CPU times of single-level-1-brk and single-level-4-brk are much larger than the bi-level approach, which implies that single-level optimization may not be a suitable option for real-time train scheduling of large-scale urban rail transit networks. The results thus show that the bi-level optimization approach can achieve a balanced trade-off between the solution quality and the computation time.

2.6.4. BI-LEVEL MPC FOR REAL-TIME TRAIN SCHEDULING

In this section, we conduct the case study under the MPC scheme to illustrate the closed-loop performance and the real-time feasibility of the developed approach. The prediction time window of MPC is one hour.

As shown in Section 2.6.3, the single-level-1-brk approach requires less computation time than single-level-4-brk with an acceptable sacrifice of performance. Considering the real-time feasibility of approaches, we select the single-level-1-brk approach to solve the optimization problems of single-level MPC. The maximum solution time for the MPC optimization problem in each step is set to be 7200 s. The simulation results of single-level MPC and bi-level MPC are displayed in Table 2.9 and Fig. 2.11, where the objective function value means the accumulated objective function value for all included simulation times. The performance of the basic timetable is also given for comparison.

Table 2.9: Comparison of different approaches for real-time train scheduling

Method	Objective function	CPU time (s)	
		t_{avg}	t_{max}
Basic timetable	$1.4859 \cdot 10^5$	-	-
Single-level MPC	$1.2451 \cdot 10^5$	3181.5	7200.0
Bi-level MPC	$1.1815 \cdot 10^5$	42.4	95.9

The simulation results indicate that, compared with the basic timetable, bi-level MPC can improve the overall performance, i.e., the objective function value, by 20.49%, while the improvement of single-level MPC is 16.21%. The average computation time for single-level MPC is 3181.5 s. Due to the time limit, single-level MPC cannot always obtain its optimal solution within the given maximum solution time in every MPC step, which influences the solution quality of single-level MPC. The average and maximum solution times of bi-level MPC are 42.4 s and 95.9 s, respectively. Simulation results indicate that bi-level MPC can compute its optimal solution within an acceptable time. However, single-level MPC is not efficient in terms of computation time, and as a result, single-level MPC may not be suitable for real-time implementation in large-scale networks.

For further illustration, the number of trains departing from the first line platform of Line 5 (down direction) is shown in Fig. 2.12 as an example. As time steps 1-6 correspond to the morning peak hours from 7:00AM - 10:00AM, compared with the basic timetable,

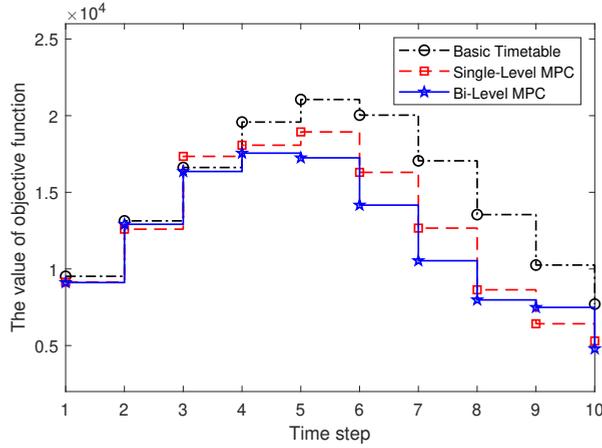


Figure 2.11: Comparison of different approaches for real-time train scheduling.

more trains are scheduled with the single-level and the bi-level MPC approaches to address the large passenger demand, which indicates that bi-level MPC is able to optimize the number of trains departing from each line according to the time-dependent passenger demands.

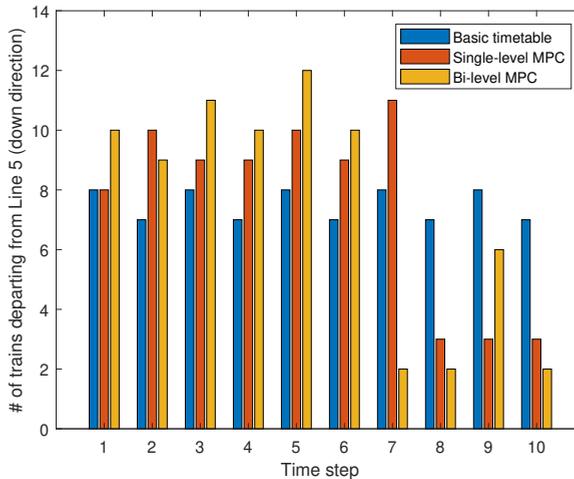


Figure 2.12: Number of trains departing from the first line platform of Line 5 (down direction) at each time step.

We select Line 5 (down direction) as a representative line to show the timetables generated by different approaches. The basic timetable of the morning peak hour from

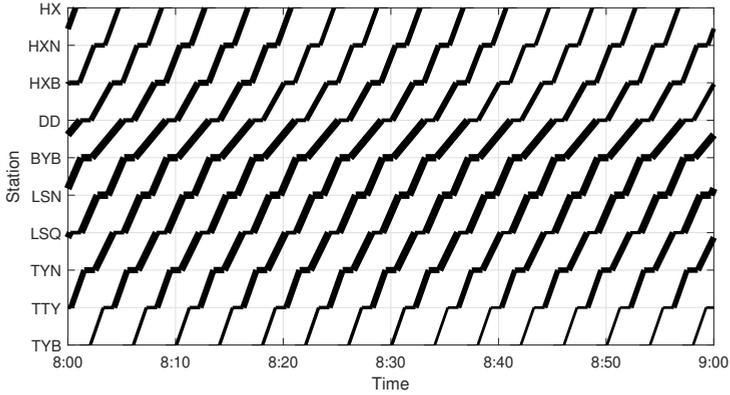


Figure 2.13: Basic timetable from station TYB to HX (the line thickness represents the number of passengers on board the train).

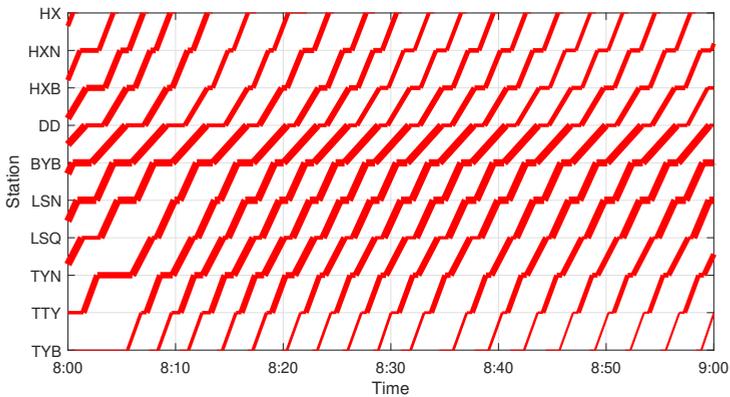


Figure 2.14: Timetable obtained by single-level MPC from station TYB to HX (the line thickness represents the number of passengers on board the train).

8:00AM to 9:00AM is shown in Fig. 2.13. The timetables generated by single-level MPC and bi-level MPC from 8:00AM to 9:00AM are respectively exhibited in Fig. 2.14 and Fig. 2.15. The time window 8:00AM to 9:00AM corresponds to time steps 3 and 4 in Fig. 13. The above simulation results indicate that the bi-level MPC approach based on the absorption model can generate practically implementable timetables online, which means the bi-level MPC approach can be implemented for real-time train scheduling of urban rail transit networks. Furthermore, the line thickness now indicates the number of passengers on board the current train. Then, it can be observed from Figures 2.13, 2.14, and 2.15 that compared with the basic timetable the optimized timetables allow more trains to transport more passengers so that passenger satisfaction can be improved.

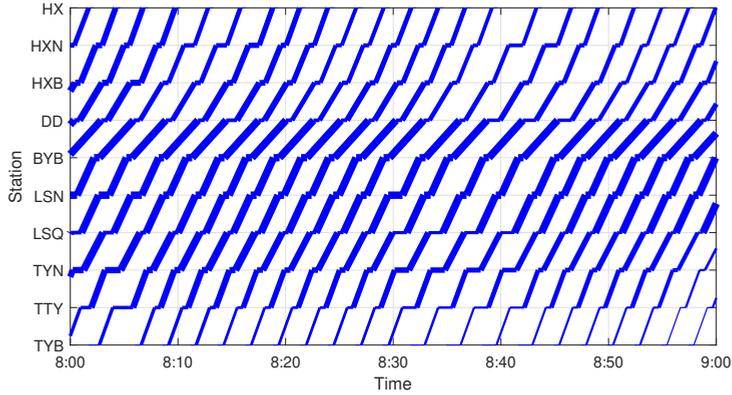


Figure 2.15: Timetable obtained by bi-level MPC from station TYB to HX (the line thickness represents the number of passengers on board the train).

2.7. CONCLUSIONS

In this chapter, we have investigated the real-time train scheduling problem considering time-dependent passenger OD demands and train speed profiles in urban rail transit networks. We have proposed a passenger absorption model to handle time-dependent passenger OD demands and rolling stock circulation in urban rail transit networks. The planning time window is divided into several phases, where the train departure frequency of each platform during each phase is considered. The passenger absorption model has been extended to a bi-level model where detailed timetables, detailed rolling stock circulation, train speed profiles, and train orders are also included. A bi-level MPC approach has been developed for real-time train scheduling of urban rail transit networks. The MPC optimization problems in both levels have been transformed into small-scale MILP problems, which enables us to solve them with existing MILP solvers. Numerical experiments show that the developed bi-level MPC approach yields a balanced trade-off between computation time and solution quality, which indicates that the developed model and the proposed bi-level MPC approach can be implemented for real-time train scheduling of urban rail transit networks.

The future work includes extending the bi-level framework to include more details of the urban rail transit system, e.g., flexible coupling of trains, regenerative braking, etc. Furthermore, uncertain passenger origin-destination demands and stochastic control approaches to deal with these uncertainties will also be a topic of future research. As the current chapter only considers time-varying passenger demands, the dynamic interactions between departure frequencies and passenger route choices still ask for further research. Moreover, some learning-based approaches, that integrate learning-based strategies to learn integer variables, can also be studied to solve the resulting optimization problem efficiently while ensuring constraint satisfaction.

3

TRAIN DEPARTURE FREQUENCY OPTIMIZATION WITH UNCERTAIN PASSENGER FLOWS

Real-time train scheduling is essential for passenger satisfaction in urban rail transit networks. This chapter focuses on real-time train scheduling for urban rail transit networks considering uncertain time-dependent passenger origin-destination demands. First, a macroscopic passenger flow model we proposed before is extended to include rolling stock availability. Then, a distributed-knowledgeable-reduced-horizon (DKRH) algorithm is developed to deal with the computational burden and the communication restrictions of the train scheduling problem in urban rail transit networks. For the DKRH algorithm, a cost-to-go function is designed to reduce the prediction horizon of the original model predictive control approach while taking into account the control performance. By applying a scenario reduction approach, a scenario-based distributed-knowledgeable-reduced-horizon (S-DKRH) algorithm is proposed to handle the uncertain passenger flows with an acceptable increase in computation time. Numerical experiments are conducted to evaluate the effectiveness of the developed DKRH and S-DKRH algorithms based on real-life data from the Beijing urban rail transit network. The simulation results indicate that DKRH can be used to achieve real-time train scheduling for the urban rail transit network, while S-DKRH can handle the uncertainty in the passenger flows with an acceptable sacrifice in computation time.

This chapter is based on [75] and [77].

3.1. INTRODUCTION

Urban rail transit plays an increasingly prominent role in public transportation of big cities due to its stability, high transport capacity, and energy efficiency. Real-time train scheduling is recognized as an effective way to improve passenger satisfaction and to reduce the operational costs under the infrastructure limitations of urban rail transit networks. With the rapid expansion of network scale and the growing passenger demands in urban rail transit systems, it becomes increasingly challenging to achieve real-time train scheduling while considering uncertain time-dependent passenger origin-destination (OD) demands and operational costs.

3.1.1. PASSENGER-ORIENTED TRAIN SCHEDULING FOR A SINGLE LINE

Several methods are reported in the literature to optimize arrival and departure times of trains at each platform in a single line. One important trend is to develop more practically implementable train scheduling strategy by including more attributes of train operation and infrastructure restrictions, e.g., train speed profiles [51; 135], rolling stock circulation [93; 133], train stopping plan [53]. Wang et al. [135] explored the train scheduling problem of a metro line while taking train capacity and speed profiles into account, and then an iterative convex programming approach is proposed to solve the resulting nonlinear nonconvex optimization problem. Shi et al. [111] investigated a flexible train capacity allocation strategy for a metro line where carriages are reserved for different stations based on time-dependent passenger demands, and the resulting nonlinear integer programming problem is solved through a variable neighborhood search algorithm. Zhou et al. [157] incorporated rolling stock circulation into the train scheduling problem considering passenger demands on a tidal oversaturated metro line, so that passenger demands in different phases can be satisfied.

The above studies are limited to passenger-oriented train scheduling problems of a single line. For an urban rail transit network, different lines typically interact with each other through transfer passengers. Therefore, train scheduling considering detailed passenger origin-destination (OD) demands in urban rail transit networks is regarded as an important direction to further improve passenger satisfaction [19].

3.1.2. PASSENGER-ORIENTED TRAIN SCHEDULING FOR NETWORKS

Train scheduling in urban rail transit networks with time-dependent passenger OD demands is challenging due to the requirement for network coordination and the scale of the resulting problem. In order to minimize the energy consumption of trains and the total travel time of passengers, Wang et al. [136] formulated time-dependent passenger OD demands in an urban rail transit network by an event-driven model, where arrival events, departure events, and passenger arrival rates change events are proposed to describe the movement of trains and passengers. Yin et al. [150] proposed a mixed-integer linear programming (MILP) formulation to handle the over-crowdedness of stations in an urban rail transit network, and a decomposition-based adaptive large neighborhood search approach was developed to improve the computational efficiency. Luan and Corman [84] included the train scheduling and passenger routing process in an integrated model, and the resulting mixed-integer nonlinear programming (MINLP) problem is reformulated as an MILP problem to minimize passenger disutility (i.e., passenger delay,

travel time, and the number of stranded passengers) and total train delay.

Considering the computational complexity of explicitly integrating departure and arrival times in an urban rail network with time-dependent passenger OD demands, optimizing train departure frequencies of each line has become a promising direction in passenger-oriented train scheduling [19; 30]. Canca et al. [19] optimized line frequencies and capacities by solving an MINLP problem. Liu et al. [74] developed a novel passenger flow model to determine train departure frequencies, i.e., the number of trains per unit time in each line, where time-dependent passenger OD demands and train capacities are included. The resulting optimization problem can be exactly transformed into an MILP problem, which can be solved efficiently by state-of-the-art MILP solvers [74]. Nevertheless, most existing studies in passenger-oriented train scheduling of urban rail networks do not include rolling stock availability due to the computational complexity issue, leaving an open gap for generating a practically implementable timetable.

3.1.3. MPC FOR REAL-TIME RAILWAY TRAFFIC MANAGEMENT

The train scheduling problem is a typical constrained control problem [76]. Model predictive control (MPC) is a methodology for addressing real-time constrained control problems [52; 92]. Based on a switching max-plus-linear model, a real-time train scheduling method was developed in [125] to minimize train delays and operational costs. Caimi et al. [15] dealt with train rescheduling problems for complex railway station areas by using MPC. However, as it is an optimization-based control approach, centralized MPC can be difficult to implement in real-life railway networks because of its computational complexity and global information requirements. These issues become more challenging in the case of large-scale networks.

For general large-scale systems, many researchers have developed non-centralized methods that coordinate subsystems in a decentralized, distributed, or hierarchical manner to achieve fast and effective solutions for the overall system [56; 62; 90]. Furthermore, non-centralized control methods have also been used in railway train scheduling problems. Kersbergen et al. [55] developed several distributed MPC methods for the railway traffic management problem where the arrival and departure times, breaking connections, and train orders in the railway network were jointly optimized. Luan et al. [85] applied three distributed optimization approaches, i.e., an alternating direction method of multipliers approach, a priority-rule-based approach, and a cooperative distributed robust safe but knowledgeable (CDRSBK) algorithm for real-time traffic management of railway networks. Numerical experiments show that the CDRSBK approach with train-based decomposition performs best on the basis of feasibility, optimality, and computational efficiency.

The train scheduling of urban rail transit networks with time-dependent passenger OD demands is challenging because of the large computational burden. The advanced non-centralized control methods [56; 62; 90] and their successful applications in railway [55; 85] have open opportunities to develop a new efficient distributed MPC method for passenger-oriented train scheduling problems.

3.1.4. TRAIN SCHEDULING UNDER UNCERTAINTIES

There are many uncertain attributes in railway networks, e.g., uncertain passenger flows and uncertain delays, that could influence the performance of train schedules. Cacchi-ani et al. [14] developed three different MILP formulations based on light robustness (where uncertainty is handled by inserting different protection levels) to reduce passenger inconvenience caused by uncertain passenger demands in a high-speed railway line. The scenario approach [16; 17] is a general data-driven decision-making methodology that can deal with uncertainties of a system. The scenario approach typically captured uncertainties by a collection of representative scenarios, and the decision is then made by considering these representative scenarios. By using different scenarios to capture the uncertain train operation time in the network, Yang et al. [148] developed a two-stage stochastic integer programming model to minimize the expected passenger travel time and transfer activities, where the potential transfer stations are found at the first stage while the least time paths are provided at the second stage. Gong et al. [43] formulated an MINLP problem to optimize the operational costs on an urban rail transit line where passenger distribution is represented via several different scenarios. However, most research only considered uncertain passenger demands for a single line. Passenger demands in urban rail transit networks exhibit highly dynamic and random characteristics because trains typically operate with high density, and passengers can choose different routes and different trains to reach their destinations. Therefore, efficient approaches that can explicitly include uncertain passenger demands in urban rail networks still require further research.

3.1.5. CHAPTER CONTRIBUTIONS AND STRUCTURE

The current chapter deals with the real-time train scheduling problem considering uncertain time-dependent passenger origin-destination demands in urban rail transit networks. By extending the passenger absorption model developed in [74], the train scheduling problem with rolling stock availability can be addressed by using model predictive control where the optimization problem at each time step is formulated as a mixed-integer linear programming problem. Considering the computational issues, we develop a distributed model predictive control approach where each line is regarded as one subsystem. Furthermore, as passenger flows generally exhibit some degree of uncertainty, a scenario-based approach is incorporated into the distributed model predictive control approach to deal with these uncertainties.

The main contributions of the chapter are as follows:

1. A novel distributed-knowledgeable-reduced-horizon (DKRH) algorithm is developed for the train scheduling problem, where a new cost-to-go function is proposed considering computational complexity, prediction horizon, and future performance.
2. We incorporate a scenario-based distributed control scheme into the DKRH algorithm, and a scenario-based distributed-knowledgeable-reduced-horizon algorithm is developed to handle uncertain passenger flows in large-scale urban rail transit networks.

3. The passenger absorption model of [74] is extended to include rolling stock availability by taking into account the total number of available trains so as to generate practically implementable control strategies.

The remaining part of the chapter is organized as follows: Section 3.2 introduces the mathematical model used in this chapter. In Section 3.3, a distributed knowledgeable-reduced-horizon algorithm is developed. In Section 3.4, we propose a scenario-based distributed knowledgeable-reduced-horizon algorithm. In Section 3.5, the effectiveness of the developed approaches is evaluated based on real-life data from a part of the Beijing urban rail transit network. The chapter is concluded with final remarks in Section 3.6.

3.2. MATHEMATICAL MODEL

This section starts with the description of the mathematical model proposed by the authors in [74], followed by an extension of the model to include rolling stock availability. Some general explanations for the research problem of this chapter are as follows:

1. This chapter aims to adjust train schedules for urban rail transit networks online based on real-time passenger demands. We assume the routes of passengers are given a priori. Disturbances and disruptions are not in the scope of this chapter.
2. The current chapter is based on the passenger absorption model developed in [74], which has been developed to determine train departure frequencies (i.e., the number of trains departing from each platform per unit time) for urban rail transit networks.
3. After obtaining the departure frequency of each platform, a dedicated lower-level controller [79] can determine the detailed departure and arrival times of trains, where the departure interval during each phase is determined according to the corresponding departure frequency.

We start with introducing the notations for the model formulation in Section 3.2.1. Then, the passenger absorption model is summarized in Section 3.2.2. Section 3.2.3 introduces the constraints for the model and further extends the model to include rolling stock availability.

3.2.1. NOTATIONS

Tables 3.1, 3.2, and 3.3 respectively list the indices and input parameters, decision variables, and output variables of this chapter.

3.2.2. PASSENGER ABSORPTION MODEL

In the passenger absorption model, the number of passengers $n_{p,e}(k)$ waiting at platform p with station e as their destination at the start of each phase is updated by:

$$n_{p,e}(k+1) = n_{p,e}(k) + \rho_{p,e}(k)T + n_{p,e}^{\text{trans,arrive}}(k) - n_{p,e}^{\text{absorb}}(k), \quad (3.1)$$

where $\rho_{p,e}(k)$ is the average passenger arrival rate at platform p with station e as their destination during phase k ; T is the length of a phase; $n_{p,e}^{\text{trans,arrive}}(k)$ is the number

Notations	Definition
s, e	Index of stations, $o, d \in \mathcal{S}$, \mathcal{S} is the set of stations
p	Index of platforms, $p \in \mathcal{P}$, \mathcal{P} is the set of platforms
k	Index of phases
$s^{pla}(p)$	Succeeding platform of platform p
$p^{pla}(p)$	Preceding platform of platform p
T	Length of a phase
h_p^{\min}	Minimum departure-arrival headway at platform p
τ_p^{\min}	Minimum dwell time of train at platform p
t_p	Average running time of trains from platform p to its succeeding platform
γ_p	Average time for a train from the first platform of a line to platform p
C_{train}	Maximum capacity of a train
$\alpha_{p,e}(k)$	Fraction of passengers absorbed by trains at platform p with destination d during phase k
$\chi_{p,q,e}$	Proportion of passengers transferring from platform p to q with station d as their destination
$t_{p,q}^{transfer}$	Average time for passengers walking from platform p to platform q
$\rho_{s,e}^{station}(k)$	Passenger origin-destination demands with o as origin station and d destination station during phase k
$\lambda_{s,p,e}(k)$	Proportion of passengers at origin station o choosing platform p for their travel to destination e

Table 3.1: Indices and input parameters.

Notations	Definition
$f_p(k)$	The number of trains departing from platform p during phase k

Table 3.2: Decision variables

Notations	Definition
$\rho_{p,e}(k)$	Passenger arrival rate at platform p with station e as destination during phase k
$n_{p,e}(k)$	Number of passengers waiting at platform p with station e as their destination at the beginning of phase k
$n_{p,e}^{absorb}(k)$	Number of passengers at platform p with station e as their destination absorbed by trains during phase k
$C_p(k)$	Total remaining capacity of trains visiting platform p during phase k
$n_p^{want}(k)$	Total number of passengers that want to board trains at platform p during phase k
$n_{p,e}^{on-board}(k)$	Number of passengers on board of trains, when trains arrive at platform p , with station e as their destination during phase k
$n_{p,e}^{alight}(k)$	Number of passengers alighting from trains at platform p with station e as their destination during phase k
$n_{p,q,e}^{transfer}(k)$	Number of passengers transferring from platform p to q with station e as their destination during phase k
$n_{p,e}^{trans,arrive}(k)$	Number of transfer passengers arriving at platform p with station e as their destination during phase k
$g_p(k)$	Total number of transfer passengers arriving at platform p during phase k
$n_{p,e}^{depart}(k)$	Number of passengers departing from platform p with station e as their destination during phase k
$m_p(k)$	Total number of passengers departing from platform p during phase k

Table 3.3: Output variables

of transfer passengers arriving at platform p with destination e during phase k , and $n_{p,e}^{absorb}(k)$ is the number of passengers at platform p with destination e absorbed by

trains during phase k . Then, $\rho_{p,e}(k)$, $n_{p,e}^{\text{trans,arrive}}(k)$, and $n_{p,e}^{\text{absorb}}(k)$ can be computed by

$$\rho_{p,e}(k) = \lambda_{o,p,e}(k) \rho_{o,e}^{\text{station}}(k), \forall p \in \mathcal{P}_o^{\text{sta}}, \quad (3.2)$$

$$n_{p,e}^{\text{trans,arrive}}(k) = \sum_{q \in \text{cop}(p) \setminus \{p\}} \left(\frac{T - t_{q,p}^{\text{transfer}}}{T} n_{q,p,e}^{\text{transfer}}(k) + \frac{t_{q,p}^{\text{transfer}}}{T} n_{q,p,e}^{\text{transfer}}(k-1) \right), \quad (3.3)$$

$$n_{p,e}^{\text{absorb}}(k) = \alpha_{p,e}(k) n_p^{\text{absorb}}(k), \quad (3.4)$$

where $\rho_{s,e}^{\text{station}}(k)$ is passenger origin-destination demands at phase k with s and e as the origin station and the destination station, respectively; $\mathcal{P}_s^{\text{sta}}$ defines a set of platforms at station s ; and $\lambda_{s,p,e}(k)$ is the proportion¹ of passengers at station s who choose platform p for their travel to destination e ; $\text{cop}(p)$ defines a set of platforms located at the same station as platform p ; $t_{q,p}^{\text{transfer}}$ denotes the average transfer time for passengers from platform q to platform p ; $n_{p,q,e}^{\text{transfer}}(k)$ is the number of passengers transferring from platform p to platform q with station e as their destination during phase k ; $\alpha_{p,e}(k)$ is the fraction of passengers absorbed by trains at platform p with destination e during phase k ; $n_p^{\text{absorb}}(k)$ denotes the total number of passengers absorbed by trains at platform p during phase k .

For the variable $n_p^{\text{absorb}}(k)$ in (3.4), we have

$$n_p^{\text{absorb}}(k) = \min \left(n_p^{\text{want}}(k), C_p(k) \right), \quad (3.5)$$

$$n_p^{\text{want}}(k) = n_p(k) + \rho_p(k) T + g_p(k), \quad (3.6)$$

$$C_p(k) = f_p(k) \cdot C_{\text{train}} - \sum_{d \in \mathcal{S}} \left(n_{p,e}^{\text{on-board}}(k) - n_{p,e}^{\text{alight}}(k) \right), \quad (3.7)$$

with

$$\begin{aligned} n_p(k) &= \sum_{e \in \mathcal{S}} n_{p,e}(k), \quad \rho_p(k) = \sum_{e \in \mathcal{S}} \rho_{p,e}(k), \\ g_p(k) &= \sum_{e \in \mathcal{S}} n_{p,e}^{\text{arrive,transfer}}(k), \end{aligned} \quad (3.8)$$

where $n_p^{\text{want}}(k)$ is the total number of passengers that want to board trains at platform p during phase k ; $C_p(k)$ is the total remaining capacity of trains that visit platform p during phase k ; $f_p(k)$ is the number of trains that visit platform p during phase k ; C_{train} is the maximum capacity of a train, \mathcal{S} denotes the set of stations in the urban rail transit network, $n_{p,e}^{\text{on-board}}(k)$ is the number of passengers on board of trains at platform p with destination e during phase k , and $n_{p,e}^{\text{alight}}(k)$ is the number of passengers alighting from trains at platform p with destination e during phase k .

The number of passengers $n_{p,e}^{\text{depart}}(k)$ departing from platform p with destination e during phase k is

$$n_{p,e}^{\text{depart}}(k) = n_{p,e}^{\text{on-board}}(k) - n_{p,e}^{\text{alight}}(k) + n_{p,e}^{\text{absorb}}(k), \quad (3.9)$$

¹As passenger route choices observed from metro data collection systems typically exhibit consistent patterns, we assume that the proportions of passengers choosing each route are given a priori. Thus, $\lambda_{s,p,e}(k)$ can be estimated from historical data or obtained according to the shortest paths.

and we have

$$n_{p,e}^{\text{on-board}}(k) = \frac{T - r_{\text{ppla}(p)}}{T} n_{\text{ppla}(p),e}^{\text{depart}}(k) + \frac{r_{\text{ppla}(p)}}{T} n_{\text{ppla}(p),e}^{\text{depart}}(k-1), \quad (3.10)$$

$$n_{p,e}^{\text{alight}}(k) = \begin{cases} \sum_{q \in \text{cop}(p) \setminus \{p\}} n_{p,q,e}^{\text{transfer}}(k), & \text{if } e \in \mathcal{S} \setminus \{\text{sta}(p)\}, \\ n_{p,e}^{\text{on-board}}(k), & \text{if } e = \text{sta}(p), \end{cases} \quad (3.11)$$

$$n_{p,q,e}^{\text{transfer}}(k) = \chi_{p,q,e} n_{p,e}^{\text{on-board}}(k), \quad \forall q \in \text{cop}(p) \setminus \{p\}, \quad (3.12)$$

$$m_p(k) = \sum_{e \in \mathcal{S}} n_{p,e}^{\text{depart}}(k), \quad (3.13)$$

where $r_{\text{ppla}(p)}$ refers to the average running time of trains from the preceding platform $\text{ppla}(p)$ to platform p , and $T \gg r_{\text{ppla}(p)}$; $\text{sta}(p)$ defines the station of platform p ; $\chi_{p,q,e}$ is the proportion for passengers transferring from platform p to $q \in \text{cop}(p)$ with station e as their destination; $\text{cop}(p)$ defines a set of platforms located at the same station as platform p ; $m_p(k)$ denotes the total number of passengers departing from platform p during phase k .

3.2.3. CONSTRAINTS FOR THE ABSORPTION MODEL

DEPARTURE FREQUENCY CONSTRAINTS

In this chapter, we only consider the case that each line has one depot to accommodate trains. In general, each train at a line will visit every platform of the line before it returns to depot or starts as a new train service. In this context, the number of trains running on a line can be determined by the number of trains departing from the depot. Therefore, the number of trains $f_p(k)$ departing from platform p can be calculated by

$$f_p(k) = \frac{T - \phi_p}{T} f_{\text{fst}(p)}(k - \beta_p) + \frac{\phi_p}{T} f_{\text{fst}(p)}(k - \beta_p - 1), \quad (3.14)$$

$$\beta_p = \lfloor \gamma_p / T \rfloor, \quad \phi_p = \gamma_p - \beta_p T, \quad (3.15)$$

where $\text{fst}(p)$ defines the first platform of the line corresponding to platform p , i.e., the platform connected with the depot of the line, γ_p denotes the average time for a train from platform $\text{fst}(p)$ to platform p .

To ensure the safe operation of urban rail transit systems, the number of trains departing from platform p during phase k should be constrained by

$$f_p(k) \left(h_p^{\min} + \tau_p^{\min} \right) \leq T, \quad (3.16)$$

where h_p^{\min} and τ_p^{\min} are the minimum headway and the minimum dwell time at platform p , respectively.

ROLLING STOCK AVAILABILITY CONSTRAINTS

In real-life operations, the number of trains used for each line is restricted by the total number of available trains, i.e., the total number of trains running on the line should be smaller than or equal to the total number of available trains. Therefore, the rolling

stock availability should be included in order to generate a practically implementable timetable. Considering p as the platform connected with a depot, the train departing from platform p typically visits every platform of the line and requires an average time interval c_p to return to the depot, and we define c_p as the circulation time. Then, for the passenger absorption model, the trains departing from a depot during the circulation time should satisfy

$$f_p(k) + \sum_{i=1}^{\sigma_p-1} f_p(k-i) + \frac{\omega_p}{T} f_p(k-\sigma_p) \leq N_p^{\text{ts}}, \quad \forall p \in \text{dep}(p), \quad (3.17)$$

$$\sigma_p = \lceil c_p / T \rceil, \quad \omega_p = c_p - \sigma_p T, \quad (3.18)$$

where $\text{dep}(p)$ is the set of platforms that use the same depot with platform p ; N_p^{ts} is the total number of available trains for the line corresponding to platform p .

3.3. DISTRIBUTED KNOWLEDGEABLE-REDUCED-HORIZON ALGORITHM FOR TRAIN SCHEDULING

Based on the model predictive control (MPC) framework, in this section, we first develop a knowledgeable-reduced-horizon (KRH) approach where a novel cost-to-go function is designed to shorten the prediction horizon. A distributed control framework is then proposed to further reduce the computational burden of solving the MPC optimization problem, thereby achieving real-time train scheduling in the urban rail transit network. In the distributed control framework, each local agent generates its control decisions based on its local information and information from its neighbor agents. Such a framework is in accordance with the real-life situation where global information is typically not available in large-scale urban rail transit networks.

3.3.1. PROBLEM FORMULATION IN MPC SET-UP

In an urban rail transit network, passenger satisfaction is strongly related to the total time spent in the network. Based on the absorption model, the total travel time of passengers in the urban rail transit network during phase k is represented by

$$J^{\text{pass}}(k) = \sum_{p \in \mathcal{P}} \left(n_p(k)T + m_p(k)r_p + g_p(k)t_p^{\text{transfer}} \right), \quad (3.19)$$

where $n_p(k)T$ represents the total waiting time at platform p during phase k , $m_p(k)r_p$ denotes the total running time until the next platform for passengers departing from platform p during phase k , and $g_p(k)t_p^{\text{transfer}}$ represents the total transfer time of passengers at platform p during phase k .

The operational cost of an urban rail transit system is highly related to the energy consumption of trains. Based on the absorption model, the total energy consumption for trains departing from the platform during phase k is computed by

$$J^{\text{roll}}(k) = \sum_{p \in \mathcal{P}} f_p(k)E_p, \quad (3.20)$$

where E_p represents the average energy consumption for a train to run from platform p to its succeeding platform.

Therefore, the MPC optimization problem $\mathbf{P}_{k_0}^{\text{MPC}}$ for real-time train scheduling of urban rail transit networks is formulated as

$$\begin{aligned} \min_{\mathbf{f}(k)} J(k_0) &:= \sum_{k=k_0}^{k_0+N_0-1} (J^{\text{pass}}(k) + \xi J^{\text{roll}}(k)), \\ \text{subject to} & \quad (3.1) - (3.14), (3.16) - (3.17), \end{aligned} \quad (3.21)$$

where N_0 is the prediction horizon, and ξ is a weight balancing the objectives.

As explained in [74], the nonlinear optimization problem $\mathbf{P}_{k_0}^{\text{MPC}}$ can be transformed into a mixed-integer linear programming (MILP) problem $\mathbf{P}_{k_0}^{\text{MILP}}$ with the following form, which is exactly equivalent to the original optimization problem:

$$\min_{\substack{\mathbf{x}(k), \mathbf{f}(k) \\ \boldsymbol{\delta}(k), \mathbf{z}(k)}}} J(k_0) := \sum_{k=k_0}^{k_0+N_0-1} (J^{\text{pass}}(k) + \xi J^{\text{roll}}(k)) \quad (3.22)$$

subject to

$$\mathbf{x}(k+1) = A_k \mathbf{x}(k) + B_{1,k} \mathbf{f}(k) + B_{2,k} \boldsymbol{\delta}(k) + B_{3,k} \mathbf{z}(k), \quad (3.23)$$

$$E_{2,k} \boldsymbol{\delta}(k) + E_{3,k} \mathbf{z}(k) \leq E_{1,k} \mathbf{f}(k) + E_{4,k} \mathbf{x}(k) + E_{5,k}, \quad (3.24)$$

$$\mathbf{f}(k) \leq D_0 + \sum_{i=1}^K D_i \mathbf{f}(k-i), \quad (3.25)$$

$$k = k_0, \dots, k_0 + N_0 - 1,$$

where $\mathbf{x}(k)$ and $\mathbf{f}(k)$ respectively concatenate the state variables (i.e., the variables related to the passengers) and decision variables (i.e., the number of trains) of all platforms in the network in phase k ; $\boldsymbol{\delta}(k)$ and $\mathbf{z}(k)$ respectively represent the vector of auxiliary binary variables and auxiliary continuous variables in phase k . The compact equation (3.23) represents the linear and mixed-integer linear formulations of the equations in (3.1)-(3.14). Constraint (3.24) collects all the linear and mixed-integer linear model constraints and operational constraints in a matrix form. Constraint (3.25) collects the constraints of decision variables, i.e., (3.14) and (3.17), in a matrix form, where $K = \max_{p \in \mathcal{P}} \sigma_p$.

For detailed information of transforming nonlinear terms of the model into mixed-integer linear inequalities, we refer the interested readers to [7; 139].

3.3.2. KNOWLEDGEABLE-REDUCED-HORIZON ALGORITHM FOR REAL-TIME TRAIN SCHEDULING

The computational complexity of solving MILP problem $\mathbf{P}_{k_0}^{\text{MILP}}$ increases rapidly with the prediction horizon N_0 due to the increasing number of variables. Solving $\mathbf{P}_{k_0}^{\text{MILP}}$ at every MPC step is not tractable for large prediction horizons because of the real-time feasibility restriction. Shortening the prediction horizon to reduce the computational burden; however, a short prediction horizon may negatively affect the performance of the controller as less future information can be included in the decision-making process.

Inspired by the robust-safe-but-knowledgeable (RSBK) algorithm proposed in [61; 62], we develop a knowledgeable-reduced-horizon (KRH) algorithm to shorten the prediction horizon of the original MPC controller by a customized cost-to-go function. The optimization problem $\mathbf{P}_{k_0}^{\text{KRH}}$ for the KRH algorithm is defined as

$$\min_{\substack{\mathbf{x}(k), \mathbf{f}(k) \\ \boldsymbol{\delta}(k), \mathbf{z}(k)}}} J(k_0) := \sum_{k=k_0}^{k_0+N-1} \left(J^{\text{pass}}(k) + \xi J^{\text{roll}}(k) \right) + L_N(k_0) \quad (3.26)$$

subject to

$$\mathbf{x}(k+1) = A_k \mathbf{x}(k) + B_{1,k} \mathbf{f}(k) + B_{2,k} \boldsymbol{\delta}(k) + B_{3,k} \mathbf{z}(k), \quad (3.27)$$

$$E_{2,k} \boldsymbol{\delta}(k) + E_{3,k} \mathbf{z}(k) \leq E_{1,k} \mathbf{f}(k) + E_{4,k} \mathbf{x}(k) + E_{5,k}, \quad (3.28)$$

$$\mathbf{f}(k) \leq D_0 + \sum_{i=1}^K D_i \mathbf{f}(k-i), \quad (3.29)$$

$$k = k_0, \dots, k_0 + N - 1,$$

where $L_N(k_0)$ denotes the cost-to-go function associated with the terminal states of passengers at the end of the shortened horizon.

As the target of the controller is to minimize the total travel time of the passengers, the cost-to-go function is designed to determine the cost associated with the passengers that remain at the platforms at the end of the reduced prediction window, i.e., a reasonable estimate of the remaining travel time for passengers waiting at the platforms at the end of the prediction time window.

The cost-to-go function for the remaining passengers at the platforms is defined as:

$$L_N(k_0) = \sum_{p \in \mathcal{P}} \sum_{d \in \mathcal{S}} \left(n_{p,d}(k_0 + N) \sum_{j \in \mathcal{R}_{p,d}} \eta_{p,d,j} t_{p,d,j}^{\text{total}} \right), \quad (3.30)$$

where $\mathcal{R}_{p,d}$ represents the set of possible routes for passengers from platform p to their destination d (see *Remark 3.1* below for an example of $\mathcal{R}_{p,d}$), $\eta_{p,d,j}$ is defined as the percentage of passengers at platform p that will travel to station d through route j , and we have

$$\eta_{p,d,j} = \prod_{(q,q') \in \mathcal{D}_j^{\text{pair}}} \chi_{q,q',d}, \forall j \in \mathcal{R}_{p,d}, \quad (3.31)$$

where $\mathcal{D}_j^{\text{pair}}$ represents the set of platform pairs at a transfer station in route j , and $\chi_{q,q',d}$ is the proportion for passengers transferring from platform q to q' . As the route of passengers can be represented by several pairs of platforms, (3.31) calculates the percentage of passengers that intend to travel from p to d through route j . Since $\chi_{q,q',d}$ is estimated based on historical data, $\eta_{p,d,j}$ can be calculated offline.

Then, $t_{p,d,j}^{\text{total}}$ represents the average travel time for passengers from platform p to their destination d through route j , and $t_{p,d,j}^{\text{total}}$ can be calculated offline based on the average dwell times, the average running times, and the average transfer times related to the

platforms in route j :

$$t_{p,d,j}^{\text{total}} = t_{p,j}^{\text{avg}} + \sum_{(q,q') \in \mathcal{P}_j^{\text{pair}}} (t_{q,q'}^{\text{transfer}} + t_{q',j}^{\text{avg}}), \forall j \in \mathcal{R}_{p,d}, \quad (3.32)$$

where $t_{p,j}^{\text{avg}}$ denotes the average time for passengers from platform p to reach either the next transfer station or the destination station in route j .

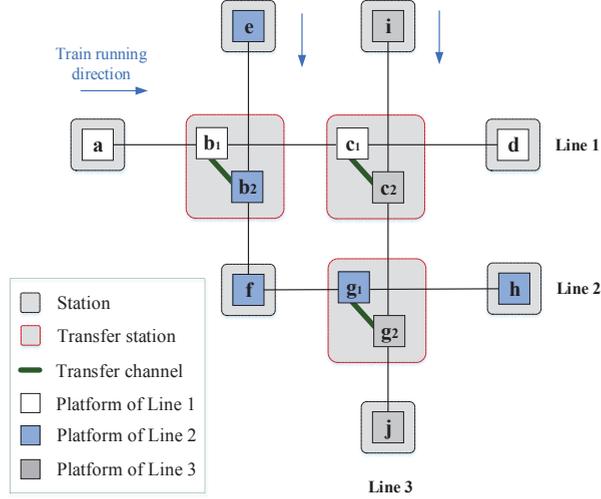


Figure 3.1: Example network

The construction of the sets $\mathcal{R}_{p,d}$ and $\mathcal{P}_j^{\text{pair}}$ is now illustrated in *Remark 3.1* through an example.

Remark 3.1. An example network is shown in Fig. 3.1. For passengers waiting at platform \mathbf{a} with destination \mathbf{h} at the end of the prediction window, there are two possible routes in the example network of Fig. 3.1. Thus, the set of possible routes for passengers from platform \mathbf{a} with destination \mathbf{h} is $\mathcal{R}_{\mathbf{a},\mathbf{h}} = \{\mathbf{a} - \mathbf{b}_1 - \mathbf{b}_2 - \mathbf{f} - \mathbf{g}_1 - \mathbf{h}, \mathbf{a} - \mathbf{b}_1 - \mathbf{c}_1 - \mathbf{c}_2 - \mathbf{g}_2 - \mathbf{g}_1 - \mathbf{h}\}$. The set of platform pairs for route $\mathbf{a} - \mathbf{b}_1 - \mathbf{b}_2 - \mathbf{f} - \mathbf{g}_1 - \mathbf{h}$ (named as route 1) is $\mathcal{P}_1^{\text{pair}} = \{(\mathbf{b}_1, \mathbf{b}_2)\}$, and the set of platform pairs for route $\mathbf{a} - \mathbf{b}_1 - \mathbf{c}_1 - \mathbf{c}_2 - \mathbf{g}_2 - \mathbf{g}_1 - \mathbf{h}$ (named as route 2) is $\mathcal{P}_2^{\text{pair}} = \{(\mathbf{c}_1, \mathbf{c}_2), (\mathbf{g}_2, \mathbf{g}_1)\}$. Then, the corresponding cost-to-go function can be calculated according to (3.30)-(3.32).

Comparing $\mathbf{P}_{k_0}^{\text{KRH}}$ and $\mathbf{P}_{k_0}^{\text{MPC}}$, we can find that the number of variables and constraints in (3.27) and (3.28) are reduced as the prediction horizon is reduced from N_0 to N . Similar to $\mathbf{P}_{k_0}^{\text{MILP}}$, the optimization problem $\mathbf{P}_{k_0}^{\text{KRH}}$ for the KRH algorithm is an MILP problem.

3.3.3. DISTRIBUTED KRH ALGORITHM FOR TRAIN SCHEDULING IN URBAN RAIL TRANSIT NETWORK

For large-scale urban rail transit networks, it may not be feasible to solve problem $\mathbf{P}_{k_0}^{\text{KRH}}$ in a centralized manner due to the computational burden and the communication re-

restrictions for collecting global information. In the urban rail transit network, different lines typically interact with their neighbor lines through transfer passengers as described in (3.3). In this section, a distributed-knowledgeable-reduced-horizon (DKRH) algorithm is developed for passenger-oriented real-time train scheduling of urban rail transit networks.

In urban rail transit networks, we can regard each line as a subsystem, where different subsystems interact with each other through transfer passengers. The corresponding objective functions associated with the travel time and energy consumption of subsystem l during phase k are

$$J_l^{\text{pass}}(k) = \sum_{p \in \mathcal{P}_l^{\text{line}}} \left(n_p(k)T + m_p(k)r_p + g_p(k)t_p^{\text{transfer}} \right), \quad (3.33)$$

$$J_l^{\text{roll}}(k) = \sum_{p \in \mathcal{P}_l^{\text{line}}} f_p(k)E_p, \quad (3.34)$$

where $\mathcal{P}_l^{\text{line}}$ is the set of platforms of line l . The cost-to-go functions corresponding to the terminal states of passengers of subsystem l is

$$L_{N,l}(k_0) = \sum_{p \in \mathcal{P}_l^{\text{line}}} \sum_{d \in \mathcal{S}} \left(n_{p,d}(k_0 + N) \sum_{j \in \mathcal{R}_{p,d}} \eta_{p,d,j} t_{p,d,j}^{\text{total}} \right). \quad (3.35)$$

The proposed DKRH algorithm is an iterative algorithm. In every control step of the proposed DKRH algorithm, different subsystems exchange information with their neighbor several times over several iterations. In each iteration, different subsystems solve their local problems in parallel, and then they exchange the new computed solution for the next iteration until the stopping criterion is met. At iteration step ϑ , the l -th subsystem calculates its control inputs through the following optimization problem, denoted as $\mathbf{P}_{l,k_0}^{\text{D}}$, by setting the variables of other subsystems as the corresponding values of the last iteration $\vartheta - 1$:

$$\min_{\substack{\delta_l(k), f_l(k) \\ \mathbf{x}_l(k), \mathbf{z}_l(k)}} J_l(k_0) := \sum_{k=k_0}^{k_0+N-1} \left(J_l^{\text{pass}}(k) + \xi J_l^{\text{roll}}(k) \right) + L_{N,l}(k_0) \quad (3.36)$$

subject to

$$\mathbf{x}_l(k+1) = A_{l,k} \mathbf{x}_l(k) + B_{1l,k} \mathbf{f}_l(k) + B_{2l,k} \boldsymbol{\delta}_l(k) + B_{3l,k} \mathbf{z}_l(k), \quad (3.37)$$

$$E_{2l,k} \boldsymbol{\delta}_l(k) + E_{3l,k} \mathbf{z}_l(k) \leq E_{1l,k} \mathbf{f}_l(k) + E_{4l,k} \mathbf{x}_l(k) + E_{5l,k}, \quad (3.38)$$

$$\mathbf{f}_l(k) \leq D_{l,0} + \sum_{i=1}^K D_{l,i} \mathbf{f}_l(k-i), \quad (3.39)$$

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

Algorithm 2 describes the DKRH algorithm, where l_{max} is the total number of lines in the network; ε is a small positive value which can be the machine precision. An initial estimate for the decision variable can be that of the basic timetable, i.e., the timetable with regular departure frequencies, which is typically used in the daily operation. As each line

is independent from the other lines, i.e., they do not share track and/or platforms with the other lines, trains in different lines will not conflict with each other. In this context, the regular departure frequencies are always feasible.

Algorithm 2 DKRH for real-time train scheduling

Input: k_{end} ; ϑ_{max} ; l_{max} ; ε ; initial estimate for the decision variable: $f_l^0(k)$, $l = 1, \dots, l_{\text{max}}$;

Output: optimal value $f_l(k)$, J_l

```

1:  $k \leftarrow k_0$ 
2: repeat
3:    $\vartheta \leftarrow 1$ 
4:   repeat
5:     for  $l = 1, \dots, l_{\text{max}}$  do
6:       solve problem  $\mathbf{P}_{l, k_0}^D$  and get  $f_l^\vartheta(k)$  and  $J_l^\vartheta$ 
7:       update (3.37), (3.38), and (3.39) for  $l$  by using  $f_l^\vartheta(k)$ 
8:     end for
9:      $\vartheta \leftarrow \vartheta + 1$ 
10:  until  $\vartheta = \vartheta_{\text{max}}$  or  $|J_l^\vartheta - J_l^{\vartheta-1}| \leq \varepsilon$ 
11:  apply control decision  $f_l(k)$  to each subsystem  $l$ 
12:   $k \leftarrow k + 1$ 
13: until  $k = k_{\text{end}}$ 

```

Remark 3.2. As we start with a feasible solution of the overall system and as the initial values of the decision variables are always feasible at every step, a feasible solution of problem $\mathbf{P}_{k_0}^D$ can always be found.

3.4. SCENARIO-BASED DKRH ALGORITHM

In this section, a scenario-based distributed-knowledgeable-reduced-horizon (S-DKRH) algorithm is developed to improve service quality in the presence of uncertain passenger flows.

For a large-scale urban rail transit network, the uncertainties generally consist of global uncertainties (e.g., the uncertainties caused by different weather conditions), and local uncertainties of each subsystems (i.e., the uncertainties due to different line conditions). Both global uncertainties and local uncertainties can be captured as several representative scenarios over the prediction window, which can be defined as global scenarios and local scenarios, respectively, based on historical data [45; 63]. If we include all combinations of global scenarios and local scenarios, the total number of combinations N_{com} is

$$N_{\text{com}} = N_{\text{glo}} \prod_{l=1}^{l_{\text{max}}} N_{\text{loc}, l}, \quad (3.40)$$

where N_{glo} denotes the number of global scenarios; $N_{\text{loc}, l}$ is the number of scenarios for subsystem l ; l_{max} is the total number of subsystems in the network; In this context, each subsystem should consider the complete set of scenarios, i.e., N_{com} scenarios, when generating its decision variables, which would rapidly increase the computational burden.

In order to address the computational complexity issue arising from the increasing number of scenarios for urban rail transit networks, we adopt a scenario reduction approach [73] into the DKRH algorithm. For subsystem l , the $N_{loc,l}$ scenarios will be directly used for subsystem l in the scenario-based approach. However, when considering the impact from subsystem l' ($l' \neq l$) on subsystem l , we use the scenario reduction approach to reduce the number of representative scenarios of subsystem l' from $N_{loc,l'}$ to $N_{l',l}$, i.e., $N_{l',l} \ll N_{loc,l'}$. In this context, subsystem l only needs to consider $N_{total,l} = N_{glo} N_{loc,l} \prod_{l' \neq l} N_{l',l}$ representative scenarios, which can be much smaller than that of original

scenario approach with $N_{glo} \prod_{l=1}^{l_{max}} N_{loc,l}$ representative scenarios. For example, Fig. 3.2 has three subsystems, and subsystem 2 only considers $N_{glo} \cdot N_{loc,2} \cdot N_{1,2} \cdot N_{3,2}$ representative scenarios instead of $N_{glo} \cdot N_{loc,1} \cdot N_{loc,2} \cdot N_{loc,3}$ representative scenarios, where $N_{1,2} \ll N_{loc,1}$, $N_{3,2} \ll N_{loc,3}$. Therefore, the computational burden of each subsystem is reduced significantly.

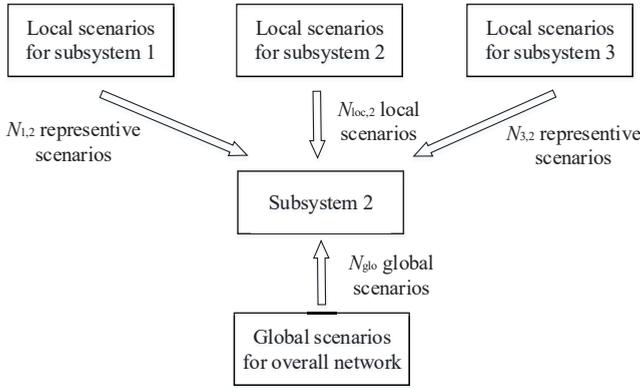


Figure 3.2: Reduced scenarios for agent 2 in an example with 3 agents.

Based on the above scenario reduction approach, we develop the S-DKRH algorithm. For subsystem l with scenario s , the corresponding objective functions are

$$J_{l,s}^{\text{pass}}(k) = \sum_{p \in \mathcal{P}_l^{\text{line}}} \left(n_{p,s}(k)T + m_{p,s}(k)r_p + g_{p,s}(k)t_p^{\text{transfer}} \right), \quad (3.41)$$

$$J_{l,s}^{\text{roll}}(k) = \sum_{p \in \mathcal{P}_l^{\text{line}}} f_p(k)E_{p,s}, \quad (3.42)$$

where $n_{p,s}(k)$, $m_{p,s}(k)$, $g_{p,s}(k)$, and $E_{p,s}$ respectively represent the values of $n_p(k)$, $m_p(k)$, $g_p(k)$, and E_p under scenario s . The corresponding cost-to-go functions is

$$L_{N,l,s}(k_0) = \sum_{p \in \mathcal{P}_l^{\text{line}}} \sum_{d \in \mathcal{S}} (n_{p,d,s}(k_0 + N) \sum_{r \in \mathcal{R}_{p,d}} \eta_{p,d,r} t_{p,d,r}^{\text{total}}), \quad (3.43)$$

with $n_{p,d,s}(k_0 + N)$ denoting the variable $n_{p,d}(k_0 + N)$ under scenario s .

In the S-DKRH algorithm, subsystem l considers only one representative scenario for each neighbor subsystem, and the variables of the neighbor subsystems are set as the corresponding values of the last iteration. At phase k_0 , the l -th subsystem generates its control decisions by solving the following chance-constraint optimization problem \mathbf{P}_{l,k_0}^S :

$$\min_{\mathbf{x}_l(k), \mathbf{f}_l(k), \boldsymbol{\delta}_l(k), \mathbf{z}_l(k)} J_l(k_0) := \sum_{s=1}^{N_{\text{total},l}} \mathbb{P}\{s\} \left(\sum_{k=k_0}^{k_0+N-1} \left(J_{l,s}^{\text{pass}}(k) + \xi J_{l,s}^{\text{roll}}(k) \right) + L_{N,l,s}(k_0) \right) \quad (3.44)$$

subject to

$$\mathbf{x}_{l,s}(k+1) = A_{l,s,k} \mathbf{x}_{l,s}(k) + B_{1l,s,k} \mathbf{f}_l(k) + B_{2l,s,k} \boldsymbol{\delta}_{l,s}(k) + B_{3l,s,k}^s \mathbf{z}_{l,s}(k), \quad (3.45)$$

$$E_{2l,s,k}^{\text{hard}} \boldsymbol{\delta}_{l,s}(k) + E_{3l,s,k}^{\text{hard}} \mathbf{z}_{l,s}(k) \leq E_{1l,s,k}^{\text{hard}} \mathbf{f}_l(k) + E_{4l,s,k}^{\text{hard}} \mathbf{x}_{l,s}(k) + E_{5l,s,k}^{\text{hard}}, \quad (3.46)$$

$$\sum_{s=1}^{N_{\text{total}}} \mathbb{P}\{s\} \mathbf{1} \left(E_{2l,s,k}^{\text{soft}} \boldsymbol{\delta}_{l,s}(k) + E_{3l,s,k}^{\text{soft}} \mathbf{z}_{l,s}(k) \leq E_{1l,s,k}^{\text{soft}} \mathbf{f}_l(k) + E_{4l,s,k}^{\text{soft}} \mathbf{x}_{l,s}(k) + E_{5l,s,k}^{\text{soft}} \right) \geq \boldsymbol{\theta}_l, \quad (3.47)$$

$$\mathbf{f}_l(k) \leq D_{l,0}^s + \sum_{i=1}^K D_{l,i}^s \mathbf{f}_l(k-i), \quad (3.48)$$

$$k = k_0, \dots, k_0 + N - 1,$$

where $\mathbb{P}\{s\}$ denotes the probability of s , and $N_{\text{total},l}$ is the total number of scenarios for agent l after scenario reduction. Eq. (3.45) represents the linear and mixed-integer linear formulations of the model explained in (3.1)-(3.14) for subsystem l under scenario s ; (3.46) collects the corresponding hard constraints; (3.47) denotes the chance constraints, i.e., the constraints related to operational performance, $\mathbf{1}(\cdot)$ defines the indicator function², and $\boldsymbol{\theta}_l \in (0, 1)$ indicates the minimally required probability that there is no constraint violation; (3.48) collects the hard constraints of decision variables, i.e., (3.14) and (3.17), for subsystem l with scenario s . By solving \mathbf{P}_{l,k_0}^S , we minimize the expected value of objective function (3.44) while including the corresponding constraint satisfaction in (3.46). Problem \mathbf{P}_{l,k_0}^S for the S-DKRH algorithm is also an MILP problem and can be solved efficiently by using existing MILP solvers.

Algorithm 3 provides the process of the S-DKRH algorithm, where \mathcal{L}_l represents the neighboring subsystems of l , i.e., lines connected with line l via transfer stations.

3.5. CASE STUDY

To evaluate the performance of the developed approaches, numerical experiments are conducted based on real-life data of the Beijing urban rail transit network. First, we present the urban rail transit network and some basic settings we use in the case study. Then, simulations are conducted to illustrate the effectiveness of the developed KRH and DKRH algorithms. Finally, we include uncertainty in the passenger flows in the simulation to show the performance of the S-DKRH algorithm.

² $\mathbf{1}(\cdot) = 1$ if the corresponding constraint is satisfied, otherwise $\mathbf{1}(\cdot) = 0$.

Algorithm 3 S-DKRH algorithm for real-time train scheduling

Input: k_{end} ; ϑ_{max} ; N_{glo} ; $N_{\text{loc},l}$; $N_{l',l}$; ε ; initial estimate for the decision variable: $f_l^0(k)$, $l = 1, \dots, l_{\text{max}}$

Output: optimal value $f_l(k)$, J_l

- 1: **for** $l = 1, \dots, l_{\text{max}}$ **do**
- 2: construct $N_{\text{local},l}$ scenarios for local controller
- 3: construct $N_{l',l}$ combined scenario for its neighbors
- 4: **end for**
- 5: $k \leftarrow k_0$
- 6: **repeat**
- 7: $\vartheta \leftarrow 1$
- 8: **repeat**
- 9: **for** $l = 1, \dots, l_{\text{max}}$ **do**
- 10: solve problem (3.44) and get $f_l^\vartheta(k)$ and J_l^ϑ
- 11: update constraints in problem \mathbf{P}_{l,k_0}^S for $l \in \mathcal{L}_l$
- 12: **end for**
- 13: $\vartheta \leftarrow \vartheta + 1$
- 14: **until** $\vartheta = \vartheta_{\text{max}}$ or $|J_l^\vartheta - J_l^{\vartheta-1}| \leq \varepsilon$
- 15: apply control decision $f_l(k)$ to each subsystem l
- 16: $k \leftarrow k + 1$
- 17: **until** $k = k_{\text{end}}$

3.5.1. SETUP

The network we consider includes four bidirectional lines of the Beijing urban rail transit network, i.e., Changping Line, Line 8, Line 13, and Line 15 (see Fig. 3.3). Therefore, we have four subsystems for the distributed control approaches. The main parameters for the case study are shown in Table 3.4, where the circulation time c_p mentioned in (3.17) and (3.18) is estimated based on the average running time and the regular dwell time. The basic timetable is generated by implementing the regular headway and the regular dwell time in Table 3.4.

Table 3.4: Parameters of the network for the case study

Parameters	Changping Line	Line 8	Line 13	Line 15
Minimum headway	120 s	120 s	120 s	120 s
Regular headway	180 s	180 s	180 s	180 s
Maximum dwell time	360 s	360 s	360 s	360 s
Minimum dwell time	30 s	30 s	30 s	30 s
Regular dwell time	60 s	60 s	60 s	60 s
Train capacity	2400	2400	2400	2400
Average transfer time	60 s	60 s	60 s	60 s
Phase time T	1800 s	1800 s	1800 s	1800 s
Total available trains	24 trains	28 trains	32 trains	24 trains
Circulation time	5407.5 s	6227.6 s	7101.9 s	3315.9 s

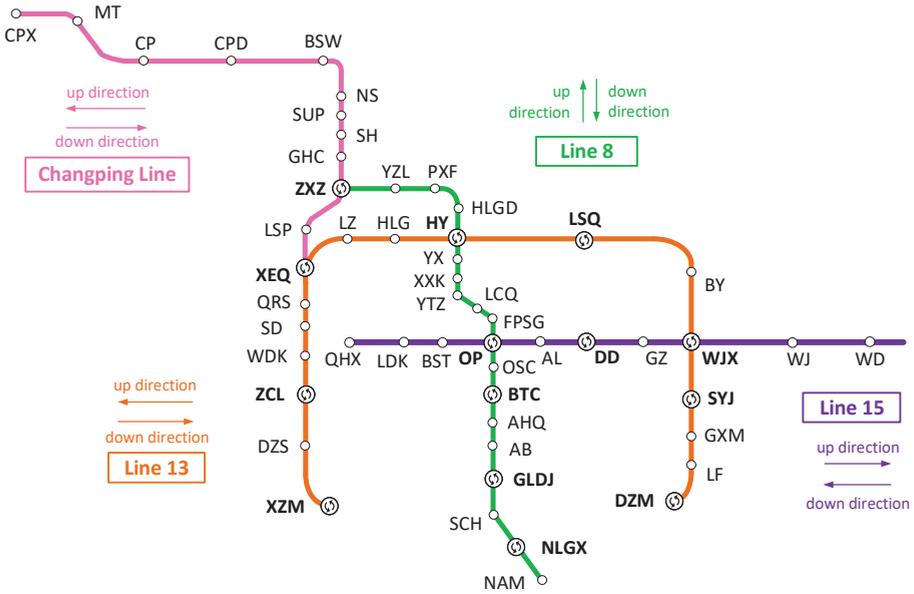


Figure 3.3: Layout of the considered urban rail transit network (with 4 lines).

The passenger OD demands are obtained based on the real-life passenger data of the Beijing urban rail transit network. In particular, we use the real-life data on passengers entering and exiting flows of each station in the network of Fig. 3.3. The data is updated every 30 minutes. In the case study, we consider passenger flows from 7:00-12:00, which includes situations of both peak hours and off-peak hours. We directly use the passenger OD demands for the simulation of the deterministic case. For the uncertain case, we generate uncertain passenger OD demands using Poisson distribution [151] based on the passenger flow data with additional variations within 30% to cover both normal and over-crowdedness cases for simulations and the scenario-based approach. The number of representative scenarios for each local subsystem is 5, while both the number of global scenarios and scenarios for neighbor subsystems are 1. In this context, each subsystem only needs to consider 5 scenarios in total.

After generating the number of trains departing from each platform during each phase, we generate the detailed departure and arrival times of each train by the lower-level controller developed in [79], where five different speed profile options are considered, which are calculated according to the method in [135], with a maximum acceleration of 0.8 m/s^2 , a maximum deceleration of 0.75 m/s^2 , and cruising speeds of 67 km/h, 70 km/h, 73 km/h, 76 km/h, 80 km/h, respectively. The length between every two consecutive stations is openly accessible on the website of Beijing Subway³. We use the passenger absorption model as the prediction model and an elaborate model from the literature (i.e., the model in [9; 136]) as the simulation model to evaluate the effective-

³<https://www.bjsubway.com/station/zjgls/>

ness of the developed approaches. In each MPC step, the resulting mixed-integer linear programming problem is solved by the gurobi solver called from MATLAB (R2019b). The simulations are performed on a computer with an Intel Xeon W-2223 CPU and 8GB RAM.

3.5.2. REAL-TIME TRAIN SCHEDULING FOR THE DETERMINISTIC CASE

We conduct simulations for the deterministic case to show the effectiveness of the developed knowledgeable-reduced-horizon (KRH) algorithm and the distributed knowledgeable-reduced-horizon (DKRH) algorithm. For comparison, we also perform simulations for the basic timetable as well as the original MPC approach.

According to the circulation time of each line, the prediction horizon of all MPC approaches should be $N \geq 4$ (i.e., the length of the prediction time window satisfies $t \geq 7200$ s) to ensure that the MPC optimization problem can cover every station in the network. The prediction horizon of the original MPC approach is set as $N = 6$, while the prediction horizon for KRH and DKRH is reduced to $N = 4$. For the DKRH approach, we use three subsystems, where each line in Fig. 3.3 is regarded as one subsystem. Considering the real-time implementation, we set the maximum solution time for each MPC step to 3600 s to meet the real-time feasibility requirement.

Table 3.5: Simulation results for different approaches under the deterministic case

Approach	Objective	Improvement	CPU time (s)	
			t_{avg}	t_{max}
Basic timetable	$8.4607 \cdot 10^4$	-	-	-
MPC (N=6)	$7.0633 \cdot 10^4$	16.52%	3600.0	3600.0
KRH (N=4)	$7.1614 \cdot 10^4$	15.36%	250.7	636.0
DKRH (N=4)	$7.1174 \cdot 10^4$	15.88%	35.5	37.8

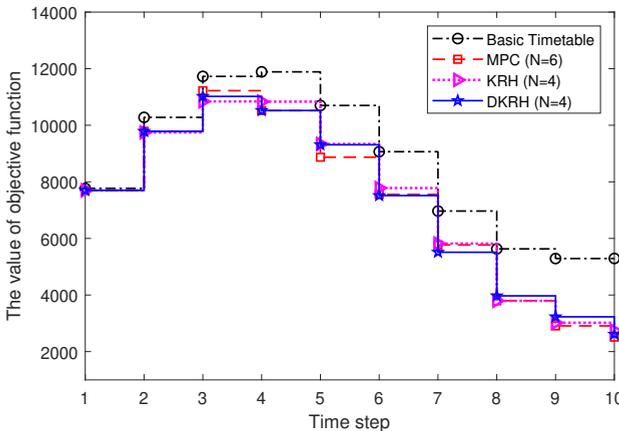


Figure 3.4: Value of the objective function at each time step.

The simulation results are displayed in Table 3.5, where the value of objective function and computation time of each approach are collected. The value of objective function in each MPC step is shown in Fig. 3.4. These results show that MPC, KRH, and DKRH can improve the performance of the basic timetable, with an improvement of 16.52%, 15.36%, and 15.88%, respectively. As a real-time control approach, the online computational burden is an essential issue for MPC, which is significantly influenced by the prediction horizon. The original MPC approach with prediction horizon $N = 6$ cannot calculate its optimal solution within 3600 s. By using the cost-to-go function in the developed KRH algorithm, the prediction horizon and the solution space are reduced. Thus, the CPU time of the KRH algorithm is reduced significantly while ensuring an acceptable level of solution quality.

As we divide the original problem into three smaller subproblems in the DKRH algorithm, the computational burden of each subproblem is further reduced. Compared with the KRH algorithm, the average CPU time for the DKRH algorithm is reduced from 250.7 s to 35.5 s, and the maximum CPU time is reduced from 636.0 s to 37.8 s. The solution time of the DKRH algorithm is further reduced while maintaining the same level of control performance.

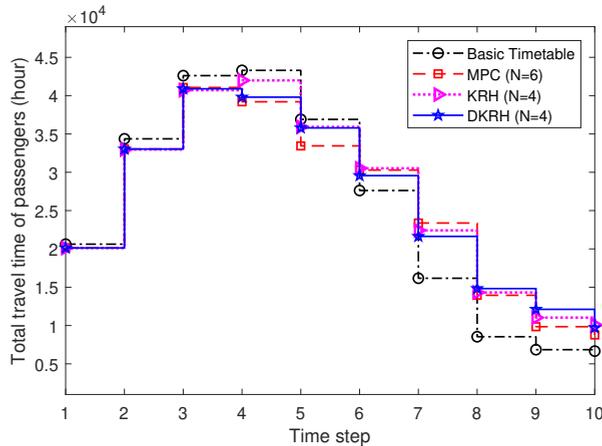


Figure 3.5: Total travel time of passengers at each time step.

The total travel time of passengers is shown in Fig. 3.5. To further illustrate the results, the number of trains departing from the depot of Line 13 is given in Fig. 3.6. In Fig. 3.5 and Fig. 3.6, time steps 1-3 represent the morning peak hours at 7:00-8:30. Compared with the basic timetable, more trains are scheduled to attend the large passenger demand in the morning peak hours. Since the maximum number of available trains for Line 13 is 32, and the circulation time for Line 13 is 7101.9 s, which is approximately equal to the length of 4 phases, the maximum number of trains scheduled for each phase is restricted. Compared with peak hours, fewer trains are scheduled in off-peak hours to reduce operational costs with an acceptable increase in the total passenger travel time. Based on the developed approaches, we can obtain the number of trains departing from

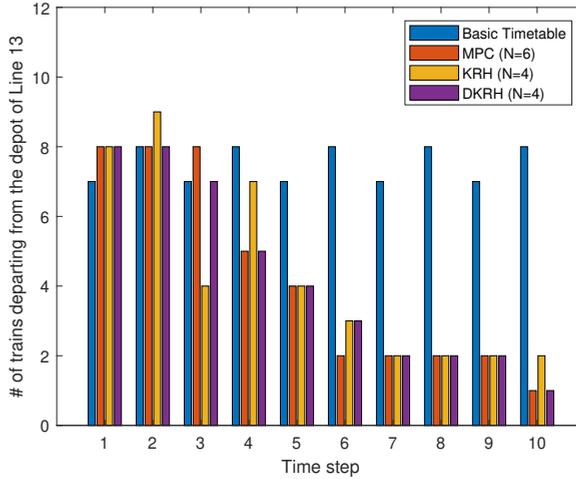


Figure 3.6: Number of trains departing from the depot of Line 13.

each platform during each phase, and the corresponding timetable can be further generated.

The simulation results indicate that both KRH and DKRH can be used for real-time train scheduling for urban rail transit networks. In particular, when there are no communication restrictions between different lines, KRH can be used to get a high-quality solution; otherwise, especially for large-scale networks when centralized control for the whole network is not possible due to the communication restrictions, DKRH can be used to achieve real-time train scheduling for the urban rail transit network.

3.5.3. REAL-TIME TRAIN SCHEDULING WITH UNCERTAIN PASSENGER FLOWS

In general, passenger demands in urban rail transit networks satisfy a Poisson distribution [151]. In this section, we perform simulations when there exists uncertainty in passenger flows to evaluate the effectiveness of the developed scenario-based distributed knowledgeable-reduced-horizon (S-DKRH) algorithm.

We first start simulations for one uncertain scenario. To have a baseline, we also conduct a simulation with perfect knowledge of the uncertainties, which is indicated as P-DKRH below. It is worth noting that P-DKRH is not realistic as it is not possible to have perfect knowledge of the uncertainties in real life. For the DKRH algorithm, we use the expected value of the passenger demands to calculate the timetable. Using the simulation results in Section 3.5.2, the prediction horizon for P-DKRH, DKRH, and S-DKRH is set as $N = 4$.

Table 3.6 and Fig. 3.7 show the simulation results of different approaches under uncertain passenger flows. Compared with the basic timetable, an improved performance can be observed for both DKRH and S-DKRH, with an improvement of 10.38% and 12.90%, respectively. Compared with DKRH, the objective function value of S-DKRH is closer to

Table 3.6: Simulation results for different approaches under the uncertain case

Approach	Objective	Improvement	CPU time (s)	
			t_{avg}	t_{max}
Basic timetable	$9.7262 \cdot 10^4$	-	-	-
P-DKRH	$8.4282 \cdot 10^4$	13.35%	35.4	40.9
DKRH	$8.7171 \cdot 10^4$	10.38%	34.4	37.7
S-DKRH	$8.4718 \cdot 10^4$	12.90%	347.6	385.9

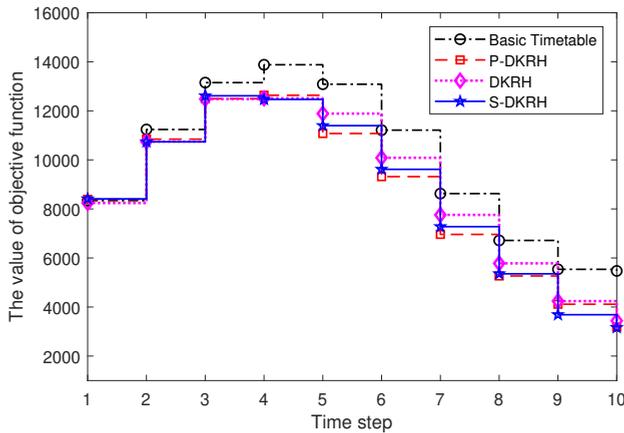


Figure 3.7: Value of the objective function at each time step.

that of P-DKRH, which implies the effectiveness of the scenario-based approach. Both DKRH and S-DKRH satisfy the real-time feasibility requirement for the given case study. The computational burden of S-DKRH is larger than that of DKRH, and the average CPU time increases from 34.4 s to 347.6 s for S-DKRH. The simulation results demonstrate that a suitable choice is required in real-life applications, i.e., when the CPU power is sufficient, S-DKRH is a better choice to obtain a higher-quality solution; otherwise, when the CPU power is not sufficient, DKRH can be used to calculate a timetable within a shorter period of time with acceptable performance.

The number of trains departing from the depot of Line 13 in the uncertain case is shown in Fig. 3.8. Time steps 1-3 are associated with the morning peak hours at 7:00-8:30, and it can be observed that more trains are scheduled at time steps 1-3 to attend the large passenger demands.

To further demonstrate the effectiveness of the developed S-DKRH algorithm, simulations are carried out in 10 different scenarios. The 10 scenarios are generated based on Poisson distribution with the real-life passenger entering and exiting flow data as the expected value. The average value and the standard deviation of the objective function values for the basic timetable, and the timetable obtained by P-DKRH, DKRH, and S-DKRH are calculated. Compared with the average objective function value of the basic timetable, P-DKRH, DKRH, and S-DKRH yield an improvement of 14.01%, 11.92%, and

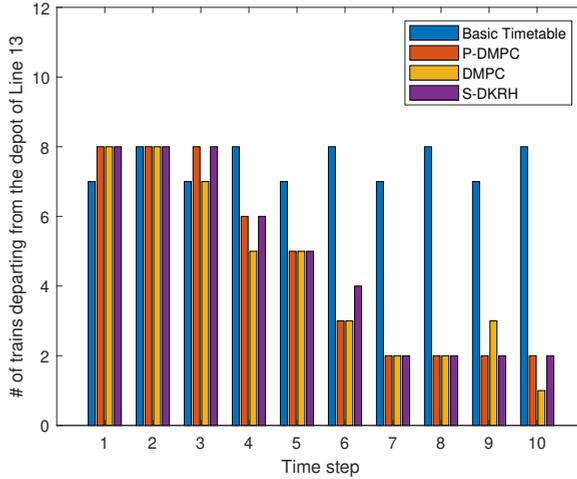


Figure 3.8: Number of trains departing from the depot of Line 13 in the uncertain case.

Table 3.7: Comparison of the objective function values for different approaches

	Average	Standard deviation
Basic timetable	$9.3997 \cdot 10^4$	$6.6490 \cdot 10^3$
P-DKRH	$8.0825 \cdot 10^4$	$6.5604 \cdot 10^3$
DKRH	$8.2794 \cdot 10^4$	$7.7432 \cdot 10^3$
S-DKRH	$8.1147 \cdot 10^4$	$7.2378 \cdot 10^3$

13.67%, respectively. Although P-DKRH outperforms DKRH and S-DKRH with respect to both the average value and the standard deviation, as stated before, P-DKRH is not realizable in real life⁴. It can be observed in Table 3.7 that the average objective function value and the standard deviation of S-DKRH are smaller than that of DKRH. The simulation results imply that S-DKRH can be a suitable choice to handle uncertain passenger flows.

3.6. CONCLUSIONS

In this chapter, we have investigated real-time train scheduling for urban rail transit networks considering uncertain time-dependent passenger OD demands. The passenger absorption model of [74] has been extended to include the rolling stock availability to generate more practically implementable timetables by considering the total number of available trains. To reduce the prediction horizon of the real-time train scheduling problem, a novel cost-to-go function has been developed. By considering different

⁴As we use the absorption model as the prediction model and the model in [136] as the simulation model, there exists a model mismatch issue, which may yield the objective function value of P-DKRH larger than that of DKRH and S-DKRH in some scenarios.

lines as different subsystems, a distributed-knowledgeable-reduced-horizon (DKRH) algorithm has been proposed considering the computational complexity and communication restrictions in practical urban rail transit networks. Furthermore, a scenario-based distributed-knowledgeable-reduced-horizon algorithm (S-DKRH) has been developed to deal with uncertain passenger flows. Numerical experiments have been conducted to illustrate that 1) DKRH can be used for real-time train scheduling of urban rail transit networks and 2) the S-DKRH algorithm yields better performance than DKRH with an acceptable increase in computation time for uncertain cases.

The results in this chapter can help the operator to optimize train schedules to handle uncertain time-dependent passenger demands. Future research includes developing efficient solution approaches for the resulting optimization problems to further improve the real-time feasibility of the approach. In particular, integrating learning-based strategies to learn integer variables can be a possible choice to speed up the optimization process. Furthermore, next to optimizing the train departure frequencies, adjusting train composition can also be a choice to handle time-dependent passenger demands.

4

MPC FOR PASSENGER-ORIENTED REAL-TIME TRAIN RESCHEDULING

Real-time timetable scheduling is an effective way to improve passenger satisfaction and to reduce operational costs in urban rail transit networks. In this chapter, a novel passenger-oriented network model is developed for real-time timetable scheduling that can model time-dependent passenger origin-destination demands with consideration of a balanced trade-off between model accuracy and computation speed. Then, a model predictive control (MPC) approach is proposed for the timetable scheduling problem based on the developed model. The resulting MPC optimization problem is a nonlinear non-convex problem. In this context, the online computational complexity becomes the main issue for the real-time feasibility of MPC. To reduce the online computational complexity, the MPC optimization problem is therefore reformulated into a mixed-integer linear programming (MILP) problem. The resulting MILP problem is exactly equivalent to the original MPC optimization problem and can be solved very efficiently by existing MILP solvers, so that we can obtain the solution very fast and realize real-time timetable scheduling. Numerical experiments based on a part of Beijing subway network show the effectiveness and efficiency of the developed model and the MILP-based MPC method.

This chapter is based on [76].

4.1. INTRODUCTION

Urban rail transit is recognized as a safe, sustainable, and high-efficiency transportation modality, and it plays an increasingly important role in the public transportation systems. Real-time timetable scheduling is one of the most effective and efficient approaches to improve passenger satisfaction and to reduce operational costs. With the rapidly growing passenger demands and the increasing urban rail network scale, advanced urban rail network models and the corresponding control approaches are crucial to obtain efficient timetables and to improve the performance of transportation services.

In the research on railway traffic management problems, one important class of studies pays attention to departure times and arrival times of trains in the network [28; 125; 142], where the aim is to improve the performance of daily timetables and to minimize the effects of delays or cascade delays caused by disturbances. Another class of studies incorporates rolling stock circulation [133], train orders [22], conflict resolution [128], etc., into timetable scheduling problems, which is particularly helpful when disruptions occur, as it can be used to adjust the impacted timetable and make the railway network recover from disruptions as soon as possible. In this chapter, we consider passenger demands when generating timetables online in order to provide high-quality service for passengers.

There are many studies related to passenger-oriented timetable scheduling. Several studies handle passenger flows while including rolling stock circulation [94; 133], speed profiles [51], and short-turning [158], but without detailed passenger origin-destination (OD) information. Another direction of studies addresses passenger OD demands on a single line [135; 152]. However, the passenger demands in networks are more complex than those of a single line due to the transfer activities of passengers, and hence, efficient approaches that consider passenger OD demands in urban rail networks are required. Some studies consider passenger OD demands in railway networks [159] or urban rail networks [136; 150]; however, the computational complexity of including the time-dependent passenger demands and the detailed number of passengers is still a challenging issue. In real life, passenger demands are typically represented as time-dependent OD matrices. Nevertheless, most studies on timetable scheduling problems do not take the detailed time-dependent passenger OD demands into account, leaving an open gap for further improving the timetable through closed-loop control while taking real-time passenger demands into account.

Generally, the timetable scheduling problem is a typical constrained control problem. Model predictive control (MPC) is a well-recognized effective method for its ability to handle multi-variable constrained control problems [38; 91; 104]. The online computational burden of the MPC optimization problem is the main challenge for real-time timetable scheduling when taking time-dependent passenger OD demands into account. Passenger flows in railway networks have a certain similarity with traffic flows in urban road networks. The efficient traffic flow model and fast MPC methods for the urban road network [71; 72] have inspired us to develop an efficient model for passenger-oriented railway traffic networks and to develop efficient MPC methods for the real-time timetable scheduling problem.

The main contributions of the chapter are listed as follows:

1. A novel model for passenger-oriented urban rail traffic networks is proposed that

- can explicitly include the number of passengers in urban rail networks under time-dependent passenger origin-destination demands.
2. Thanks to the notion of cycle time introduced in this chapter, the time-varying passenger demands are approximated as piecewise constant functions in the model to achieve a trade-off between model accuracy and computation speed.
 3. An MPC approach is proposed for the real-time timetable scheduling problem based on the developed model. The nonlinear MPC optimization problem is exactly transformed into an MILP problem to reduce the online computational burden.

The rest of this chapter is structured as follows. Section 4.2 summarizes the literature related to this chapter. In Section 4.3, the passenger-oriented urban rail traffic model is proposed. In Section 4.4, the MPC controller is designed for the passenger-oriented timetable scheduling problem based on the proposed model. In Section 4.5, the MPC optimization problem is solved with different methods, and an MILP-based approach is proposed. Section 4.6 provides case studies to illustrate the accuracy of the model and the efficiency of the developed method. Finally, conclusions are given in Section 4.7.

4.2. STATE OF THE ART

4.2.1. MODELS FOR TIMETABLE SCHEDULING

In the literature, many models and methods have been explored for the timetable scheduling problem. One direction of research is based on event-driven models where train actions are defined as different events with predefined rules determining the orders of events. In [28], the timetable was formulated as an alternative graph model, and a branch-and-bound algorithm was proposed to find solutions efficiently. Based on the alternative graph model, a tabu search algorithm was proposed to reroute trains in [24]. In [143], the interaction between train speeds and headway under the quasi-moving block system was considered, when rescheduling high-speed trains based on the alternative graph model. The timetable scheduling problem can also be formulated through an event-activity network (a directed graph), which can be used to minimize the total weighted train delay and the number of canceled trains [154], to optimize passengers' routes [158], and to integrate passenger reassignment and timetable scheduling [159]. Furthermore, max-plus models [44] and switching max-plus-linear models [54; 125] have also been used to efficiently generate efficient timetables; as the models make use of properties from max-plus algebra, the resulting problem can be reduced efficiently, and less time is required to get the solution.

Another important direction of research is based on time-driven models, where train actions are formulated with respect to time constraints. Time-driven models are widely used in the literature as they can directly include different factors in railway traffic, such as passenger demands, train speeds, and energy consumption. In [117], the timetable and the train speed profile of one urban rail line with several stations were jointly optimized within a bi-level scheme, where a numerical approach was proposed to allocate the total time to each section, given the optimal speed profile of a fixed running time for each section. In [26], it was indicated that the timetable can be optimized in real

time with a closed-loop control framework by predicting the traffic conditions through the real-time train positions and speed profiles information. In [86; 87], the timetable and train speed profile were integrally optimized by a mixed-integer nonlinear programming (MINLP) approach, a mixed-integer linear programming (MILP) approach, and a simplified MILP approach considering different train speed profile options. In [22], the rescheduling of large-scale railway traffic networks was formulated as a bi-level MILP problem, and an MPC scheme was applied to handle disruptions and disturbances in real time.

4.2.2. PASSENGER-ORIENTED TIMETABLE SCHEDULING

In recent decades, many studies have focused on passenger-oriented timetable scheduling, where passenger demands are explicitly taken into account to provide high-quality services for passengers. In [18], a nonlinear integer programming model was proposed to optimize arrival and departure times of trains with the objective of minimizing operational costs and passenger waiting times. In [134], the train speed and stop-skipping were incorporated into the timetable scheduling problem to minimize the energy consumption and the passenger travel time, and a bi-level approach was proposed to solve the resulting MINLP problem. Furthermore, an iterative convex programming approach was developed to improve the computational speed in [135]. In [97], an MINLP problem was formulated to minimize passenger waiting time with consideration of time-varying passenger demands. In [152], a Lagrangian relaxation-based heuristic timetable scheduling algorithm was proposed to minimize passenger waiting times and operational costs by using a space-time network. An integer linear programming problem was formulated to jointly optimize the timetable and passenger flow control strategies for an over-saturated railway line in [112]; then, a hybrid algorithm was developed to solve the resulting optimization problem. However, most research only focuses on the timetable scheduling of a single *line*, and hence leaving an open gap for improving the operational performance of *urban rail transit networks*.

Passenger-oriented timetable scheduling of urban rail networks is more challenging than that of a single line as different lines will interact with each other through the transfer passengers. An urban rail network including time-dependent passenger OD demands was modeled as a detailed event-driven model in [136], and then the passenger travel time and the train energy consumption were collaboratively optimized. Furthermore, the event-driven model was extended as a disruption management model for an integrated disruption management problem with the objective of recovering the impacted timetable and minimizing passenger waiting times in [9]. In [19], an MINLP model was proposed to optimize line frequencies and capacities in railway rapid transit networks; the objective of that paper was to minimize operational costs and passenger trip time and transfer time given a certain OD matrix. In [150], feasible passengers routes in the urban rail network were defined through a directed graph, so that the passenger OD demands and the transfer actions can be included explicitly; then, a decomposed adaptive large-neighborhood search method was proposed to minimize the number of waiting passengers in the busiest station. However, incorporating time-dependent passenger OD demands in the urban rail network timetable scheduling problem is still a challenging task because of the network's size, high non-linearity of the problem, and the

large computational burden. Accurate models for urban rail networks that include time-dependent passenger OD demands and fast solution methods for passenger-oriented timetable scheduling are urgently needed for real-time timetable scheduling.

4.2.3. MPC FOR RAILWAY TRAFFIC MANAGEMENT

As an efficient real-time control approach for constrained systems, MPC has been applied in railway timetable scheduling problems to optimize and adjust the timetable in real time. In [125], MPC was used for railway timetable scheduling based on the switching max-plus-linear models to minimize train delays and operational costs of breaking connections or changing the order of trains. Furthermore, the switching max-plus-linear model-based timetable scheduling problem was solved in a distributed manner to handle large-scale cases [54]. In [15], an MPC approach was proposed to cope with train rescheduling problems in the complex station areas. MPC was also used in railway traffic management in case of disruptions, and the MPC optimization problem was transformed into an MILP problem to reduce the computational burden [22]. A hierarchical MPC approach was proposed for real-time high-speed railway delay management and train control problem, where the train delay was minimized at the upper level while the detailed train speed control was conducted at the lower level [138]. The optimization problem in both levels of the hierarchical MPC approach were also formulated as MILP problems to increase the online feasibility. The existing literature indicates that the online computational burden of the MPC optimization problem must be reduced for real-time scheduling of large-scale railway networks. The problem is even more challenging when taking time-dependent passenger OD demands into account.

This chapter proposes a novel timetable scheduling model which can take time-dependent passenger OD demands into account. An MPC approach is then proposed for real-time timetable scheduling. Based on the proposed model the MPC optimization problem can be easily transformed into an MILP problem, to overcome computational complexity issues.

4.3. PASSENGER-ORIENTED REAL-TIME TIMETABLE SCHEDULING MODEL

In this section, we propose a novel model for passenger-oriented real-time timetable scheduling in urban rail traffic networks. Some general explanations and assumptions adopted for the model formulation throughout this chapter are as follows:

1. Since the number of passengers is very large, the approximation error of treating it as a real-valued variable is relatively small. Hence, variables indicating the number of passengers are regarded as real-valued variables.
2. The chapter focuses on optimizing arrival and departure times of trains, and hence, short-turning, stop-skipping, and rolling stock circulation are not considered.
3. A platform can only accommodate one train at a time, and the order of trains at a platform is fixed.

The notations used in this chapter are introduced in Section 4.3.1. Then, the simplified passenger flow model is proposed in Section 4.3.2. In Section 4.3.3, the train operation model related to the simplified passenger flow model is given.

4.3.1. NOTATIONS

Notations	Definition
j	Index of stations, $j \in S$, S is the set of stations
p	Index of platforms
k_p	Index of cycles at platform p ; also indicating the train visiting platform p at cycle k_p
$s^{pla}(p)$	Successor platform of platform p
$p^{pla}(p)$	Predecessor platform of platform p

Table 4.1: Sets and Indices

Notations	Definition
$c_p(k_p)$	Length of cycle k_p at platform p
$L_p(k_p)$	Starting time of cycle k_p at platform p
$r_p^{\min}(k_p)$	Minimum running time of train from platform p to its successor platform at cycle k_p
$r_p^{\max}(k_p)$	Maximum running time of train from platform p to its successor platform at cycle k_p
τ_p^{\min}	Minimum dwell time of train at platform p
h_p^{\min}	Minimum headway of platform p
$\lambda_{j,m}^{\text{station}}(k_p)$	Passenger arrival rate at station j with station m as their destination at cycle k_p
$\beta_{j,p,m}$	Splitting rate of passengers at station j who are assigned to platform p with destination m as their destination
$\beta_{p,q,m}^{\text{train}}$	Transfer rate of passengers from platform p to platform q with station m as their destination
$\theta_{p,q}^{\text{trans}}$	Average walking time for passengers walking from platform p to platform q
$\theta_{p,q}^{\text{duration}}$	Duration time for the transfer process from platform p to platform q

Table 4.2: Input Parameters

Notations	Definition
$a_p(k_p)$	Arrival time of train at cycle k_p of platform p
$d_p(k_p)$	Departure time of train at cycle k_p of platform p

Table 4.3: Decision variables

4.3.2. SIMPLIFIED PASSENGER FLOW MODEL

The passenger origin-destination demands can be described as a time-varying matrix, and the element of the matrix is denoted as $\lambda_{j,m}^{\text{station}}(t)$, with j and m indicating the origin and destination stations, respectively. Passengers usually care about whether there are regular departures at a platform so that they can plan their journey easily and do not have to wait too long for the next train if they missed the current train. A train only visits a platform at a certain time period, and the passenger arrival rate generally does not change significantly during a short time period. Therefore, at each platform, we discretize the planning time window into several time intervals of equal length, where

Notations	Definition
$r_p(k_p)$	Running time of train from platform p to its successor platform $s^{\text{pla}}(p)$ in cycle k_p
$\tau_p(k_p)$	Dwell time of train at cycle k_p of platform p
$\lambda_{p,m}(k_p)$	Passenger arrival rate at platform p with station m as their destination at cycle k_p
$n_{p,m}(k_p)$	Number of passengers with station m as their destination waiting at platform p immediately after time $k_p c_p$
$n_{p,m}^{\text{arrive,new}}(k_p)$	Number of passengers outside the urban rail network with destination m arriving at platform p at cycle k_p
$n_{p,m}^{\text{arrive,trans}}(k_p)$	Number of transfer passengers with destination m arriving at platform p at cycle k_p
$n_{p,m}^{\text{before}}(k_p)$	Number of passengers at platform p with station m as their destination immediately before the departure of train k_p
$n_{p,m}^{\text{board}}(k_p)$	Number of passengers with station m as their destination boarding on the train at cycle k_p
$n_{p,m}^{\text{depart}}(k_p)$	Number of passengers on train k_p departing from platform p with station m as their destination
$n_{p,m}^{\text{after}}(k_p)$	Number of passengers at platform p with station m as their destination immediately after the departure of train k_p
$n_{p,q,m}^{\text{trans}}(k_p)$	Number of passengers alighting from train k_p of platform p who want to transfer to platform q with station m as their destination
$n_{p,m}^{\text{remain}}(k_p)$	Number of passengers who continue to stay on train k_p after the alighting process
$n_{p,m}^{\text{alight}}(k_p)$	Number of passengers with station m as their destination alighting from train k_p at platform p

Table 4.4: Output variables

every time interval includes one and only one arrival-departure pair of a train at the same platform so as to provide reliable service for passengers. In addition, we assume the passenger arrival rate is constant in each time interval. In the sequel, we refer to these time intervals as cycles. The cycle time for a given platform is then the length of the cycle for that platform¹. The cycle times for platform p and platform q , which are represented by $c_p(k_p)$ and $c_q(k_p)$ respectively, can be different from each other.

The passenger arrival rate $\lambda_{p,m}^{\text{original}}(t)$ at platform p with station m as destination is determined by

$$\lambda_{p,m}^{\text{original}}(t) = \beta_{j,p,m} \lambda_{j,m}^{\text{station}}(t), \forall p \in P_j, \forall m \in S, \quad (4.1)$$

where P_j defines a set of platforms at station j ; S is the set of stations in the urban rail network; $\beta_{j,p,m}$ is the splitting rate of passengers at station j who are assigned to platform p with destination m as their destination, $\sum_{p \in P_j} \beta_{j,p,m} = 1, \forall m \in S$, and $\beta_{j,p,m}$ can be

obtained based on the historical data.

Fig. 4.1 illustrates the procedure of approximating the original passenger arrival rate for the simplified passenger flow model, where k_p represents the index of the cycle at platform p , and the approximated arrival rate can be calculated by:

$$\lambda_{p,m}(k_p) = \frac{1}{c_p(k_p)} \int_{L_p(k_p)}^{L_p(k_p) + c_p(k_p)} \lambda_{p,m}^{\text{original}}(t) dt, \quad (4.2)$$

¹The cycle time at a platform can be equal to the expected departure-departure headway of the basic timetable. Then, we can adjust departure and arrival times to further improve the basic timetable based on the detailed passenger demands. We can also generate the expected departure-departure headway by a higher-level controller; for more details, we refer to our recent work [74; 79] (see also Chapter 2).

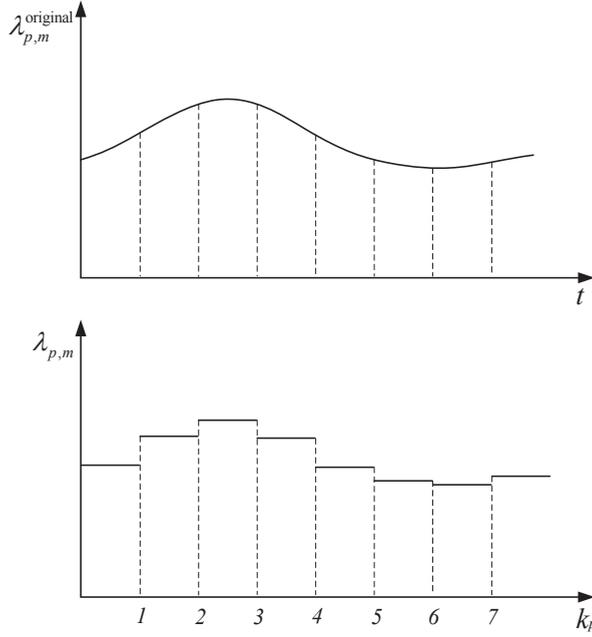


Figure 4.1: Illustration of approximating passenger arrival rate.

where $\lambda_{p,m}^{\text{original}}(t)$ represents the original passenger arrival rate, $L_p(k_p)$ represents the starting time of cycle k_p , and $c_p(k_p)$ is the length of cycle k_p . By introducing the cycle time, the computational efficiency for calculating passenger-related factors can be significantly improved. Note that the approximation can be conducted offline to reduce the online computational burden.

According to the definition of cycle, only one train would visit platform p at cycle k_p ; therefore, in this chapter, for the sake of simplification, we use “train k_p ” to represent the train visiting platform p at cycle k_p .

At each cycle, the number of passengers waiting at the platform is updated as some passengers have boarded on a train and departed from the platform. The number of passengers waiting at platform p is updated at every cycle, according to the new arriving passengers $n_{p,m}^{\text{arrive,new}}(k_p)$ from outside the station, the transfer passengers $n_{p,m}^{\text{arrive,trans}}(k_p)$ from other lines, and the boarding passengers $n_{p,m}^{\text{board}}(k_p)$, by

$$n_{p,m}(k_p + 1) = n_{p,m}(k_p) + n_{p,m}^{\text{arrive,new}}(k_p) + n_{p,m}^{\text{arrive,trans}}(k_p) - n_{p,m}^{\text{board}}(k_p), \quad (4.3)$$

where $n_{p,m}(k_p)$ denotes the number of passengers with station m as their destination waiting at platform p at the beginning of cycle k_p .

As depicted in Fig. 4.1, in each cycle, the passenger arrival rate is regarded as constant, and the number of new passengers $n_{p,m}^{\text{arrive,new}}(k_p)$ arriving at platform p with destination m between k_p and $k_p + 1$ can be calculated based on the passenger arrival rate:

$$n_{p,m}^{\text{arrive,new}}(k_p) = c_p(k_p)\lambda_{p,m}(k_p), \quad (4.4)$$

where $\lambda_{p,m}(k_p)$ is the passenger arrival rate at platform p with station m as their destination at cycle k_p .

Define $\theta_{q,p}^{\text{trans}}$ as the average walking time for passengers walking from platform q to platform p , $a_p(k_p)$ and $d_p(k_p)$ as the arriving and departure times of train k_p at platform p , respectively. Then, we introduce a binary variable $y_{k_q,q,k_p,p}$ to represent the connection of trains at a transfer station:

$$y_{k_q,q,k_p,p} = \begin{cases} 1, & \text{if } d_p(k_p-1) < a_q(k_q) + \theta_{q,p}^{\text{trans}} \leq d_p(k_p); \\ 0, & \text{otherwise,} \end{cases} \quad (4.5)$$

with $y_{k_q,q,k_p,p} = 1$ denoting that passengers from train k_q of platform q connect to train k_p of platform p , i.e., passengers from train k_q at platform q could arrive at platform p between the departure of train $k_p - 1$ and k_p ; otherwise, when $y_{k_q,q,k_p,p} = 0$, the passengers from train k_q at platform q cannot connect to train k_p at platform p .

With $y_{k_q,q,k_p,p}$ defined as in (4.5), the number of passengers $n_{p,m}^{\text{arrive,trans}}(k_p)$ transferring from other platforms of station j and arriving at platform p before the departure of train k_p can be calculated by

$$n_{p,m}^{\text{arrive,trans}}(k_p) = \sum_{q \in \text{plat}(p)} \sum_{k_q \in \mathcal{N}_q} y_{k_q,q,k_p,p} n_{q,p,m}^{\text{trans}}(k_q), \quad (4.6)$$

where $\text{plat}(p)$ is the set of the platforms at the same station as platform p , and \mathcal{N}_q collects the indices of all the cycles of platform q .

Then, the number of passengers $n_{p,m}^{\text{before}}(k_p)$ at platform p with station m as their destination immediately before the departure of train k_p can be computed by

$$n_{p,m}^{\text{before}}(k_p) = n_{p,m}(k_p) + (d_p(k_p) - L_p(k_p))\lambda_{p,m}(k_p) + n_{p,m}^{\text{arrive,trans}}(k_p), \quad (4.7)$$

Then, the total number of passengers $n_p^{\text{before}}(k_p)$ waiting at platform p immediately before the departure of train k_p is

$$n_p^{\text{before}}(k_p) = \sum_{m \in S} n_{p,m}^{\text{before}}(k_p). \quad (4.8)$$

The total number of passengers $n_p^{\text{board}}(k_p)$ boarding the train at cycle k_p can be computed by

$$n_p^{\text{board}}(k_p) = \min\left(C_{\max,k_p} - n_p^{\text{remain}}(k_p), n_p^{\text{before}}(k_p)\right), \quad (4.9)$$

where C_{\max,k_p} represents the capacity of train k_p at platform p , and $n_p^{\text{remain}}(k_p)$ is the number of passengers remaining on train k_p after the alighting process at platform p .

Therefore, the number of passengers $n_p^{\text{after}}(k_p)$, who cannot board train k_p , waiting at platform p immediately after train k_p departs can be computed by

$$n_p^{\text{after}}(k_p) = n_p^{\text{before}}(k_p) - n_p^{\text{board}}(k_p). \quad (4.10)$$

If we define

$$\lambda_p(k_p) = \sum_{m \in S} \lambda_{p,m}(k_p), \quad (4.11)$$

then the number of passengers who cannot board train k_p at platform p with different destinations can be calculated by

$$n_{p,m}^{\text{after}}(k_p) = n_p^{\text{after}}(k_p) \frac{\lambda_{p,m}(k_p)}{\lambda_p(k_p)}, \quad (4.12)$$

which means the proportion of waiting passengers with different destinations, who cannot board train k_p at platform p , is assumed not to change significantly compared with the proportion of passengers arriving in the current cycle. As $\lambda_{p,m}(k_p)$ is defined as a known constant, $n_{p,m}^{\text{after}}(k_p)$ can be computed linearly.

Then, the number of boarding passengers $n_{p,m}^{\text{board}}(k_p)$ with destination m can be computed by

$$n_{p,m}^{\text{board}}(k_p) = n_{p,m}^{\text{before}}(k_p) - n_{p,m}^{\text{after}}(k_p). \quad (4.13)$$

When train k_p arrives at platform p , the number of passengers $n_{p,q,m}^{\text{trans}}(k_p)$ with station m as their destination on train k_p transferring from platform p to platform q can be calculated by

$$n_{p,q,m}^{\text{trans}}(k_p) = \beta_{p,q,m}^{\text{train}} n_{p^{\text{pla}}(p),m}^{\text{depart}}(k_p), \forall q \in \text{plat}(p) \setminus \{p\}, \quad (4.14)$$

where $n_{p^{\text{pla}}(p),m}^{\text{depart}}(k_p)$ denotes the number of passengers with destination m on train k_p immediately after the train departure from the predecessor platform $p^{\text{pla}}(p)$ of platform p , and $\beta_{p,q,m}^{\text{train}}$ is the transfer rate of passengers on train k_p , transferring from platform p to $q \in \text{plat}(p)$ with destination m immediately after arrival at platform p , and

$$\sum_{q \in \text{plat}(p)} \beta_{p,q,m}^{\text{train}} = 1. \quad (4.15)$$

The transfer rate of passengers can be obtained based on the historical data or by a shortest path algorithm, e.g., Yen's algorithm [149], assuming that passengers select the platform corresponding to the shortest path to reach their destination.

Remark 4.1. It is worth noting that $\beta_{p,p,m}^{\text{train}}$ denotes the proportion of passengers with m as their destination remaining on train k_p at platform p after the alighting process, i.e., no transfer behavior is needed; thus, we have $n_{p,p,m}^{\text{trans}}(k_p) = 0$. In particular, If the arrival station is not a transfer station, then $\beta_{p,p,m}^{\text{train}} = 1$.

Remark 4.2. Define $\text{sta}(p)$ as the station corresponding to platform p . For passengers whose destination is the arrival station, i.e., $j = \text{sta}(p)$, we set $\beta_{p,p,j}^{\text{train}} = 1$ and $\beta_{p,q,j}^{\text{train}} = 0, \forall q \in \text{plat}(p) \setminus \{p\}$, which means passengers who have arrived at their destination will directly exit the station j from platform p without any transfer behavior, and we have $n_{p,q,j}^{\text{trans}}(k_p) = 0, \forall q \in \text{plat}(p)$.

The number of passengers $n_{p,m}^{\text{remain}}(k_p)$ remaining on the train at platform p in cycle k_p with destination m after the alighting process can be calculated by

$$n_{p,m}^{\text{remain}}(k_p) = \beta_{p,p,m}^{\text{train}} n_{p^{\text{pla}}(p),m}^{\text{depart}}(k_p), \forall m \in S \setminus \{\text{sta}(p)\}. \quad (4.16)$$

In other words, $n_{p,m}^{\text{remain}}(k_p)$ represents the number of passengers who continue to stay on train k_p after the alighting process. In particular, passengers, who have arrived at their destination station when train k_p arrives at platform p , will alight from the train directly, i.e., no passengers with destination $\text{sta}(p)$ will remain on train k_p after arriving at station $\text{sta}(p)$, $n_{p,\text{sta}(p)}^{\text{remain}}(k_p) = 0$.

Having (4.16), the total number of passengers $n_p^{\text{remain}}(k_p)$ remaining on train k_p at platform p after the alighting process can be calculated by

$$n_p^{\text{remain}}(k_p) = \sum_{m \in S} n_{p,m}^{\text{remain}}(k_p). \quad (4.17)$$

Then, the number of passengers $n_{p,m}^{\text{depart}}(k_p)$ with station m as their destination, who will depart from platform p at time k_p , can be computed by

$$n_{p,m}^{\text{depart}}(k_p) = n_{p,m}^{\text{remain}}(k_p) + n_{p,m}^{\text{board}}(k_p). \quad (4.18)$$

The total number of passengers $n_p^{\text{depart}}(k_p)$, who will depart from platform p at time k_p , can be calculated by

$$n_p^{\text{depart}}(k_p) = \sum_{m \in S} n_{p,m}^{\text{depart}}(k_p). \quad (4.19)$$

The total number of passengers $n_p^{\text{alight}}(k_p)$ alighting from train k_p at platform p can be calculated by

$$n_p^{\text{alight}}(k_p) = n_{p^{\text{pla}}(p)}^{\text{depart}}(k_p) - n_p^{\text{remain}}(k_p), \quad (4.20)$$

where $n_{p^{\text{pla}}(p)}^{\text{depart}}(k_p)$ denotes the total number of passengers on board of train k_p departing from the predecessor platform $p^{\text{pla}}(p)$ of platform p .

4.3.3. TRAIN OPERATION MODEL

In this chapter, we assume the order of trains at each platform is fixed, and the aim is to generate departure and arrival times by incorporating the detailed time-dependent passenger OD demands of the urban rail network to further improve passenger satisfaction. In this context, for a general urban rail transit timetable scheduling problem, the operation of trains can be described by arrival times, dwell times, departure times, and running times. These variables interact with each other by several constraints to guarantee the conflict-free and efficient traffic operation.

Based on the definition of the cycle, we can generate the lower and upper bounds of each cycle according to the expected departure-departure headway. Then, the arrival and departure times of train k_p at platform p should satisfy

$$L_p(k_p) < a_p(k_p) < d_p(k_p) \leq L_p(k_p) + c_p(k_p), \quad (4.21)$$

where $L_p(k_p)$ is the starting time of cycle k_p at platform p , and $c_p(k_p)$ is the length of cycle k_p ; $a_p(k_p)$ and $d_p(k_p)$ represent the arrival time and the departure time of train k_p at platform p , respectively.

The dwell time $\tau_p(k_p)$ of train k_p at platform p can be calculated by

$$\tau_p(k_p) = d_p(k_p) - a_p(k_p), \quad (4.22)$$

and $\tau_p(k_p)$ should be constrained by

$$\tau_p(k_p) \geq \tau_p^{\min}, \quad (4.23)$$

where τ_p^{\min} is the minimum dwell time.

Then, the arrival time of train k_p at platform p is also constrained by the departure-arrival headway constraint

$$a_p(k_p) \geq d_p(k_p - 1) + h_p^{\min}, \quad (4.24)$$

where $d_p(k_p - 1)$ is the departure time of train $(k_p - 1)$ at platform p , and h_p^{\min} is the minimum headway between two successive trains at platform p .

The arrival time of train k_p at the successor platform $s^{\text{pla}}(p)$ of platform p is

$$a_{s^{\text{pla}}(p)}(k_p) = d_p(k_p) + r_p(k_p), \quad (4.25)$$

where $r_p(k_p)$ represents the running time of train k_p from platform p to platform $s^{\text{pla}}(p)$, and $r_p(k_p)$ should be constrained by

$$r_p^{\min}(k_p) \leq r_p(k_p) \leq r_p^{\max}(k_p), \quad (4.26)$$

where $r_p^{\max}(k_p)$ and $r_p^{\min}(k_p)$ are maximal and minimal running time of train k_p from platform p to $s^{\text{pla}}(p)$, respectively. The minimum running time is limited by the condition of the line, speed limit, and train characteristics, and the maximum running time is determined by the operational requirement.

4.4. MPC FOR PASSENGER-ORIENTED TIMETABLE RESCHEDULING

Model predictive control is a control method that repeatedly solves finite-horizon optimization problems and implements optimized decisions in a moving horizon manner [92]. In the MPC scheme, the current control action is obtained by solving an optimization problem over a finite-horizon window. The optimization yields a control sequence, but only the first control action is implemented in the real system. At the next control step, the optimization is conducted again using updated state information and with a shifted finite-horizon window. This moving horizon optimization procedure is repeated until the end of the overall control period.

In this chapter, the control time interval of each platform is defined as the cycle time of the platform. Given the train is assumed to run from the starting platform to the terminal platform of a line, the cycle times of all platforms of a line are identical. As cycle times can be different for different lines, we introduce control time interval T_{ctrl} , and the control time step is indexed as κ . The number of cycles included in one time step for different platforms can be different. The MPC method can be described by the following three elements:

1. *Prediction model.*

The passenger-oriented urban rail traffic network model developed in Section 4.3 can be used as the prediction model for the MPC controller. The model is a non-linear model, and, for each cycle, it can be represented as follows:

$$n_{p,m}(k_p + 1) = f\left(n_{p,m}(k_p), n_{q,p,m}^{\text{trans}}(k_q), g_p(k_p)\right), \quad (4.27)$$

where $n_{p,m}(k_p)$ is the number of passengers waiting at platform p with station m as their destination at the beginning of cycle k_p ; $n_{q,p,m}^{\text{trans}}(k_q)$ represents the number of passengers transferring from other platforms (denotes as q) at the same station; $g_p(k_p)$ collects the decision variables including arrival and departure times of trains at cycle k_p of platform p .

2. *Optimization problem.*

The waiting time of passengers at the platform is an important criterion to evaluate passenger satisfaction. Furthermore, to further improve passenger satisfaction a penalty factor is added for passengers who cannot board a train because of the train capacity. Hence, in this chapter, an objective function of the following form is considered:

$$J = \sum_{p \in P} \sum_{k_p \in \mathcal{N}_p(\kappa)} \left(n_p^{\text{before}}(k_p) c_p(k_p) + \xi n_p^{\text{after}}(k_p) c_p(k_p) \right), \quad (4.28)$$

where $\mathcal{N}_p(\kappa)$ is the set indices of trains visiting platform p within the prediction window starting at control step κ , P denotes the set of platforms of the considered urban rail network; $n_p^{\text{before}}(k_p)$ and $n_p^{\text{after}}(k_p)$ represent the number of passengers waiting at platform p immediately before the departure of train k_p and immediately after the departure of train k_p , respectively, and ξ is a non-negative weight.

Generally speaking, passengers waiting at a platform consist of two classes of passengers, i.e., passengers who cannot board the previous train and the new arrival passengers. For all the passengers waiting at the platform, the largest waiting time is the time interval between two adjacent departure times, therefore the first term in (4.28) is used as the cost function of total passenger waiting time, which, loosely speaking, provides an upper bound of the passenger waiting time. The passengers who cannot board the train have to stay at the platform and wait for the next train, so a penalty factor $n_p^{\text{after}}(k_p) c_p(k_p)$ is employed to make the trains carry as many passengers as possible.

Therefore, the optimization problem for MPC in each control step is

$$\begin{cases} \min_{\mathbf{g}(\kappa)} J := \sum_{p \in P} \sum_{k_p \in \mathcal{N}_p(\kappa)} \left(n_p^{\text{before}}(k_p) + \xi n_p^{\text{after}}(k_p) \right) c_p(k_p), \\ \text{s.t.} \quad (4.1) - (4.14), (4.16) - (4.26), \end{cases} \quad (4.29)$$

where $\mathbf{g}(\kappa)$ collects all decision variables $g_p(k_p | \kappa)$ for all platform p and all $k_p \in \mathcal{N}_p(\kappa)$.

3. Moving horizon optimization.

Solving the optimization problem (4.29) results in a sequence of decision variables represented by $\mathbf{g}(\kappa)$, and only the decision variables at the current time step are implemented to the real-life urban rail network. At the next control time step $\kappa + 1$, the time window is shifted for one step, and the optimization problem is solved again based on the new information collected from the urban rail network. The procedure of the closed-loop control scheme is shown in Fig. 4.2.

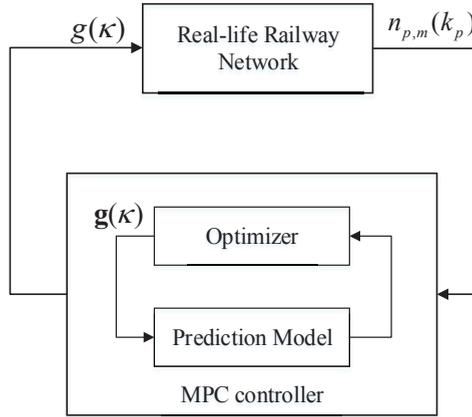


Figure 4.2: MPC for passenger-oriented timetable scheduling.

As the length of cycle time at a platform can be equal to the departure headway of a basic timetable, cycle times that can ensure constraint satisfaction of problem (4.29) can always be found, i.e., a feasible solution is always available if we use the basic timetable. Therefore, the recursive feasibility of MPC can be ensured.

4.5. SOLUTION APPROACHES

The resulting optimization problem in Section 4.4 is a nonlinear non-convex problem because of (4.5), (4.6), and (4.9). The problem can be solved by nonlinear optimization approaches, e.g., sequential quadratic programming approach. In order to increase the online feasibility of the problem, the MPC optimization problem is formulated as a mixed-integer linear programming (MILP) problem and a simplified mixed-integer linear programming (SMILP) problem, which can be solved efficiently by existing solvers.

4.5.1. SEQUENTIAL QUADRATIC PROGRAMMING APPROACH

Sequential quadratic programming (SQP) approach is a gradient-based nonlinear programming approach, which is widely used in many fields to solve nonlinear optimization problems [10]. In SQP, a sequence of quadratic programming problems is solved to get descent directions of the original problem. The objective function and the con-

straints of the optimization problem should be continuously differentiable when applying the SQP algorithm. In this chapter, the optimization problem has some points of non-smoothness due to the min function in (4.9). As the optimal solution is generally not obtained at the points of non-smoothness, the SQP approach can jump over these points. Since the SQP algorithm might obtain a local optimal solution when handling non-convex problems, multi-start SQP is used to improve the solution quality of SQP in this chapter.

4.5.2. MIXED-INTEGER LINEAR PROGRAMMING APPROACH

In this section, the MPC optimization problem is transformed into an MILP problem, by introducing auxiliary binary variables to handle the nonlinear terms in (4.5), (4.6), and (4.9).

In order to transform (4.5) into a mixed logical dynamical (MLD) system [7], the time checking binary variable $x_{k_q, q, k_p, p}$ is introduced as

$$x_{k_q, q, k_p, p} = \begin{cases} 1, & \text{if } a_q(k_q) + \theta_{q,p}^{\text{trans}} \leq d_p(k_p); \\ 0, & \text{otherwise,} \end{cases} \quad (4.30)$$

where $a_q(k_q)$ is the arrival time of train k_q at platform q , $\theta_{q,p}^{\text{trans}}$ represents the average transfer time from platform q to platform p , and $d_p(k_p)$ denotes departure time of train k_p at platform p .

We define M_t and m_t as the maximum and minimum value of the departure (arrival) time, which are finite as we consider problems in a finite time window². Then, (4.30) is equivalent to

$$\begin{cases} a_q(k_q) + \theta_{q,p}^{\text{trans}} - d_p(k_p) \leq (1 - x_{k_q, q, k_p, p})(M_t - d_p(k_p)); \\ a_q(k_q) + \theta_{q,p}^{\text{trans}} - d_p(k_p) \geq \varepsilon + x_{k_q, q, k_p, p}(m_t - d_p(k_p) - \varepsilon), \end{cases} \quad (4.31)$$

where ε is a sufficient small number (generally the machine precision) [7]. Define

$$y_{k_q, q, k_p, p} = x_{k_q, q, k_p, p} - x_{k_q, q, k_{p-1}, p}. \quad (4.32)$$

Then, based on Lemma 4.3, (4.5) is equivalent to (4.31) and (4.32).

Lemma 4.3. Given $y_{k_q, q, k_p, p} = x_{k_q, q, k_p, p} - x_{k_q, q, k_{p-1}, p}$, $d_p(k_{p-1}) < a_q(k_q) + \theta_{q,p}^{\text{trans}} \leq d_p(k_p)$ holds if and only if $y_{k_q, q, k_p, p} = 1$; otherwise, $y_{k_q, q, k_p, p} = 0$.

Proof. From the definition of $x_{k_q, q, k_p, p}$ in (4.30), we have $x_{k_q, q, k_p, p} \geq x_{k_q, q, k_{p-1}, p}$. Then, we have the following three situations based on the value of $a_q(k_q) + \theta_{q,p}^{\text{trans}}$:

if $a_q(k_q) + \theta_{q,p}^{\text{trans}} > d_p(k_p)$, we have $x_{k_q, q, k_p, p} = 0$ and $x_{k_q, q, k_{p-1}, p} = 0$; then, $y_{k_q, q, k_p, p} = 0$;
if $d_p(k_{p-1}) < a_q(k_q) + \theta_{q,p}^{\text{trans}} \leq d_p(k_p)$, we have $x_{k_q, q, k_p, p} = 1$ and $x_{k_q, q, k_{p-1}, p} = 0$; then, $y_{k_q, q, k_p, p} = 1$;
if $a_q(k_q) + \theta_{q,p}^{\text{trans}} \leq d_p(k_{p-1})$, we have $x_{k_q, q, k_p, p} = 1$ and $x_{k_q, q, k_{p-1}, p} = 1$; then, $y_{k_q, q, k_p, p} = 0$. \square

²The value of M_t can be the length of the planning time window, i.e., $M_t = t_{\text{end}}$, and m_t can be equal to 0.

The min function in (4.9) can be handled by introducing the auxiliary binary variable $\delta_{k_p,p}^{\text{board}}$ and the auxiliary real variable $f_{k_p,p}$. Define

$$f_{k_p,p} = \left(C_{\max,k_p} - n_p^{\text{remain}}(k_p) \right) - n_p^{\text{before}}(k_p), \quad (4.33)$$

Then, the expression $\delta_{k_p,p}^{\text{board}} = 1 \Leftrightarrow f_{k_p,p} \leq 0$ is equivalent to

$$\begin{cases} f_{k_p,p} \leq M_p (1 - \delta_{k_p,p}^{\text{board}}), \\ f_{k_p,p} \geq \varepsilon + (m_p - \varepsilon) \delta_{k_p,p}^{\text{board}}, \end{cases} \quad (4.34)$$

where M_p and m_p are the maximum value and the minimum value of $f_{k_p,p}$, respectively³. Having (4.34), the expression (4.9) is equivalent to

$$n_p^{\text{board}}(k_p) = \delta_{k_p,p}^{\text{board}} \left(C_{\max,k_p} - n_p^{\text{remain}}(k_p) \right) + \left(1 - \delta_{k_p,p}^{\text{board}} \right) n_p^{\text{before}}(k_p). \quad (4.35)$$

After introducing auxiliary variables in (4.30) and (4.34), we still have nonlinear terms, i.e., the product of binary variables and real variables in (4.6), (4.31), and (4.35). The product of binary variables and real variables can be transformed into linear inequalities by introducing some auxiliary variables by using the method presented in [7; 139]. The details of the transformation can be found in *Transformation property 2.2*.

In summary, we introduce three equivalence transformations, i.e., (4.5) with (4.31)-(4.32), (4.9) with (4.33)-(4.35), and *Transformation property 2.2*. The proof for "(4.5) is equivalent to (4.31)-(4.32)" is provided in *Lemma 4.3*. The other two transformations can be found in [7; 139]. Based on the above transformations, we can finally obtain an MILP problem that is exactly equivalent to the original optimization problem.

4.5.3. SIMPLIFIED MIXED-INTEGER LINEAR PROGRAMMING APPROACH

In Section 4.5.2, several auxiliary variables and constraints are introduced to handle the train capacity constraints in (4.9) which calculates the possible number of boarding passengers at a platform. These constraints play an important role in accurately calculating the number of passengers in peak hours, when there are a large number of passengers waiting at platforms. During the peak hours, not all passengers can board the current train, and, instead, some passengers must wait for the next train at the platform. However, in off-peak hours, the number of passengers waiting at the platform is relatively small, and almost all passengers can board the current train upon their arrival. In this case, we can disregard the train capacity constraints in (4.9), and hence the constraints (4.33), (4.34), and (4.35) are not required. Therefore, we can further reduce the computational burden.

With this simplification, the number of passengers who can board the train at cycle k_p is equal to the number of waiting passengers, i.e., (4.9) will be replaced with:

$$n_p^{\text{board}}(k_p) = n_p^{\text{before}}(k_p). \quad (4.36)$$

³The value of M_p can be a very large value related to train capacities, i.e., $M_p = 10 \cdot C_{\max,k_p}$, and m_p can be a small value, i.e., $m_p = -10 \cdot C_{\max,k_p}$.

The simplification results in a simplified mixed-integer linear programming (SMILP) problem.

As mentioned in Section 4.5.1, the SQP algorithm might get stuck in a local optimal solution when handling non-convex problems. In this context, several starting points are required for SQP, so as to improve the solution quality. The simplified problem is solved by disregarding train capacity constraints, and other constraints are identical with the original MILP problem. Therefore, instead of doing multi-start SQP, the SMILP formulation can be used to get an initial solution; then, this initial solution is employed as the starting point of SQP for the original nonlinear optimization problem.

4.6. CASE STUDY

In this section, simulations are performed to evaluate the effectiveness of the developed passenger-oriented urban rail traffic model and the MILP-based MPC approach. We first simulate the urban rail network using the proposed model and the model in [9; 136] based on the real-life operation data of part of the Beijing metro network, and simulation results are used to test the accuracy of the proposed model. Then, numerical experiments are designed to test the performance of the solution approaches and the corresponding MPC controller.

4.6.1. ASSESSMENT OF THE PROPOSED MODEL

To the best of the authors' knowledge, there is no commonly recognized accurate model for passenger-oriented urban rail networks, and the most accurate model we found in the literature is the model in [9; 136]. Therefore, in this chapter, we define the model in [9; 136] as an "accurate model" to simulate the real-life urban network and to test the accuracy of our model.

The real-life network we use is shown in Fig. 4.3. The network contains two bi-directional lines that consist of 19 stations and 40 platforms. The passenger OD data used for the case study are obtained based on the real-life entering and exiting passenger flows of automatic fare collection systems. The passenger flows over each half-hour are recorded and stored. In the real-life data used for the case study, passenger arrival rates in different stations have different dynamics. The lines we use contains both normal and over-saturated lines. For the simulation, we use the real-life passenger data from the Beijing Subway, which is one of the busiest subway systems in the world. Line 9 is one of the busiest lines in the Beijing subway network. In order to show the effectiveness of the developed method in severely congested situations, we select the data corresponding to Line 9 during the morning peak hours from 7:00 to 9:00 for the simulation.

We use MATLAB (R2019b) for simulations on a computer with an Intel Xeon W-2223 CPU and 8 GB RAM. The main parameters associated with the simulation are listed in Table 4.5. In the developed model, we use the departure-departure headway as the cycle time, which is equal to the sum of the dwell time and the departure-arrival headway of the basic timetable. In the developed model, variables related to the number of passengers for all platforms are updated every cycle.

At each platform, the comparisons are conducted with three key values in the model, i.e., the accumulated number of passengers boarding the trains, the number of departing

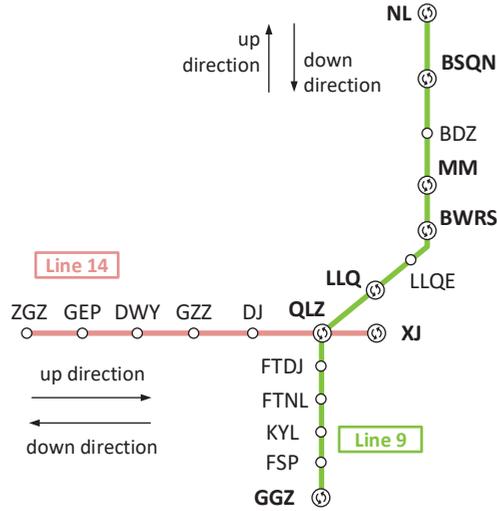


Figure 4.3: Real-life network of 2 lines from Beijing subway.

Table 4.5: Parameters for simulation of Line 9 and Line 14

Parameters	Line 9	Line 14
Dwell time $\tau_p(k_p)$	60 s	60 s
Departure-arrival headway	180 s	180 s
Cycle time $c_p(k_p)$	240 s	240 s
Number of trains	20	20
Train capacity	2400	2400
Average transfer time $\theta_{p,q}^{\text{trans}}$	60 s	60 s
Average transfer duration $\theta_{p,q}^{\text{duration}}$	60 s	60 s
Cruising speed	80 km/h	80 km/h

passengers, and the accumulated number of passengers that cannot board. The number of boarding passengers and departing passengers can reflect the utility of trains, which are related to operational costs, as the train operation company wants to transport as many passengers as possible with the available trains. The number of passengers who cannot board is related to passenger satisfaction because if passengers cannot board the current train upon their arrival, they have to wait for the next train.

We conduct simulations using both the accurate model and the developed model. For each line and each platform, we get the accumulated number of boarding passengers, the number of departing passengers, and the accumulated number of passengers that cannot board. The computation times needed to simulate the accurate model and the proposed model for the given period are 1.17 s and 0.24 s, respectively. The platform with the largest deviation between the proposed model and the accurate model is selected to illustrate the accuracy of the proposed model. The deviations are shown in Table 4.6.

Table 4.6: The largest deviation for each line

Passengers	Line 9	Line 14
Acc. # of boarding passengers	8.14%	0.58%
Number of departing passengers	11.59%	1.25%
Acc. # of pass. who cannot board	5.43%	0.1%

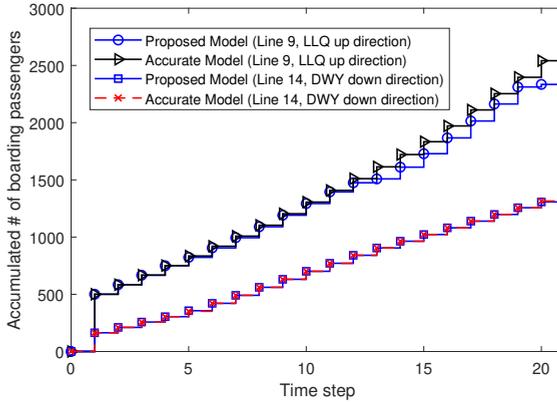


Figure 4.4: Accumulated number of boarding passengers at platforms.

For the accumulated number of boarding passengers, Line 9 Station LLQ (up direction platform) and Line 14 Station DWY (down direction platform) have the largest deviation, with an error of 8.14% and 0.58%, respectively. The simulation results of the platforms are also shown in Fig. 4.4.

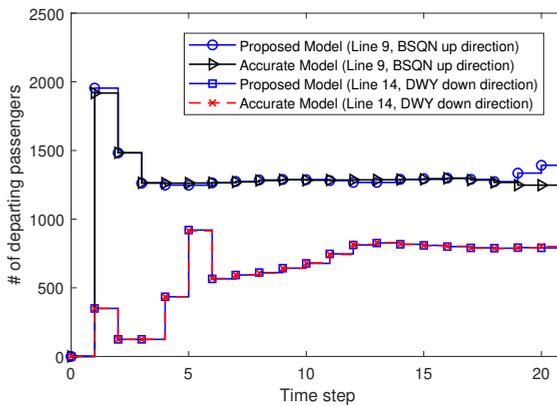


Figure 4.5: Number of departing passengers at each time step.

The largest deviation of the number of departing passengers for the lines occurs at Line 8 Station BSQN (up direction platform) and Line 14 Station DWY (down direction platform), with an error of 11.59% and 1.25%, respectively (see Fig. 4.5).

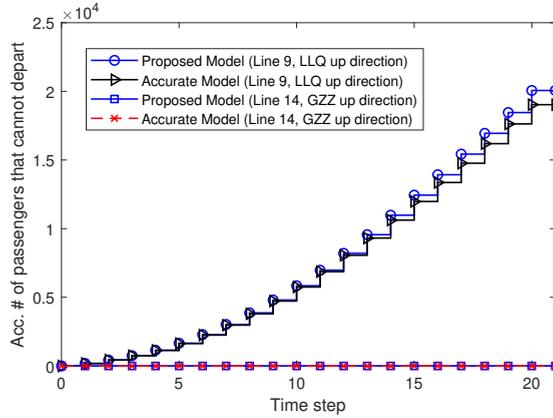


Figure 4.6: Accumulated number of passengers that cannot board at platforms.

For the accumulated number of passengers that cannot board, the largest deviation happens at Line 9 Station LLQ (up direction platform) and Line 14 Station GZZ (up direction platform), with an error of 5.43% and 0.1%, respectively, and we also provide the simulation results in Fig. 4.6.

According to above simulation results, we can conclude that the developed model can model the passenger flows with a maximal error of around 10% while the simulating time is reduced with a factor about 5, compared with the accurate model. Therefore, with an acceptable loss of accuracy, the proposed model can efficiently incorporate time-dependent passenger OD demands into the real-time timetable scheduling problem, which provides more possibilities to develop fast solution methods.

4.6.2. OPEN-LOOP OPTIMIZATION BASED ON THE PROPOSED MODEL

Now we perform numerical experiments for open-loop optimization to illustrate the solution quality and computation time of the approaches provided in Section 4.5, which can reflect the effectiveness and the real-time feasibility of the developed MPC controller. The model in [9; 136] is also used as the accurate model to simulate the “real-life network”, in order to compare and evaluate the performance of the approaches.

We use the same urban rail network as introduced in Section 4.6.1, and the parameters for train operation constraints are listed in Table 4.7, where r_{regular} indicates the running time from the corresponding platform to its successor platform of the basic timetable.

For the SQP approach, we use the `fmincon` function of the MATLAB Optimization Toolbox, and we adopt the `gurobi` solver implemented in MATLAB (R2019b) to solve the MILP problem. The experiments are performed on a computer with an Intel Xeon W-2223 CPU and 8GB RAM.

Table 4.7: Parameters for train operation constraints

Parameters	Line 9	Line 14
Minimum dwell time	30 s	30 s
Minimum headway	144 s	144 s
Maximum running time	$1.3 \cdot r_{\text{regular}}$	$1.3 \cdot r_{\text{regular}}$
Minimum running time	$0.8 \cdot r_{\text{regular}}$	$0.8 \cdot r_{\text{regular}}$

The basic timetable of the given urban rail network can be calculated by the parameters in Table I and the distance between each pair of consecutive platforms. The basic timetable represents the case without optimization. In the case study, we use the same data set with Section 4.6.1. We optimize the arrival and departure times of each platform for 5 time steps (i.e., $5 \cdot T_{\text{ctrl}}$). As the real-time feasibility is also important for the online implementation of an approach, the maximum solution time is set as 3600 s.

Simulation results are shown in Table 4.8, where the performance is the value of the objective function in (4.28). We find that all the approaches have better performance than basic timetable. In particular, the MILP approach has the best performance with the improvements for 22.66% compared with the basic timetable, while the improvement of SQP (with 1 starting point), SQP (with 10 starting points), and SMILP+SQP are 17.87%, 18.74%, and 18.30%, respectively.

In order to investigate the impact of regarding the variables related to the number of passengers as real-valued variables, we conduct an extra case study using the MILP formulation and by regarding passengers' number as integer variables, which is indicated as MILP-int in Table IV. We can find that the objective function value of MILP-int is very close to that of MILP. As the number of integer variables grows rapidly, the CPU time however increases dramatically, and the MILP-int approach cannot get its optimal solution within 3600 s, which indicates that MILP-int is not a suitable choice for real-time timetable scheduling.

Table 4.8: Comparison of performance and computation time corresponding to different problem formulations

Method	Objective function	CPU time (s)
Basic timetable	$1.3813 \cdot 10^4$	-
SQP (1 starting point)	$1.1344 \cdot 10^4$	264.3
SQP (10 starting points)	$1.1225 \cdot 10^4$	2845.7
SMILP (+SQP)	$1.1285 \cdot 10^4$	8.2
MILP-int	$1.0831 \cdot 10^4$	3600.0
MILP	$1.0683 \cdot 10^4$	6.4

The simulation results show that MILP performs best in terms of solution quality and solution time, which indicates that we can use the MILP-based MPC controller for real-time timetable scheduling. We can also find that the SQP approach is a bit time consuming compared with the MILP approach. SQP can easily fall into a suboptimal solution of the non-convex optimization problem, and the implementation of multi-start SQP can help to improve the performance of SQP. However, the computational burden

of multi-start SQP is much larger than single-start SQP, which would also influence the real-time feasibility of MPC. The SMILP approach can be used to find a starting point for the SQP approach so as to further improve the performance. In the case study, the solution obtained from SMILP approach is already a suboptimal solution of SQP; therefore, the application of SQP cannot further improve the performance of SMILP.

4.6.3. CLOSED-LOOP CONTROL FOR REAL-TIME TIMETABLE SCHEDULING

In Section 4.6.1 and Section 4.6.2, we have illustrated the effectiveness of the developed model and the MILP-based approach, respectively. In this section, numerical experiments are conducted from the control side based on the accurate model (i.e. the model of [9; 136]) and the newly developed model.

The urban rail network is shown in Fig. 4.3, and all settings related to the numerical experiment are identical with Section 4.6.2. The simulation is conducted for 15 time steps and the prediction horizon of MPC is 5 (i.e. $5 \cdot T_{\text{ctrl}}$). In the developed model, variables related to the number of passengers are updated every time step.

Table 4.9: Performance of MPC in real-time timetable scheduling

	Prediction model	Performance	CPU time (s)	
			t_{avg}	t_{max}
Basic timetable	Accurate model	$7.0692 \cdot 10^4$	-	-
SQP-based MPC	Accurate model	$6.1104 \cdot 10^4$	1799.4	2680.5
MILP-based MPC	Proposed model	$5.6763 \cdot 10^4$	4.0	9.1

It has been illustrated in Section 4.6.2 that MILP-based formulation performs best among the optimization approaches provided in Section 4.4; therefore, we only use the MILP-based MPC when employing the newly developed model as the prediction model. For the accurate model, we use SQP-based MPC as it is difficult to transform the MPC optimization problem of the accurate model into an MILP or SMILP problem. As real-time feasibility is important for MPC, in this section, we conduct numerical experiments for SQP-based MPC (with one starting point) to obtain an acceptable performance. For further improvement of SQP-based approach (with the cost of increasing computational burden), we refer to multi-start SQP approach which has been included in Section 4.6.2.

As Table 4.9 shows, both SQP-based MPC and MILP-based MPC perform much better than the basic timetable, with an improvement of 13.56% and 19.70% respectively in the performance, which indicates that SQP-based MPC and MILP-based MPC can be used to improve the performance of the basic timetable. Although we use a more accurate model for SQP-based MPC, MILP-based MPC performs slightly better than SQP-based MPC, as SQP can fall into suboptimal solution in the timetable scheduling problem.

We collect the computation time of the MPC optimization problem in each control step. The average and maximum CPU time of SQP-based MPC are 1799.4 s and 2680.5 s, respectively, which indicates SQP-based MPC may not be a suitable choice for real-time timetable scheduling. MILP-based MPC is time efficiency, with average and maximum CPU time as 4.0 s and 9.1 s, respectively.

In order to graphically show the results, we depict a part of the timetable from Line 9 in the considered time window. The basic timetable, the timetable generated by SQP-

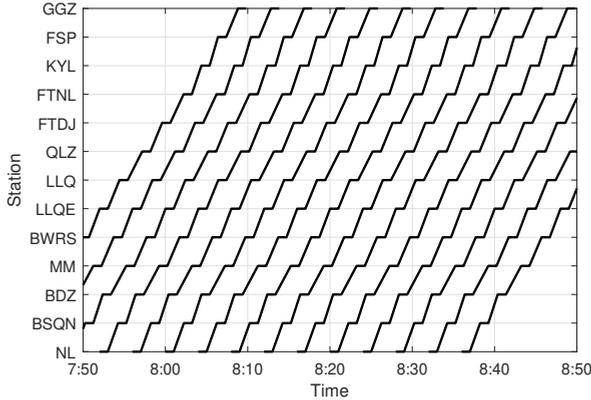


Figure 4.7: Basic timetable.

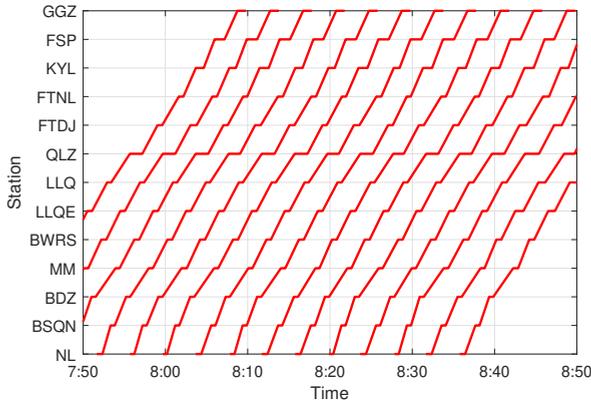


Figure 4.8: Timetable generated by SQP-based MPC.

based MPC, and the timetable generated by MILP-based MPC are shown in Fig. 4.7, Fig. 4.8, and Fig. 4.9, respectively. Both SQP-based MPC and MILP-based MPC can adjust the arrival and departure times in real time so that the performance of the corresponding timetable is improved compared with that of the basic timetable. The timetable of SQP-based MPC is not the same as that of MILP-based MPC, because we only take one starting point (considering the real-time feasibility of the approach), which would typically result in a suboptimal solution. In order to show the impact on the passengers of different timetables more clearly, the variables related to the number of passengers are analyzed in the following.

The total number of departing passengers for all lines and all platforms is depicted in Fig. 4.10. The timetable obtained from the MILP-based MPC approach results in more

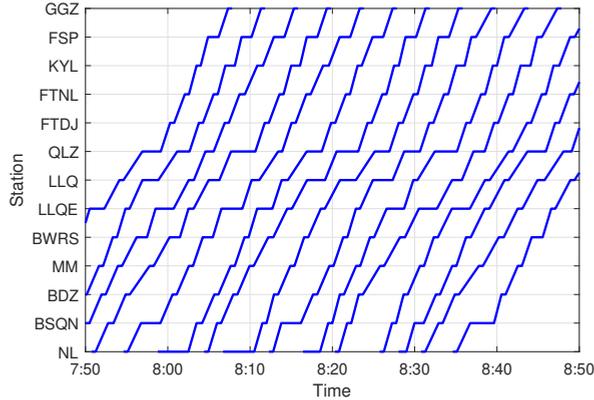


Figure 4.9: Timetable generated by MILP-based MPC.

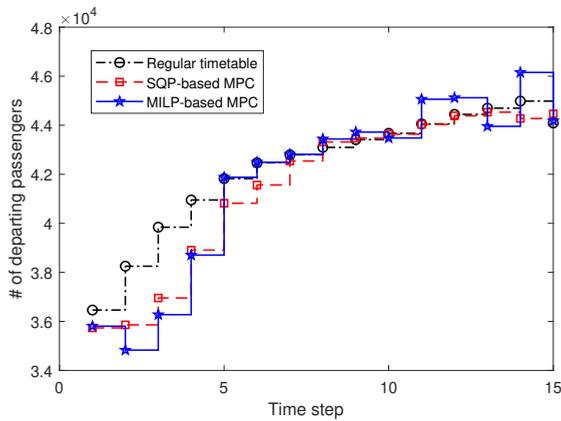


Figure 4.10: Total number of departing passengers at each time step.

boarding and departing passengers, which means the resulting timetable can make better use of the available trains.

The total number of waiting passengers before the train departs and the total number of passenger who cannot board the train, for all lines and all platforms, is depicted in Fig. 4.11 and Fig. 4.12, respectively. We can find that the timetable obtained from the MILP-based MPC controller results in less number of waiting passengers and less number of passengers who cannot depart, i.e., more passengers can board their target trains, which indicates that MILP-based MPC can help to improve passenger satisfaction.

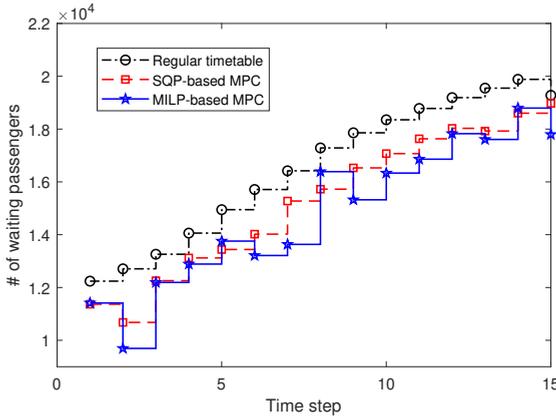


Figure 4.11: Total number of waiting passengers at each time step.

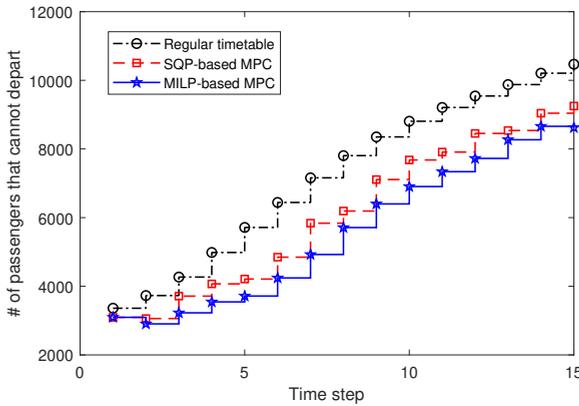


Figure 4.12: Total number of passengers that cannot depart at each time step.

4.7. CONCLUSIONS

In this chapter, we have proposed a novel passenger flow model for real-time timetable scheduling of urban rail networks. By introducing the cycle time, the time-dependent passenger origin-destination demands can be modeled very efficiently, with a loss of accuracy at around 10% compared with an accurate model for a simulation including part of Beijing urban rail network. Furthermore, a model predictive control framework was proposed for real-time timetable scheduling. In order to increase the real-time feasibility of MPC, the optimization problem in MPC has been transformed into a mixed-integer linear programming problem, which can be solved very fast by existing MILP solvers. Simulation results indicate that the MILP approach can greatly reduce the online computational burden of the MPC controller with the developed model. The developed

model and MILP-based MPC controller can be used in real-time timetable scheduling for real-life passenger-oriented urban rail networks.

In our future work, we will investigate the possibility of using MILP-based MPC combined with more accurate models by designing efficient methods to transform or approximate the integral of the passenger arrival rates into mixed-integer linear inequalities. We will design distributed control approaches for large-scale networks, where the developed MILP-based MPC controller will be used as the local controller. Furthermore, flexible coupling of trains will be considered, so that the capacity of trains at each cycle can be adjusted based on passenger demands. The influence of uncertain passenger demands and the order of trains will also be a topic for future research.

5

LEARNING-BASED MPC FOR TRAIN RESCHEDULING WITH FLEXIBLE TRAIN COMPOSITION

This chapter focuses on passenger-oriented real-time train scheduling, considering flexible train composition and rolling stock circulation, by integrating learning-based and optimization-based approaches. A learning-based model predictive control (MPC) approach is developed for real-time train rescheduling including train compositions, and taking into account rolling stock circulation to address time-varying passenger demands. In the proposed approach, first the values of the integer variables are obtained by pre-trained long short-term memory (LSTM) networks; next, they are fixed and the values of continuous variables are determined via nonlinear constrained optimization. The learning-based MPC approach enables us to jointly consider efficiency and constraint satisfaction by combining learning-based and optimization-based approaches. In order to reduce the number of integer variables, four presolve techniques are developed to prune a subset of integer decision variables. Numerical simulations based on real-life data from the Beijing urban rail transit system are conducted to illustrate the effectiveness of the developed learning-based MPC approach.

This chapter is based on [80], for which X. Liu focused on the research framework, model, integration of application and learning approaches, joint programming, and paper writing.

5.1. INTRODUCTION

Urban rail transit has become indispensable in metropolitan areas due to its reliability, high capacity, and eco-friendly characteristics. Urban rail transit systems prioritize safe and efficient train operations while providing high-quality service to passengers. Effective real-time train scheduling is essential for enhancing passenger satisfaction and minimizing operational costs within infrastructure limitations. However, the rapid expansion of urban rail transit systems and the increasing passenger demands pose significant challenges to real-time scheduling. Advanced scheduling models and control strategies are required to develop efficient timetables and to improve the overall performance of urban rail transit systems.

5.1.1. PASSENGER-ORIENTED TRAIN SCHEDULING

In urban rail transit systems, passenger demands vary throughout the day, necessitating adjustments of the train schedules to accommodate these demand variations while considering operational costs. One direction addresses time-varying passenger demands by optimizing the departure and arrival times of trains at each station, while taking into account several attributes of train operations and infrastructure restrictions, e.g., train stopping plans [14; 103], rolling stock circulations [133], and train speed levels [51; 137]. Qi et al. [103] optimized train stopping plans and timetables of a high-speed railway line considering time-varying passenger demands by formulating a mixed-integer linear programming (MILP) problem. The aim of [103] is to find a solution that considers passenger preferences for departure times while ensuring trains operate within capacity limits. Considering the passenger load of trains, Wu et al. [141] minimized the passenger waiting time and the energy consumption by developing a heuristic algorithm to solve the resulting nonlinear integer programming problem. Wang et al. [137] formulated a mixed-integer nonlinear programming (MINLP) problem to minimize train energy consumption and passenger waiting times by optimizing train speed levels and headway deviations. However, the above studies optimize train schedules within the fixed transport capacity of a rail line, such as fixed train compositions or fixed train departure frequencies. As transport capacity directly impacts passenger flows, further research is required to improve passenger satisfaction by including transport capacity as a decision variable.

Another direction for train scheduling problems addresses time-varying passenger demands by optimizing transport capacity explicitly. Several studies focus on optimizing transport capacity by adjusting train departure frequencies, with higher frequencies during peak hours and lower frequencies during off-peak hours [19; 77; 79]. However, passengers typically expect regular departures at each platform to plan their travels conveniently and have a predictable waiting time in case they miss a connecting train [76]. Therefore, instead of changing the departure frequency, which would significantly impact the timetable, in recent years, many researchers have focused on optimizing the train composition [153; 156]. Pan et al. [99] developed a column-generation-based approach to optimize the timetable, train composition, and rolling stock circulation plan of an urban rail transit line. Their paper concludes that flexible train composition can provide additional adaptability to better match time-varying passenger demands. Wang et al. [129] investigated flexible train composition and rolling stock circulation of an urban rail transit line, and solved the resulting MILP problem by developing an approximation

approach and a two-stage meta-heuristic algorithm. Yang et al. [147] investigated the train scheduling problem with flexible train composition for an urban rail transit line, and they applied an adaptive large neighborhood search algorithm to solve the resulting integer programming problem. As the inclusion of train composition optimization and rolling stock circulation planning introduces additional integer variables, the above studies indicate that the online computational complexity is the main challenge of incorporating flexible train composition into the real-time train scheduling problem.

5.1.2. MPC FOR REAL-TIME TRAIN SCHEDULING

Model predictive control (MPC) has been widely adopted in various applications for its ability to handle multivariable constrained control problems [46; 92; 104]. The train scheduling problem is a typical contained control problem, and many studies have applied MPC for real-time train scheduling. De Schutter et al. [31] first applied MPC in the train scheduling problem to minimize train delays by adjusting transfer connections. Caimi et al. [15] developed an MPC algorithm to optimize timetables, transfer connections, and train assignment plans in complex station areas. Cavone et al. [22] proposed an MPC approach for train rescheduling during disruptions and delays, where in each step the resulting MILP problem is solved by combining bi-level heuristics and distributed optimization. The above studies only handle operator-related factors in railway systems, leaving an open gap in including passenger demands in real-time train scheduling problems to improve the service quality of urban rail transit systems.

In recent years, several studies have focused on MPC for real-time passenger-oriented train scheduling. Assis and Milani [5] applied MPC to compute the train timetable of a metro line considering train headway and passenger load, where a linear programming problem is solved at each step. Based on a state-space model, Li et al. [68] developed a robust MPC approach to minimize the upper bound of the timetable deviation from the nominal timetable under uncertain disturbances. By solving linear matrix inequalities, they constructed a Lyapunov function to ensure the attenuation of the timetable deviation. An event-triggered MPC approach is further developed in [130] to reduce the computational burden of updating control variables in each step. However, these studies do not explicitly consider train capacity limitations and time-varying passenger demands, and the results are based on the assumption that the maximum number of passengers does not exceed the maximum train capacity. Liu et al. [76] explicitly incorporated time-varying passenger demands and train capacity limitations into the real-time train scheduling problem. In [76], the time-varying passenger demand is approximated as a piecewise constant function by dividing the planning time window into several time intervals of equal length, and an MILP-based MPC approach is adopted for real-time train scheduling. The main challenge of applying MPC in real time is the online computational burden. Including additional attributes, such as train capacity, train composition, and rolling stock circulation, will further increase the computational burden. Therefore, further research is required to develop efficient MPC approaches for real-time passenger-oriented train scheduling.

5.1.3. LEARNING-BASED TRAIN SCHEDULING

Learning-based approaches are effective in reducing the online computational burden and have been developed for dynamic control problems in different fields, such as road transportation systems [120], smart building systems [127], and power systems [39]. Learning-based approaches, including supervised learning (SL) and reinforcement learning (RL), have also been applied in train scheduling problems in recent years [124]. SL trains models on labeled data to make accurate predictions or classifications, while RL trains an agent to make decisions through trial-and-error using rewards and penalties. Kuppusamy et al. [60] applied a deep learning approach to an energy-efficient timetable rescheduling problem, where a long short-term memory (LSTM) network is trained to select the optimal operation mode. Šemrov et al. [110] applied Q-learning in the train rescheduling problem to reduce delays caused by disturbances and disruptions, and they illustrated their method on a single-lane track with three trains. Yin et al. [151] proposed an approximate dynamic programming approach to address train rescheduling problems, aiming to reduce passenger delays, total travel time, and train energy consumption. In [151], the states include disturbance information, train arrival times, the number of boarding passengers, and the number of waiting passengers, while the actions include rescheduling the dwell times and running times. Khadilkar [57] applied RL to determine track allocations and timetables of bidirectional railway lines to minimize the priority-weighted delay. Ying et al. [153] developed a proximal policy optimization approach for the train scheduling problem in an urban rail transit line considering flexible train composition. In [153], the control policy and the value function are parameterized by artificial neural networks, and scheduling constraints are handled by a devised mask scheme. Simulation results show that this approach reduces the computational burden and improves solution quality compared to the genetic algorithm and differential evolution. More studies of learning-based approaches in railway systems can be found in the recent review paper [124].

In the above studies, scalability and constraint satisfaction are two main challenges in developing learning-based train scheduling approaches. The train scheduling problem is typically formulated as an MILP or MINLP problem, and the computational complexity increases rapidly as the number of integer variables increases. In recent years, some research has combined learning-based and optimization-based approaches for MILP or MINLP problems by using learning-based approaches to obtain the integer variables. Having the integer variables fixed, the continuous variables are then obtained by solving a linear or nonlinear programming problem. In this context, the aim is to combine the advantages of both learning-based and optimization-based approaches, i.e., the online computational efficiency of learning-based approaches and the constraint satisfaction of optimization-based approaches [21; 108]. Promising results of learning-based approaches in railway systems and the novel learning-based approaches in mixed integer programming problems have inspired us to develop new learning-based frameworks for real-time train scheduling.

5.1.4. CONTRIBUTIONS AND STRUCTURE OF THE CHAPTER

This chapter addresses the real-time, passenger-oriented train rescheduling problem taking into account time-varying passenger demands, flexible train composition, and

rolling stock circulation. The main contributions of the chapter are listed as follows:

1. A passenger-oriented train scheduling model [76] is extended to include time-varying sectional passenger demands, flexible train composition, and rolling stock circulation. The time-varying passenger demands can be approximated as a piecewise constant function by dividing the prediction horizon into several time intervals.
2. Four presolve techniques are developed to streamline optimization processes by pruning a subset of integer decision variables. After implementing the presolve techniques, long short-term memory (LSTM) networks are applied to obtain the remaining integer variables with reduced dimensions.
3. A learning-based MPC approach is developed for real-time train rescheduling. To improve the online computational efficiency of MPC, the learning-based approach is applied to obtain integer variables while the detailed timetable is obtained by solving a constrained optimization problem with the fixed integer variables.

The remaining part of the chapter is organized as follows: Section 5.2 provides the problem description and general explanations. Section 5.3 introduces the passenger-oriented train scheduling model. In Section 5.4, the problem formulation and the MINLP-based MPC approach are presented. In Section 5.5, we propose a learning-based MPC approach for real-time train rescheduling. Section 5.6 provides an illustrative case study. Section 5.7 concludes the chapter.

5.2. PROBLEM DESCRIPTION AND EXPLANATIONS

In urban rail transit systems, passengers typically expect regular departures at every platform to plan their trips conveniently and to avoid extended waiting times in case they miss a connecting train [76]. In this context, we consider the regular departure of trains and optimize the transport capacity by adjusting train composition.

In urban rail transit systems, a line is defined as the route of trains with the same origin, intermediate, and destination stations. A train service is defined as a train departure from its origin station, visiting every station in the line, and finally returning to the depot. As illustrated in Fig. 5.1, a train service consists of one or several train units, and the composition can be changed at the station connected to a depot by adding or removing train units.

In this chapter, we jointly optimize train compositions and timetables considering time-varying passenger demands and rolling stock circulation. We aim to minimize the total waiting time of passengers and the train energy consumption, and the control actions are the train composition, train orders, and train timetables. Flexible train composition and rolling stock circulation relate to the set of integer variables, while train timetables relate to the set of continuous variables, such as departure and arrival times. The train operates under several constraints, including train capacity constraints, train availability in a depot, and headway constraints. By applying MPC, we solve the resulting mixed-integer programming problem in a moving horizon manner for real-time train scheduling. To improve the online computational efficiency of MPC, we use the

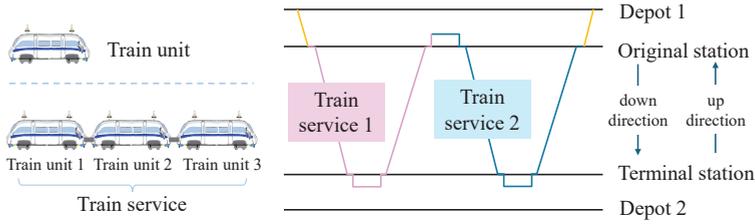


Figure 5.1: Illustration of definitions used in this chapter.

learning-based approach to obtain integer variables, i.e., train compositions and train orders; then, we optimize the detailed timetable with the fixed integer variables by solving a constrained optimization problem.

5.3. MATHEMATICAL FORMULATION FOR PASSENGER-ORIENTED TRAIN SCHEDULING

In this section, we develop a passenger-oriented train scheduling model for urban rail transit systems. The notations for the model formulation are provided in Section 5.3.1. The train operation constraints of the model are introduced in Section 5.3.2. In Section 5.3.3, the rolling stock circulation constraints related to the model are introduced. In Section 5.3.4, passenger flow constraints of the model are presented.

5.3.1. NOTATIONS

5.3.2. TRAIN OPERATION CONSTRAINTS

In urban rail transit systems, each line typically comprises two directions, i.e., the up direction and the down direction, as shown for a line of P stations in Fig. 5.2. For each line, a train service can be defined as a train running from the starting platform to the terminal platform, e.g., from Platform 1 to Platform $2P$ in Fig. 5.2. Trains generally oper-

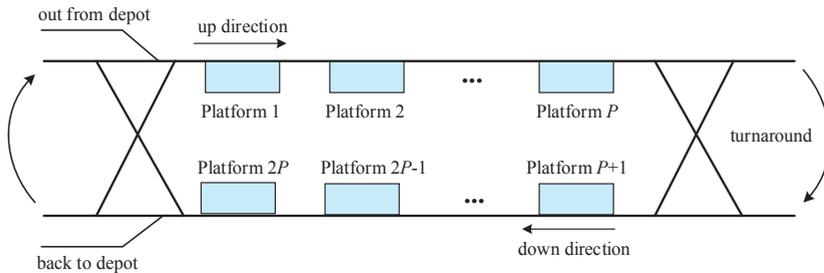


Figure 5.2: Layout of a bidirectional urban rail transit line.

ate following a predetermined timetable, and the predetermined departure time of train service k_p at platform p is represented by $d_p^{\text{pre}}(k_p)$.

Notations	Definition
p	Index of platforms, $p \in \mathcal{P}$; \mathcal{P} is the set of platforms
k_p	Index of train services at platform p , $k_p \in \mathcal{K}_p$; \mathcal{K}_p is the set of train services departing from platform p
z	Index of depots; $\text{pla}(z)$ defines the set of platforms directly connected with depot z
$p^{\text{pla}}(p)$	Predecessor platform of platform p
$s^{\text{pla}}(p)$	Successor platform of platform p
$d_p^{\text{pre}}(k_p)$	Predetermined departure time of train service k_p at platform p
h_p^{min}	Minimum departure-arrival headway at platform p
r_p^{min}	Minimum running time of trains from platform p to its succeeding platform
r_p^{max}	Maximum running time of trains from platform p to its succeeding platform
C_{max}	Maximum capacity of a train unit
ℓ^{min}	Minimum number of train units allowed to be included in any train service
ℓ^{max}	Maximum number of train units allowed to be included in any train service
σ_p	Parameter indicating whether the train composition can be adjusted at platform p , i.e., if the train composition can be adjusted at platform p , then, $\sigma_p = 1$, otherwise, $\sigma_p = 0$
τ_p^{min}	Minimum dwell time of a train service at platform p
t_p^{cons}	Time required for changing the train composition at platform p
t_p^{roll}	Time for trains from platform p to other platforms corresponding to the same depot
$\text{dep}(z)$	Set of platforms directly connected with depot z
$\text{pla}(p)$	Set of platforms belonging to the same station as platform p
$\rho_p(k_p)$	Passenger demands from platform p to its successor platform during $d_{k-1,p}^{\text{pre}}$ to $d_{k,p}^{\text{pre}}$
N_z^{train}	Total number of train units available at depot z
$\chi_{k_q,q,k_p,p}$	Binary parameter; if train k_q at platform q has transfer connection with train k_p at platform p , $\chi_{k_q,q,k_p,p} = 1$; otherwise, $\chi_{k_q,q,k_p,p} = 0$.
t_q^{trans}	Average transfer time from platform q to the corresponding platforms at the same station
E_p^{energy}	Average energy consumption for a train unit running from platform p to its successor platform
E_p^{add}	Additional cost of changing train composition at platform p

Table 5.1: Indices and Input Parameters

Notations	Definition
$d_p(k_p)$	Departure time of train service k_p at platform p
$a_p(k_p)$	Arrival time of train service k_p at platform p
$\ell_p(k_p)$	Number of train units included in train service k_p at platform p , $\ell_p(k_p) \in \mathbb{Z}_+$

Table 5.2: Decision variables

In practice, premature departure is usually not permitted; thus, the predetermined departure time defines a lower bound of the actual departure time. In general, passengers expect regular departures at every platform so that they can conveniently plan their travels and avoid extended waiting times for the next train in case they miss a connecting train. Therefore, in this chapter, we do not change the departure frequency of trains when adjusting the timetable and train composition. Hence, the actual departure time is constrained by

$$d_p^{\text{pre}}(k_p) \leq d_p(k_p) < d_p^{\text{pre}}(k_p + 1), \quad (5.1)$$

where $d_p(k_p)$ is the actual departure time of train service k_p at platform p determined by

$$d_p(k_p) = a_p(k_p) + \tau_p(k_p), \quad (5.2)$$

Notations	Definition
$\tau_p(k_p)$	Dwell time of train service k_p at platform p
$h_p(k_p)$	Departure-arrival headway between train service k_p and train service $k_p + 1$ at platform p
$r_p(k_p)$	Running time of train service i from platform p to its successor platform
$y_p(k_p)$	Number of train units coming to/from the depot for train service k_p , $y_p(k_p) \in \mathbb{Z}$
$\tau_p^{\text{add}}(k_p)$	Additional time required for changing the composition of train service k_p at platform p
$\eta_{k_p,p}$	Binary variable; if the composition of train service k_p is changed at platform p , $\eta_{k_p,p} = 1$; otherwise, $\eta_{k_p,p} = 0$
$\xi_{k_p,k_{p'},p,p'}$	Binary variable; if the train units from train service $k_{p'}$ at platform p' can be used for train service k_p at platform p , $\xi_{k_p,k_{p'},p,p'} = 1$; otherwise, $\xi_{k_p,k_{p'},p,p'} = 0$
$n_p(k_p)$	Number of passengers waiting at platform p at time $d_p^{\text{pre}}(k_p)$
$n_p^{\text{trans}}(k_p)$	Number of transfer passengers arriving at platform p for train k_p
$n_p^{\text{depart}}(k_p)$	$n_p^{\text{depart}}(k_p)$ denotes the number of passengers departing from platform p with train service k_p
$n_p^{\text{arrive}}(k_p)$	Number of passengers arriving at platform p with train k_p from the predecessor platform $p^{\text{pla}}(p)$
$n_p^{\text{before}}(k_p)$	Number of passengers waiting at platform p immediately before the departure of train service k_p
$C_p(k_p)$	Total capacity of train service k_p at platform p
$n_p^{\text{after}}(k_p)$	Number of passengers waiting at platform p immediately after train service k_p departs from platform p

Table 5.3: Output variables

where $a_p(k_p)$ and $\tau_p(k_p)$ are the arrival time and dwell time of train service k_p at platform p .

For the safe operation of trains, the headway constraint should be satisfied:

$$a_p(k_p + 1) = d_p(k_p) + h_p(k_p), \quad (5.3)$$

$$h_p(k_p) \geq h_p^{\min}, \quad (5.4)$$

where $h_p(k_p)$ is headway of train service k_p at platform p , and h_p^{\min} denotes the minimum headway.

The arrival time of train service k_p at the successor platform of platform p should also satisfy

$$a_{s^{\text{pla}}(p)}(k_p) = d_p(k_p) + r_p(k_p), \quad (5.5)$$

$$r_p^{\min} \leq r_p(k_p) \leq r_p^{\max}, \quad (5.6)$$

where $r_p(k_p)$ is the running time of train service k_p from platform p to platform $s^{\text{pla}}(p)$, and r_p^{\min} and r_p^{\max} are the minimum and maximum running time from platform p to platform $s^{\text{pla}}(p)$.

5.3.3. ROLLING STOCK CIRCULATION CONSTRAINTS

At the terminal station, a turnaround action is required for the continuation of the train service. The turnaround constraints can be formulated as:

$$a_{\text{spla}(p)}(k_{\text{spla}(p)}) = d_p(k_p) + r_p^{\text{turn}}(k_p), \quad (5.7)$$

$$r_p^{\text{turn},\min} \leq r_p^{\text{turn}}(k_p) \leq r_p^{\text{turn},\max}, \quad (5.8)$$

where $r_p^{\text{turn}}(k_p)$ represents the turnaround time of train service k at platform p , and $r_{p,\min}^{\text{turn}}$ and $r_{p,\max}^{\text{turn}}$ denote the minimum and maximum turnaround times at platform p .

An urban rail transit line typically has a limited number of train units that either operate on the line or are stored in the depot. The train composition can be adjusted at the platform that is linked with the depot, and the number of train units $\ell_p(k_p) \in \mathbb{Z}_+$ for train service k_p at platform p is determined by

$$\ell_p(k_p) = \ell_{\text{ppla}(p)}(k_p) + \sigma_p y_p(k_p), \quad (5.9)$$

$$\ell_{\min} \leq \ell_p(k_p) \leq \ell_{\max}, \quad (5.10)$$

where σ_p is a parameter related to the network layout indicating whether the train composition can be adjusted at platform p , i.e., if the train composition can be adjusted at platform p , then, $\sigma_p = 1$, otherwise, $\sigma_p = 0$. Moreover, $y_p(k_p) \in \mathbb{Z}$ represents the number of train units coming to/from the depot for train service k ; specifically, if $y_p(k_p) > 0$, then, $y_p(k_p)$ extra train units will come from the depot to be added to train service k_p ; if $y_p(k_p) < 0$, train service k_p will be decomposed and $|y_p(k_p)|$ train units will return to the depot; if $y_p(k_p) = 0$ the composition of train service k_p will not be changed at platform p . Furthermore, ℓ_{\min} and ℓ_{\max} represent the minimum and the maximum number of train units allowed to be included in any train service.

Remark 5.1: If the depot is linked with the terminal platform (e.g., Platform 1 in Fig. 5.2) and train service k is performed by the turnaround trains of the previous train service, then (5.9) becomes $\ell_1(k_1) = \ell_{2p}(k_{2p}-1) + \sigma_1 y_1(k_1)$, i.e., we set $\ell_1(k_1) = \ell_{2p}(k_{2p}-1)$.

To capture the changes of train composition at platform p , we introduce a binary variable $\eta_{k_p,p}$ as

$$\eta_{k_p,p} = \begin{cases} 1, & \text{if } |y_p(k_p)| > 0; \\ 0, & \text{otherwise,} \end{cases} \quad (5.11)$$

where $\eta_{k_p,p} = 1$ indicates the composition of train service k_p is changed at platform p ; otherwise, $\eta_{k_p,p} = 0$.

In general, additional dwell time is required when changing the train composition; thus, the dwell time $\tau_p(k_p)$ of train service k_p at platform p should satisfy:

$$\tau_p(k_p) \geq \tau_p^{\min} + \sigma_p \tau_p^{\text{add}}(k_p), \quad (5.12)$$

where τ_p^{\min} is the minimum dwell time at platform p , and $\tau_p^{\text{add}}(k_p)$ represents the additional time required for changing the composition of train service k_p at platform p , which can be determined by

$$\tau_p^{\text{add}}(k_p) = \eta_{k_p,p} \cdot t_p^{\text{cons}}, \quad (5.13)$$

where t_p^{cons} is a constant representing the time required for changing the train composition at platform p .

A depot typically connects to at least one platform, and the availability of train units in a depot is influenced by the departure order of train services at the corresponding platforms, i.e., if a newly arriving train service requires changing its composition, the total number of train units in the corresponding depot will change. To represent the relation of departure time of train services corresponding to the same depot, we define a binary variable $\xi_{k_p, k_{p'}, p, p'}$ as

$$\xi_{k_p, k_{p'}, p, p'} = \begin{cases} 1, & \text{if } d_p(k_p) \geq d_{p'}(k_{p'}) + t_{p'}^{\text{roll}}; \\ 0, & \text{otherwise,} \end{cases} \quad (5.14)$$

where $d_p(k_p)$ is the departure time of train service k_p at platform p , $d_{p'}(k_{p'})$ is the departure time of train service $k_{p'}$ at platform p' , and $t_{p'}^{\text{roll}}$ is the time for trains from platform p' to other platforms corresponding to the same depot. In (5.14), $\xi_{k_p, k_{p'}, p, p'} = 1$ indicates train units in train service $k_{p'}$ from platform p' is available for train service k_p at platform p ; otherwise, $\xi_{k_p, k_{p'}, p, p'} = 0$.

Then, the rolling stock circulation constraint is

$$\sum_{k_p \in \mathcal{S}_p} y_p(k_p) + \sum_{p' \in \text{dep}(z) \setminus \{p\}} \sum_{k_{p'} \in \mathcal{S}_{p'}} \xi_{k_p, k_{p'}, p, p'} y_{p'}(k_{p'}) \leq N_z^{\text{train}}, \quad (5.15)$$

where z is the index of the depot; $\text{dep}(z)$ defines the set of platforms directly connected with depot z ; $\mathcal{S}_{p'}$ defines the set of train services departing from platform p' ; and N_z^{train} represents the total number of train units available at depot z . Eq. (5.15) indicates that the total number of train units departing from depot z before time $d_p(k_p)$ should be less than or equal to the total number of available train units in the depot.

5.3.4. PASSENGER FLOW CONSTRAINTS

A predetermined timetable is generally designed based on the time-varying passenger demand to guide the daily operation of trains. At each platform, the predetermined timetable naturally divides the planning time window into several time intervals, with the predetermined train departure times at the platforms as the partition points. In this chapter, we approximate the passenger arrival rates during each time interval as constants. The resulting piecewise approximation is shown in Fig. 5.3 where $d_p^{\text{pre}}(k_p)$ is the predetermined departure time of train service k_p ; and $\rho_p(k_p)$ denotes passenger demands during $d_p^{\text{pre}}(k_p - 1)$ to $d_p^{\text{pre}}(k_p)$ for passengers aiming to leave platform p with train service k_p .

The number of passengers waiting at a platform immediately after the predetermined departure time of train service $k_p + 1$ at platform p can be calculated by

$$n_p(k_p + 1) = n_p(k_p) + \rho_p(k_p + 1) (d_p^{\text{pre}}(k_p + 1) - d_p^{\text{pre}}(k_p)) + n_p^{\text{trans}}(k_p) - n_p^{\text{depart}}(k_p), \quad (5.16)$$

where $n_p(k_p)$ represents the number of passengers waiting at platform p at time $d_p^{\text{pre}}(k_p)$; $\rho_p(k_p + 1) (d_p^{\text{pre}}(k_p + 1) - d_p^{\text{pre}}(k_p))$ calculates the number of passengers arriving at platform p between $d_p^{\text{pre}}(k_p)$ and $d_p^{\text{pre}}(k_p + 1)$; $n_p^{\text{trans}}(k_p)$ denotes the number of transfer

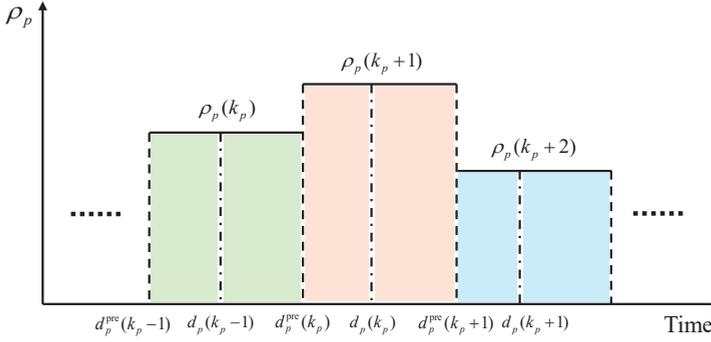


Figure 5.3: Illustration of piecewise approximation of passenger demands for platform p .

passengers arriving at platform p for train k_p , and $n_p^{\text{depart}}(k_p)$ denotes the number of passengers departing from platform p with train service k_p .

Since the departure time of each train service is adjusted according to (5.1), there exists one departure in each time interval. Having Fig. 5.3, the number of passengers $n_p^{\text{before}}(k_p)$ waiting at platform p immediately before the departure of train service k_p can be calculated by

$$n_p^{\text{before}}(k_p) = n_p(k_p) + \rho_p(k_p + 1) (d_p(k_p) - d_p^{\text{pre}}(k_p)) + n_p^{\text{trans}}(k_p). \quad (5.17)$$

The number of passengers $n_p^{\text{depart}}(k_p)$ departing from platform p with train service k_p should satisfy

$$n_p^{\text{depart}}(k_p) \leq C_p(k_p), \quad (5.18a)$$

$$n_p^{\text{depart}}(k_p) \leq n_p^{\text{before}}(k_p), \quad (5.18b)$$

where $C_p(k_p)$ is the total capacity of train service k_p at platform p .

The total capacity of train service k_p at platform p is computed by

$$C_p(k_p) = \ell_p(k_p) C_{\max}, \quad (5.19)$$

where $\ell_p(k_p)$ is the number of train units composing train service k_p at platform p .

The number of passengers $n_p^{\text{arrive}}(k_p)$ arriving at platform p with train k_p from the predecessor platform $p^{\text{pla}}(p)$ can be calculated by

$$n_p^{\text{arrive}}(k_p) = n_{p^{\text{pla}}(p)}^{\text{depart}}(k_p). \quad (5.20)$$

Then, the number of passengers $n_p^{\text{trans}}(k_p)$ transferring to platform p for train k_p is computed by

$$n_p^{\text{trans}}(k_p) = \sum_{q \in \text{pla}(p)} \sum_{k_q \in \mathcal{I}_q} \chi_{k_q, q, k_p, p} n_p^{\text{arrive}}(k_p), \quad (5.21)$$

where $\text{pla}(p)$ defines the set of platforms belonging to the same station as platform p , and $\chi_{k_q, q, k_p, p}$ is the binary parameter denoting transfer connection between train k_q at platform q and train k_p at platform p , which is defined as

$$\chi_{k_q, q, k_p, p} = \begin{cases} 1, & \text{if } d_p^{\text{pre}}(k_p - 1) < d_q^{\text{pre}}(k_q) + t_q^{\text{trans}} \leq d_p^{\text{pre}}(k_p); \\ 0, & \text{otherwise,} \end{cases} \quad (5.22)$$

where t_q^{trans} represents the average transfer time from platform q to the corresponding platforms at the same station.

After train service k_p departs from platform p , the number of passengers waiting at the platform can be calculated by

$$n_p^{\text{after}}(k_p) = n_p^{\text{before}}(k_p) - n_p^{\text{depart}}(k_p), \quad (5.23)$$

where $n_p^{\text{after}}(k_p)$ represents the number of passengers waiting at platform p immediately after train service k_p departs from platform p .

5

5.4. MPC FOR REAL-TIME TRAIN RESCHEDULING

5.4.1. PROBLEM FORMULATION

Based on the developed model, we can formulate the problem to minimize passenger delays and operational costs. The passenger delays corresponding to train service k_p at platform p can be formulated as

$$J_p^{\text{pass}}(k_p) = n_p(k_p) (d_p(k_p) - d_p^{\text{pre}}(k_p)) + n_p^{\text{after}}(k_p) (d_p^{\text{pre}}(k_p + 1) - d_p(k_p)). \quad (5.24)$$

As passengers expect to depart at the predetermined departure time $d_p^{\text{pre}}(k_p)$, the term $n_p(k_p) (d_p(k_p) - d_p^{\text{pre}}(k_p))$ in (5.24) represents the delay for passengers departing from platform p with train service k_p ; the term $n_p^{\text{after}}(k_p) (d_p^{\text{pre}}(k_p + 1) - d_p(k_p))$ denotes the expected delay for passengers that could not board train service k_p , hence they have to wait for the next train at the platform.

Assigning more train units to a train service can increase the capacity for transporting passengers while leading to higher energy consumption. Furthermore, changing the train composition may require additional workload from operators and thus lead to additional costs. The operational costs of train service k_p running from platform p to its successor platform can be expressed as

$$J_p^{\text{cost}}(k_p) = \ell_p(k_p) E_p^{\text{energy}} + \eta_{k_p, p} E_p^{\text{add}}, \quad (5.25)$$

where E_p^{energy} represents the average energy consumption for a train unit running from platform p to its successor platform, and E_p^{add} denotes the additional cost for changing the train composition of a train service at platform p .

In urban rail transit systems, a train service typically departs from a depot, visiting each platform along a line before returning to the depot. As we want to ensure the regular departure of trains at each platform, we define the time interval between two consecutive predetermined departure times as the control time step, and the length of the

control time step, denoted as T_{ctrl} for all platforms along a line is identical, and the control time step is indexed as κ .

Therefore, the optimization problem for the train rescheduling problem is

$$\begin{aligned} \min_{\mathbf{g}(\kappa_0)} J(\kappa_0) &:= \sum_{p \in \mathcal{P}} \sum_{k_p \in \mathcal{N}_p(\kappa_0)} \left(w_1 J_p^{\text{pass}}(k_p) + w_2 J_p^{\text{cost}}(k_p) \right), \\ \text{s.t.} \quad & (5.1) - (5.25), \end{aligned} \quad (5.26)$$

where $\mathbf{g}(\kappa_0)$ collects all the variables of problem (5.26), \mathcal{P} is the set collecting all the platforms of the line, $\mathcal{N}_p(\kappa_0)$ is the set indices of trains departing from platform p within the prediction time window starting at time step κ_0 , and w_1 and w_2 are weights balancing two objectives.

5.4.2. MINLP-BASED MPC FOR REAL-TIME TRAIN RESCHEDULING

Problem (5.26) contains piecewise constant (“if-then”) constraints in (5.13) and (5.14). We apply the following transformation properties to convert (5.13) into a mixed logical dynamical (MLD) system [139].

Transformation property 5.2: If we introduce an auxiliary continuous variable $o_{k,p}$ and an auxiliary binary variable $\gamma_{k,p}$ with $\gamma_{k,p} = 1 \Leftrightarrow o_{k,p} = y_p(k_p)$ and $\gamma_{k,p} = 0 \Leftrightarrow o_{k,p} = -y_p(k_p)$, and then $o_{k,p} = |y_p(k_p)|$ is equivalent to

$$\begin{cases} o_{k,p} - y_p(k_p) \geq 0, \\ o_{k,p} - y_p(k_p) \leq 2Y_{\max}(1 - \gamma_{k,p}), \\ o_{k,p} + y_p(k_p) \geq 0, \\ o_{k,p} + y_p(k_p) \leq 2Y_{\max}\gamma_{k,p}, \end{cases} \quad (5.27)$$

where Y_{\max} denotes the maximum value of $y_p(k_p)$.

Transformation property 5.3: Based on *Transformation property 5.2*, (5.11) is equivalent to $\eta_{k,p} = \begin{cases} 1, & \text{if } o_{k,p} > 0; \\ 0, & \text{otherwise,} \end{cases}$ which can be converted to

$$\begin{cases} o_{k,p} \leq \eta_{k,p} O_{\max}, \\ o_{k,p} \geq \epsilon + (1 - \eta_{k,p})(O_{\min} - \epsilon). \end{cases} \quad (5.28)$$

where O_{\max} and O_{\min} are the minimum and maximum values of $o_{k,p}$, respectively, and ϵ is a sufficiently small number, typically representing machine precision.

Therefore, (5.13) can be transformed to

$$\tau_p^{\text{add}}(k_p) = \eta_{k,p} t_p^{\text{cons}}. \quad (5.29)$$

Transformation property 5.4: If we define m_a and M_a as the minimum and maximum values of $a_{p'}(k_{p'})$, respectively, then following the transformation property in [7], (5.14) is equivalent to the following inequalities

$$\begin{cases} a_{p'}(k_{p'}) - a_p(k_p) \leq \left(1 - \xi_{k_{p'}, k_p, p', p}\right) (M_a - a_p(k_p)), \\ a_{p'}(k_{p'}) - a_p(k_p) \geq \epsilon + \xi_{k_{p'}, k_p, p', p} (m_a - a_p(k_p) - \epsilon). \end{cases} \quad (5.30)$$

Based on the transformations described above, we can convert problem (5.26) into an MINLP problem. The nonlinearity in the MINLP arises from the nonlinear objective function. The integer variables in this problem encompass the train composition variables ($y_p(k_p)$ and $\ell_p(k_p)$), the train ordering variable ($\xi_{k_p, k_{p'}, p, p'}$), and auxiliary binary variables ($\gamma_{k_p, p}$ and $\eta_{k_p, p}$).

For compactness, we rewrite the resulting MINLP problem in the following form:

$$\min_{\mathbf{x}(\kappa_0), \mathbf{u}(\kappa_0), \boldsymbol{\delta}(\kappa_0)} J(\kappa_0) := \sum_{\kappa=\kappa_0}^{\kappa_0+N-1} L(x(\kappa), u(\kappa), \delta(\kappa)) \quad (5.31a)$$

$$\text{s.t. } x(\kappa + 1) = A_\kappa x(\kappa) + B_{1,\kappa} u(\kappa) + B_{2,\kappa} \delta(\kappa), \quad (5.31b)$$

$$D_{3,\kappa} x(\kappa) + D_{1,\kappa} u(\kappa) + D_{2,\kappa} \delta(\kappa) \leq D_{4,\kappa}, \quad (5.31c)$$

$$\kappa = \kappa_0, \dots, \kappa_0 + N - 1,$$

where N is the total number of time steps, $x(\kappa)$, $u(\kappa)$, and $\delta(\kappa)$ collect all the independent variables, continuous decision variables, and discrete decision variables for time step κ , respectively, and $\mathbf{x}(\kappa_0) = [x^\top(\kappa_0), x^\top(\kappa_0 + 1), \dots, x^\top(\kappa_0 + N - 1)]^\top$, $\mathbf{u}(\kappa_0) = [u^\top(\kappa_0), u^\top(\kappa_0 + 1), \dots, u^\top(\kappa_0 + N - 1)]^\top$, and $\boldsymbol{\delta}(\kappa_0) = [\delta^\top(\kappa_0), \delta^\top(\kappa_0 + 1), \dots, \delta^\top(\kappa_0 + N - 1)]^\top$. In (5.31), $L(x(\kappa), u(\kappa), \delta(\kappa))$ represents the nonlinear objective function for time step κ , (5.31b) collects all equality constraints, and (5.31c) collects all inequality constraints.

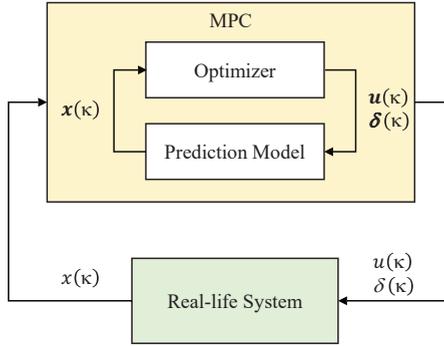


Figure 5.4: Model predictive control for real-time train rescheduling.

Solving (5.31) leads to a series of continuous decision variables and discrete decision variables, and only the decision variables at time step κ_0 are applied. At the next time step, the prediction time window is shifted for one step, and a new optimization is formulated. The framework is depicted in Fig. 5.4.

Lemma 5.5 (Recursive Feasibility): If problem (5.31) is feasible at time step κ with initial state $x(\kappa)$, then problem (5.31) is also feasible at time step $\kappa + 1$.

Proof. The proof relies on finding a feasible solution for time step $\kappa + 1$. Recall that the planning time window at each platform is divided into several intervals of equal length, with a train service departing from the platform at each interval according to (5.1). In general, there are two types of depots: (i) depots connected to the terminal platform

(e.g., Platform 1 in Fig. 5.2), and (ii) depots connected to intermediate platforms (e.g., Platform 2 in Fig. 5.2).

i) For a depot connected to the terminal platform: if the problem (5.31) is feasible at time step κ , then at each time step, a train service returns to the depot from the opposite direction of the line. In this context, the new train service departing from the terminal platform at step $\kappa + 1$ can directly utilize the train units by performing a turnaround from the opposite direction of the line following (5.7).

ii) For any depot connected to an intermediate platform: a feasible solution is obtained by maintaining the composition of each train service the same as it was at time step κ . \square

5.4.3. MILP-BASED MPC FOR REAL-TIME TRAIN RESCHEDULING

In Section 5.4.1, a nonlinear objective function is defined in (5.24) to calculate the passenger delays. The nonlinear objective function yields the MINLP-based MPC in Section 5.4.2. In general, the nonlinear term significantly increases the computational burden. In this context, we simplify the nonlinear objective function to reduce the computational burden.

As we divide the planning time window as in Fig. 5.3 and the actual departure time is constrained by $d_p^{\text{pre}}(k_p) \leq d_p(k_p) < d_p^{\text{pre}}(k_p + 1)$, then by using the upper bound and lower bound of $d_p(k_p)$ we can approximate the nonlinear objective function (5.24) by

$$J_p^{\text{pass}}(k_p) = w_3 n_p(k_p) (d_p^{\text{pre}}(k_p + 1) - d_p^{\text{pre}}(k_p)) + n_p^{\text{after}}(k_p) (d_p^{\text{pre}}(k_p + 1) - d_p^{\text{pre}}(k_p)), \quad (5.32)$$

where w_3 is a weight used to balance the approximated errors. In particular, w_3 can be defined as

$$w_3 = \frac{1}{\sum_{p \in \mathcal{D}} |\mathcal{S}_p|} \sum_{p \in \mathcal{D}} \sum_{k_p \in \mathcal{S}_p} \frac{\bar{d}_p(k_p) - d_p^{\text{pre}}(k_p)}{d_p^{\text{pre}}(k_p + 1) - \bar{d}_p(k_p)}, \quad (5.33)$$

where $|\mathcal{S}_p|$ represents the cardinality of \mathcal{S}_p , and $\bar{d}_p(k_p)$ represents the average value of $d_p(k_p)$ from historical data.

Other settings are identical to those of Section 5.4.2. With this approximation, a linear objective function (5.24) is obtained, and due to the objective function and constraints being all linear, we obtain an MILP-based MPC approach for real-time train rescheduling.

5.5. LEARNING-BASED MPC FOR REAL-TIME TRAIN RESCHEDULING

For the MINLP-based MPC in Section 5.4.2, an MINLP problem should be solved at each step, which is typically not computationally affordable for real-time application as the number of integer variables significantly influences the computational complexity. To handle the computational complexity issues, we develop a learning-based model predictive control approach where the integer variables are obtained by deep learning, and then the MPC optimizer only needs to solve a continuous nonlinear optimization problem with fewer variables than the original problem at each time step.

The framework of the approach is provided in Fig. 5.5. At each step, the algorithm generates discrete variables based on the current state. Once the discrete variables have been determined, the MPC optimization problem of Section 5.4.2 becomes a continuous-variable nonlinear programming (NLP) problem, while the MPC optimization problem of Section 5.4.3 reduces to a linear programming (LP) problem. By solving the resulting NLP or LP problem, the optimal continuous variable values can be obtained, and the new state of the railway network can be obtained after implementing the obtained discrete and continuous variables.

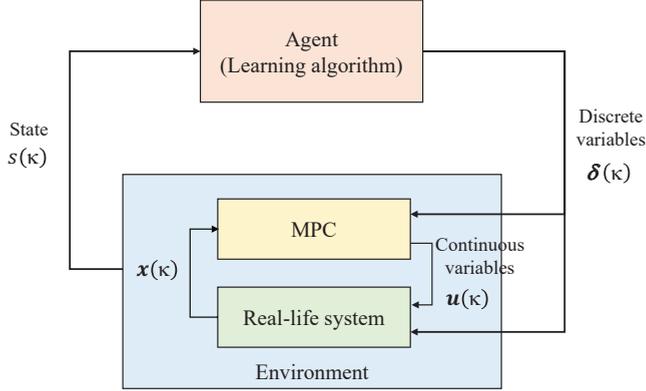


Figure 5.5: Learning-based MPC for real-time train rescheduling.

5.5.1. PRESOLVE TECHNIQUES

Presolve techniques streamline optimization processes by pruning a subset of decision variables with values predetermined by other coupled variables and constraints, setting them to predefined values. In this chapter, we consider the following presolve techniques.

Presolve technique 1: As we adjust the departure time of train service k at platform p according to $d_p^{\text{pre}}(k_p) \leq d_p(k_p) < d_p^{\text{pre}}(k_p + 1)$, the departure order between some trains has already been determined: If $d_p^{\text{pre}}(k_p) \geq d_{p'}^{\text{pre}}(k_{p'} + 1) + t_{p'}^{\text{roll}}$, then $\xi_{k_p, k_{p'}, p, p'} = 1$. If $d_p^{\text{pre}}(k_p + 1) \leq d_{p'}^{\text{pre}}(k_{p'}) + t_{p'}^{\text{roll}}$, then $\xi_{k_p, k_{p'}, p, p'} = 0$.

Presolve technique 2: According to the definition of $\xi_{k_p, k_{p'}, p, p'}$ in (5.14), the order of trains at the same platform should be kept consistent, i.e., $\xi_{k_p, k_{p'}, p, p'} \geq \xi_{k_{p+1}, k_{p'}, p, p'}$.

Presolve technique 3: The train composition cannot be changed at a station that is not linked with a depot: If $\sigma_p = 0$, then $y_p(k_p) = 0$.

Presolve technique 4: Let $t_0 = \kappa_0 T$ represent the current time. If train service k_p has already departed from station p at time t_0 , the composition cannot be changed at that station: If $d_p(k_p) \leq t_0$, $y_p(k_p) = y_p^*(k_p)$, $\forall p \in \mathcal{P}$, where $y_p^*(k_p)$ represents the value of $y_p(k_p)$ obtained before train k departs from platform p and \mathcal{P} denotes the set of all platforms of the line.

5.5.2. ENVIRONMENT SETTING

The environment of the Learning-based MPC algorithm includes the system and an MPC optimization problem. The state and variables that interact with the environment are defined as follows:

State $\mathbf{s}(\kappa) \in S$: The state space ought to encompass all necessary information of the framework so that the neural network can be trained such that the input can capture as much possible situations as possible. Hence, state at time step k is defined as:

$$\mathbf{s}(\kappa) = [\mathbf{n}^\top(\kappa), \boldsymbol{\rho}^\top(\kappa), \mathbf{N}^\top(\kappa)]^\top, \quad (5.34)$$

where $\mathbf{n}(\kappa)$ includes the variables $n_p(k_p)$ for train service k_p at its corresponding platform p with $\kappa T \in [d_p^{\text{pre}}(k_p - 1), d_p^{\text{pre}}(k_p))$, $\boldsymbol{\rho}(\kappa)$ collects the passenger demands $\rho_p(k_p)$ for all train services k_p departing from all platforms p from time $t = \kappa T$ to the end of the prediction time window, and $\mathbf{N}(\kappa)$ collects the number of available trains for all depots at time $t = \kappa T$.

Discrete variables $\boldsymbol{\delta}(\kappa) \in A$: The discrete variables correspond to the discrete variables of the MPC optimization problem (5.31) in each step. Before we evaluate the discrete variable, we first implement the presolve techniques in Section 5.5.1 to avoid infeasible actions, and then, we solve the resulting problem corresponding to the discrete action.

Continuous variables $\mathbf{u}(\kappa) \in U$: The continuous variables represent the continuous decision variables at time step κ in the MPC optimization problem (5.31).

5.5.3. OFFLINE TRAINING FOR LEARNING-BASED MPC ALGORITHM

In practice, the train schedules across consecutive time intervals are interdependent due to the headway relation between trains and the physical connections between stations. The neural network should be able to capture and retain essential information over sequences time intervals. Therefore, a long short-term memory (LSTM) network [50] is implemented to train the agent. As a deep recurrent neural network (RNN), the LSTM architecture enables the network to remember the dynamic interdependencies within train schedules, ensuring effective adaptation and learning in response to evolving temporal dynamics.

The training procedure of the learning-based MPC approach is shown in Fig. 5.6. At each step, the LSTM network takes the current state as input, and the hidden state h_κ is initially processed through a feedforward layer to generate objective function values for all possible discrete variables. The hidden state h_κ and c_κ are then passed to the LSTM network at the next time step.

The mean squared error (MSE) is applied to update the parameters of the LSTM network as the optimizer with the following loss function:

$$L^{\text{cross}} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \sum_{\kappa=\kappa_0}^{\kappa_0+N-1} (J_s^*(\kappa) - J_s(\kappa, \boldsymbol{\delta}(\kappa)))^2, \quad (5.35)$$

where L^{cross} represents the loss function, \mathcal{S} defines a set collecting all transitions, $|\mathcal{S}|$ denotes the cardinality of \mathcal{S} , $J_s^*(\kappa)$ is the optimal value of the objective function of transition s at time step κ , and $J_s(\kappa, \boldsymbol{\delta}(\kappa))$ is the value of objective function of transition s at time step κ with discrete variable $\boldsymbol{\delta}(\kappa)$.

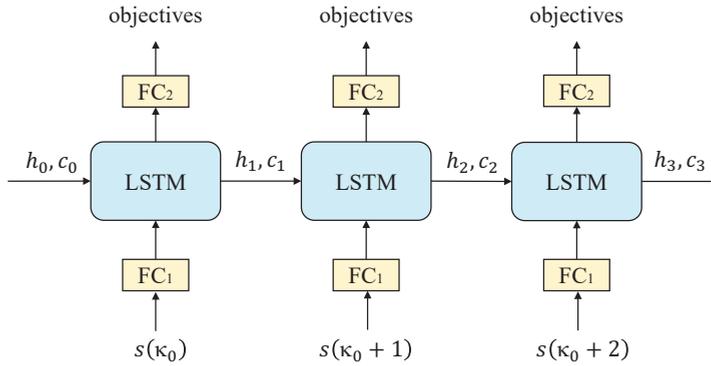


Figure 5.6: Training procedure of the learning-based MPC approach with LSTM.

5

5.6. CASE STUDY

5.6.1. BASIC SETTING

In this section, we illustrate the proposed approaches based on real-life data from a network of three lines in the Beijing urban rail transit network. As shown in Fig. 5.7, the network including 3 bi-directional lines with 45 stations. There are 3 transfer stations, i.e., Station ZXZ, Station XEQ, and Station HY, where passengers can transfer from one line to another. For each line, there is a depot connected with the starting station of the line, i.e., Station CPX for Changping Line, Station XZM for Line 13, and Station ZXZ for Line 8. The values of parameters for the case study are given in Table 5.4. The original timetable is generated based on the regular headway, regular dwell time, and average running times in Table 5.4. According to the definition, the length of a time step is the sum of the regular headway and the regular dwell time. The number of train units in the depot for each line has been selected as a random integer number with the value varying among the range given in Table 5.4.

The length between every two consecutive stations is openly accessible on the website of Beijing Subway¹. In the case study, the average running time and the average energy consumption of a train between every two consecutive stations are calculated using the method in [135] with the maximum acceleration of 0.75 m/s^2 , the maximum deceleration of 0.7 m/s^2 , and the cruising speed of 70 km/h , respectively. The sectional passenger demands are obtained based on real-life passenger flow data from the Beijing urban rail transit network, collected in January 2020. We have selected data from 6:00 AM to 10:00 PM for simulation, so the data contains both peak hours and off-peak hours. For training agents and simulation, we have generated passenger demands based on a Poisson distribution by using real-life passenger flow data as the expected value.

The simulations have been conducted using Python as a programming language, PyTorch as the machine learning library, and gurobi to solve optimization problems. Adam [59] is applied in the offline training process to minimize MSE, and dropout [114] is used to handle the over-fitting issue. Moreover, the experiments were conducted on a

¹<https://www.bjsubway.com/station/zjgls/>

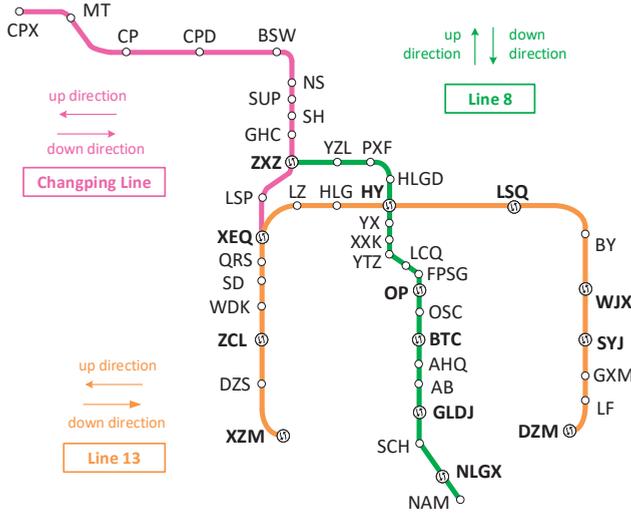


Figure 5.7: The layout of considered urban rail transit network (with 3 lines).

Table 5.4: Main parameters for the case study

Parameter	Symbol	Value
Regular headway	h_p^{regular}	180 s
Minimum headway	h_p^{min}	120 s
Regular dwell time	τ_p^{regular}	60 s
Minimum dwell time	τ_p^{min}	30 s
Average turnaround time	r_p^{turn}	52.9 s
Minimum running time	r_p^{min}	$0.8 \cdot r_p^{\text{avrg}}$
Maximum running time	r_p^{max}	$1.2 \cdot r_p^{\text{avrg}}$
Time for changing train composition	t_p^{cons}	60 s
Time for rolling stock circulation	t_p^{roll}	240 s
Transfer rate at a transfer station	$\beta_{q,p}$	10%
Capacity of a train unit	C_{max}	400 persons
Regular train composition of a train service	ℓ_p^{regular}	2 train units
Minimum number of train units included in a train service	ℓ_p^{min}	1 train unit
Maximum number of train units included in a train service	ℓ_p^{max}	4 train units
Weighted term	w_1	10^{-4}
Weighted term	w_2	10^{-1}
Weighted term	w_3	10^{-1}
Number of train units for Changping Line	N_z^{train}	[55, 75]
Number of train units for Line 13	N_z^{train}	[70, 90]
Number of train units for Line 8	N_z^{train}	[60, 80]
Prediction horizon of MPC	N	40

computing cluster with Intel XEON E5-6248R CPUs. The dataset consists of 96000 states and the corresponding optimal solutions, and it has been built using 60 CPU cores and 240GB of RAM in 24 hours. The training and hyperparameter tuning processes have been conducted using 144 CPU cores, 864GB of RAM, using more than 200000 iterations for 24 hours.

To reduce the solution time of the MINLP solver without significantly compromising optimality, a simple early termination criterion was employed: if the optimality gap does not decrease by 0.5% within 10 seconds, the solution process terminates, and the solver outputs the best solution found. Moreover, as the MILP-based approach typically has a significantly shorter solution time than the MINLP-based approach, the integer variables generated by the MILP-based approach are used as a warm-start rule for the MINLP-based approach.

In this section, the developed train scheduling approaches, i.e., MINLP-based MPC, warm-start-MINLP-based MPC, MILP-based MPC, learning-NLP-based MPC, and learning-LP-based MPC are evaluated. As defined in Section 5.3, the length of a time step is 240 s. Hence, to ensure that a solution can be obtained for each time step, we set the maximum solution time for each approach as 240 s. In addition, as a longer solution time typically yields better objective function value, we use the MINLP approach with warm-start and a longer maximum solution time, i.e., 600 s, as a benchmark to evaluate the performance of the developed approaches.

As the inference time of the LSTM networks and the time required for feasibility checks are sufficiently low, instead of using a single LSTM network, an ensemble of 15 LSTM networks was employed in a sequential way to generate integer variables for the learning-based approaches to improve the overall feasibility of the learning-based approaches. The inference process of the LSTM networks in the ensemble is performed sequentially, where the $(i+1)$ th network is evaluated only if the i th and all preceding networks fail to produce a feasible solution. In particular, the ensemble consists of LSTM networks with hidden sizes from the set {512, 1024}, dropout rates from the set {0, 0.5}, learning adjusting in the set {on, off}, and output masking in the set {on, off}.

5.6.2. SIMULATION AND RESULTS

We have conducted simulations to train the learning algorithm, and the learning process for one of the networks is shown in Fig. 5.8, where a 1000-step moving average approach is applied to smooth the learning curve. From Fig. 5.8, we can see that the learning curve converges very quickly during the first 100000 iterations, and then the performance gradually improves until iteration 200000. We have performed simulations for open-loop control using the developed learning-based approaches across more than 1000 different scenarios, i.e., 1038 different scenarios. For comparison, simulations have been also performed with the MINLP approach, the warm-start MINLP approach, MILP approach.

The optimality gap, CPU time, and feasibility rate among the simulations are given in Table 5.5. From Table 5.5, we see that the MINLP approach without warm-start has the worst optimality gap and the worst CPU times. In general, the nonlinear objective function and the integer terms significantly influence the performance of MINLP. By applying the warm start, the optimality gap and the solution time are reduced; however,

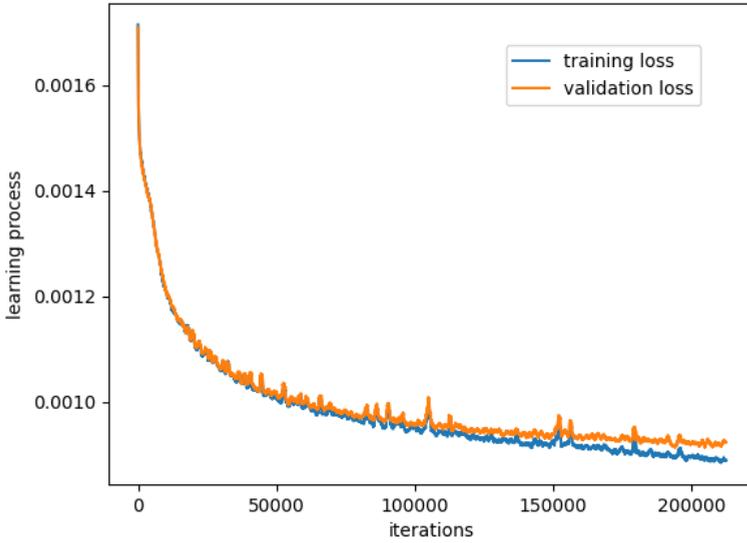


Figure 5.8: Learning process of the learning algorithm.

Table 5.5: Simulation results for different train scheduling approaches

Approach	Optimality gap			CPU time (s)			Feasibility rate
	max	average	min	max	average	min	
Benchmark	-	-	-	600.10	222.18	3.67	100%
MINLP	100.81%	7.22%	0%	240.14	239.84	52.47	100%
Warm-start MINLP	12.96%	0.04%	-0.24 %	240.10	96.71	3.58	100%
MILP	1.54%	0.47%	-33.73%	240.01	8.77	0.37	100%
Learning + NLP	3.48%	-0.11%	-34.28%	112.22	6.89	1.80	98.94%
Learning + LP	1.73%	0.22%	-33.42%	0.25	0.13	0.11	98.55%

the maximum solution time and the average solution time of the warm-start MINLP approach are still large with values of 240.10 s and 96.71 s, respectively. By approximating the nonlinear objective function, the solution time of the MILP approach is further reduced without sacrificing too much optimality. However, the MILP also reached the maximum allowed solution time in some cases, i.e., the maximum CUP time of MILP in Table 5.5 still reached 240 s.

The developed learning-based NLP approach and learning-based LP approach achieve comparable performance with an average optimality gap of -0.11% and 0.22% among the feasible cases, while significantly reducing the solution time to an average of 6.89 s and 0.13 s, respectively. Furthermore, by approximating the nonlinear objective function, the learning-based LP approach further reduces the solution time, enabling the optimized timetable to be obtained in under 1 second. Both the learning-based NLP and LP approaches demonstrate high feasibility rates, at 98.94% and 98.55%, respectively. For the infeasible case, the heuristic as stated in Lemma 5.5 can be applied to get the train decomposition and generate a feasible timetable.

5

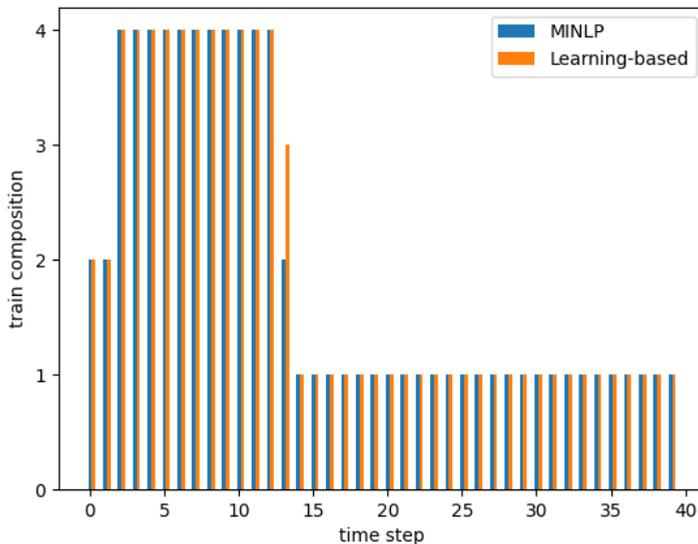


Figure 5.9: Number of train units departing from Station CPX at each time step.

To further show the performance of the learning-based approach, the number of train units departing from Station CPX at each time step is given in Fig. 5.9, where time step 1 corresponds to 10:40 AM. The train decomposition and timetable before 10:40 AM are aligned with the regular timetable with parameters indicated in Table 5.4. The results indicate that both the learning-based approach achieve performance nearly equivalent to the MINLP-based approach in most cases with the learning-based approach showing a deviation at only one step in Fig. 5.9. In practice, rail operators typically expect to

obtain a timetable as quickly as possible. The simulation results indicate that the developed learning-based approaches can be applied to generate timetables that minimize passenger delays and train energy consumption within a relatively short time.

5.7. CONCLUSIONS

In this paper, we have investigated the passenger-oriented train rescheduling problem considering flexible train composition and rolling stock circulation. The passenger-oriented train scheduling model [76] has been extended to include train compositions and rolling stock circulation considering time-varying passenger demands. To improve the online computational efficiency of model predictive control, we have combined the optimization-based and learning-based approaches, where the learning-based approach obtains integer variables, i.e., train compositions and train orders, by using pre-trained long short-term memory networks; then, the detailed timetables are optimized by solving a constrained optimization problem with the fixed integer variables. We have developed several presolve techniques to prune the subset of integer decision variables. Simulation results show that the developed learning-based framework can achieve comparable performance compared to the exact approach with an acceptable loss of feasibility while the solution time is significantly reduced.

In the future, we will investigate the integration of reinforcement learning (instead of supervised learning) and model predictive control for real-time train scheduling to improve the learning ability of the approach. Among several directions, multi-agent learning-based approaches can also be a promising direction for handling large-scale urban rail transit networks.

6

COOPERATIVE DISTRIBUTED MPC FOR VIRTUALLY COUPLED TRAINS

Virtual coupling is regarded as an efficient way to improve the line capacity of rail transportation systems by reducing the spacing between consecutive trains. This study is the first to compare and assess different distributed model predictive control (MPC) approaches, i.e., cooperative distributed MPC, serial distributed MPC, and decentralized MPC, for virtually coupled trains with a nonlinear train dynamic model. To make a balanced trade-off between computational complexity and efficiency, we also propose and assess convex approximations of the above control approaches. Furthermore, we are the first to introduce the relaxed dynamic programming approach to analyze the stability of the MPC-based nonlinear train control problem. By using the relaxed dynamic programming approach, a distributed stopping criterion with a stability guarantee is developed for the cooperative distributed MPC approach. In real life, masses of trains are different and can change at stations due to changes in passenger loads. This change in mass can significantly affect the dynamics and control of the virtually coupled trains when not taken into account in the control design. Therefore, we explicitly consider heterogeneous train masses when designing MPC approaches. We evaluate the different distributed MPC approaches through case studies based on the data of the Beijing Yizhuang Line. Simulation results indicate that the cooperative distributed MPC approach has the best tracking performance, while the serial distributed MPC approach can reduce communication requirements and computation capabilities with sacrifices of tracking performance.

6.1. INTRODUCTION

The transport demand for rail transportation systems has increased rapidly, and the need to enhance rail line capacity while ensuring operational safety remains a paramount concern for rail operators. The line capacity is associated with the spacing between consecutive trains, which is determined by the signal systems. Currently, the widely applied signal system in urban rail transit is the moving block system [3; 96], which determines the distance between two consecutive trains based on the absolute braking distance, i.e., the distance a train needs to fully stop from its current speed.

In recent years, an advanced signaling technology, i.e., virtual coupling, has been recognized as an efficient way to further improve the line capacity by reducing the spacing between consecutive trains [3; 11]. In a platoon of virtually coupled trains, the distance between two consecutive trains is determined based on the relative braking distance, which also takes into account the braking characteristics of the predecessor train [81; 106]. Different from platoons of connected and automated vehicles (CAVs) in road traffic [67], a platoon of virtually coupled trains features a long train braking distance, and trains in a platoon should run on the same rail track, leading to larger spacing between trains. Furthermore, the communication between non-adjacent trains is typically not considered due to the longer headway in railway systems as communication over longer distances may become unreliable [113]. Hence, one cannot just adopt control approaches of CAVs to virtually coupled trains.

As a novel signaling technology, virtual coupling significantly relies on vehicle-to-vehicle communication and cooperative train control schemes [2; 140; 146]. Generally, the communication topology and the cooperative control schemes are highly intertwined. Several control approaches have been developed for virtually coupled trains based on different communication topologies. Cao *et al.* [20] applied generalized predictive control (GPC) to virtually coupled trains with the aim to ensure the expected tracking distance and to prevent collisions. Xun *et al.* [145] applied model predictive control (MPC) to realize centralized control and in addition they developed a speed protection mechanism for virtually coupled trains. Su *et al.* [118] developed a centralized MPC approach for virtually coupled trains in the cruising phase, and they applied a generalized minimum-residual-method-based approach to solve the resulting nonlinear optimization problem. The above papers focus on centralized control approaches that rely on a centralized controller, thereby significantly increasing the communication and computation burden [90].

In contrast to those centralized control approaches, decentralized control strategies have gained attention due to their potential to alleviate the communication and computation burden. Felez *et al.* [33] formulated a decentralized MPC approach for virtually coupled trains based on a linear model with nonlinear constraints. Two cases are considered in [33], i.e., the case that the follower receives predicted states from its predecessor train, and the case that the followers have to predict the states of its predecessor train based on the measured information. Considering uncertainties in the dynamic model and train positioning, Felez *et al.* [35] developed a decentralized robust MPC approach based on the min-max principle. This work is further extended in [126] by including more uncertain factors, such as modeling errors, positioning errors, communication delays, and possible adhesion losses. Di Meo *et al.* [32] developed a decentralized control

approach based on local state variables and the information received from other trains, and they analyzed the exponential stability under communication delays by introducing a Lyapunov-Krasovskii function. By using sliding mode control (SMC) and a nonlinear train control model, Park *et al.* [100] developed a robust gap controller based on the measurement of the position and velocity of the predecessor trains. Basile *et al.* [6] developed a deep deterministic policy gradient approach to design a decentralized control law for virtually coupled trains with heterogeneous train dynamics and uncertain disturbances, showing lower computational burden and energy consumption compared to MPC. However, the safety distance in [6] is considered by using a penalty term in the reward function, which does not provide a theoretical guarantee of safety. The above papers primarily emphasize the significance of decentralized control strategies for virtual coupling, highlighting their ability to alleviate the communication burden while ensuring system performance. However, these decentralized approaches often rely on measurement information or limited communication information, and trains make independent decisions without coordinating their actions with those of other trains, highlighting the potential for distributed and/or cooperative control approaches¹ that can leverage communication data even more.

The advanced vehicle-to-vehicle communication technology enables communication-enhanced information exchange between virtually coupled trains [113; 140], prompting the exploration of distributed control methods that can leverage more extensive communication data. Quaglietta *et al.* [105] analyzed the safety margin of virtually coupled trains to handle the safety risk caused by communication delays, control delays, positioning errors, and train braking characteristics. Su *et al.* [119] considered the heterogeneous train braking distance and developed a feedback control law to ensure the string stability of the train platoon. Liu *et al.* [81] linearized the train movement model and developed a distributed MPC approach for a platoon of virtually coupled trains, where trains are assumed to be close to each other, and therefore the slope difference between different trains is ignored; then, they analyzed the local stability of each individual train based on a terminal controller. By ignoring the slope difference between trains, Liu *et al.* [82] developed an optimal control (OC) approach based on Pontryagin's principle, and analyzed the local stability and the head-to-tail string stability. By considering the resistance caused by tracks and winds as bounded disturbances, Luo *et al.* [88] introduced tube-based distributed MPC based on a linear train model, where the safety constraint can be ensured in any situation in the robust control scheme. In the aforementioned distributed control approaches, each train computes its control input based on the information received from its predecessor train only, and thus the approach is also called the serial distributed control approach. Zhang *et al.* [155] introduced the fixed-time tracking control (FTC) approach and developed a cooperative control approach to achieve virtual coupling within the fixed time. Wang *et al.* [127] introduced a Q-learning-based cooperative control approach for virtually coupled trains where monitoring sensors and wireless communication networks are used to obtain the operational status of trains; however, only two virtually coupled trains are considered in [127], and the extension to more trains still requires further research. In summary, these studies indicate the poten-

¹Each agent in a distributed control scheme only focuses on its own objective, while cooperative distributed control enables agents to take into account the objective of the overall system [115; 116].

tial for enhanced control and coordination among virtually coupled trains facilitated by vehicle-to-vehicle communication technologies.

Table 6.1: Summary of studies on control for virtually coupled trains

Literature	Model	Control scheme	Control approach	Train heterogeneity
Cao <i>et al.</i> (2021) [20]	linear	centralized	GPC	no
Xun <i>et al.</i> (2020) [145]	linear	centralized	MPC	no
Su <i>et al.</i> (2021) [118]	nonlinear	centralized	MPC	no
Felez <i>et al.</i> (2019) [33]	nonlinear	decentralized	MPC	no
Felez <i>et al.</i> (2022) [35]	nonlinear	decentralized	MPC	no
Vaquero-Serrano <i>et al.</i> (2023) [126]	nonlinear	decentralized	MPC	no
Di Meo <i>et al.</i> (2019) [32]	linear	decentralized	PID	no
Park <i>et al.</i> (2020) [100]	nonlinear	decentralized	SMC	train mass
Basile <i>et al.</i> (2024) [6]	nonlinear	decentralized	DDPG	train dynamics
Liu <i>et al.</i> (2021) [81]	linear	distributed	MPC	no
Liu <i>et al.</i> (2021) [82]	linear	distributed	OC	no
Luo <i>et al.</i> (2023) [88]	linear	distributed	MPC	no
Su <i>et al.</i> (2023) [119]	nonlinear	distributed	feedback control	braking dynamics
Zhang <i>et al.</i> (2021) [155]	nonlinear	cooperative	FTC	no
Wang <i>et al.</i> (2020) [127]	nonlinear	cooperative	Q-learning	no
This chapter	linear, nonlinear	cooperative, distributed, decentralized	MPC	train mass

GPC: generalized predictive control; MPC: model predictive control; PID: proportional–integral–derivative; SMC: sliding mode control; DDPG: deep deterministic policy gradient; OC: optimal control; FTC: fixed-time tracking control.

6

In a set of virtually coupled trains, trains may have different characteristics, resulting in heterogeneity. In particular, heterogeneous trains may have different lengths, masses, and braking characteristics, which should be considered in the controller design to ensure efficient and safe operation [106; 119]. Train mass is a crucial factor influencing train dynamics and varies according to train type and passenger load. Therefore, without loss of generality, we focus on train mass in this chapter as an illustrative example of the various aspects of heterogeneous trains.

Table 6.1 summarizes the aforementioned studies, outlining the differences in the model, control scheme, control approach, and train heterogeneity they used. From Table 6.1, we can observe the application of both linear and nonlinear train dynamic models. Notably, the nonlinear model generally yields more accurate results but also comes with a higher computational burden compared to the linear model. According to different communication topologies, different control schemes, i.e., centralized, decentralized, distributed, and cooperative distributed, are studied. We find that MPC stands out as the most widely adopted train control approach in virtual coupling research. For more studies in virtual coupling, we refer to the recent review papers [34; 140; 146]. It is worth noting that only the study presented in [100] explicitly incorporates train masses when designing the controller, and there is still no research on an MPC design for virtually coupled trains explicitly considering masses of trains. Furthermore, a comprehensive comparison and assessment considering different models and different control schemes for virtually coupled trains is still unaddressed in the existing literature.

The chapter contributes to the state of the art as follows:

1. A comprehensive comparison and assessment of distributed MPC approaches for virtually coupled trains are provided, which would benefit the process of control method design and selection for virtually coupled trains.

2. We are the first to incorporate the relaxed dynamic programming (RDP) approach into the train control field and to use it to ensure the stability of the nonlinear train control problem. By using RDP, a stopping criterion under the distributed control scheme with a stability guarantee is developed for the cooperative distributed MPC approach.
3. The mass of trains can significantly affect the dynamics and control of virtually coupled trains if not considered in the control design. We are the first to explicitly account for changes in train masses when designing MPC approaches, and we demonstrate the impact of incorporating train masses in the control design through simulations.

The rest of this chapter is structured as follows. In Section 6.2, the problem statement and preliminaries are provided. In Section 6.3, the mathematical model of the system is provided. In Section 6.4, several distributed MPC approaches are presented. In Section 6.5, we conducted case studies to illustrate the performance of the approaches, and in Section 6.6, the conclusions and the outlook for future works are provided.

6.2. PROBLEM STATEMENT AND PRELIMINARIES

6.2.1. PROBLEM STATEMENT

In a platoon of virtually coupled trains, trains are coupled virtually through train-to-train communication. We consider heterogeneous trains, and in particular, we focus on heterogeneous masses in this chapter. The leader train receives reference signals from the infrastructure and operates following a reference speed profile, and each follower train follows its predecessor train while keeping a safe distance.

Let us define s_i , v_i , and u_i as the position, speed, and control input of train i , respectively, and define Δs_i and Δv_i as the position difference and speed difference between train i and its predecessor train, respectively. As stated in [113], ultra-reliable low-latency communications are typically required when the distance between trains is less than 50 m. Moreover, the latency of 50 ms can be achieved for wireless train control and monitoring system [109; 113]. The field tests and simulations in [131] also indicated that the average transmission delay of train-to-train communications is below 20 ms. Therefore, in this chapter, we only consider the case that a train can communicate with its predecessor train and follower train, and train-to-train communication under a relatively short distance can be ensured. As shown in Fig. 6.1, three possible communication topologies realized in practice are considered, i.e., bidirectional communication, unidirectional communication, and measurement, and different communication topologies require different control methods. The bidirectional communication in Fig. 6.1(a) allows trains to include their neighbors' real-time control inputs, speeds, and positions when generating control inputs. Hence, trains can compute their control inputs in parallel and exchange information with their neighbors [113; 155], which involves adjusting control inputs, to achieve cooperative control; however, the communication burden of bidirectional communication is relatively large.

For the unidirectional communication in Fig. 6.1(b), trains compute control inputs sequentially in the virtual coupled train string: each train computes the control input

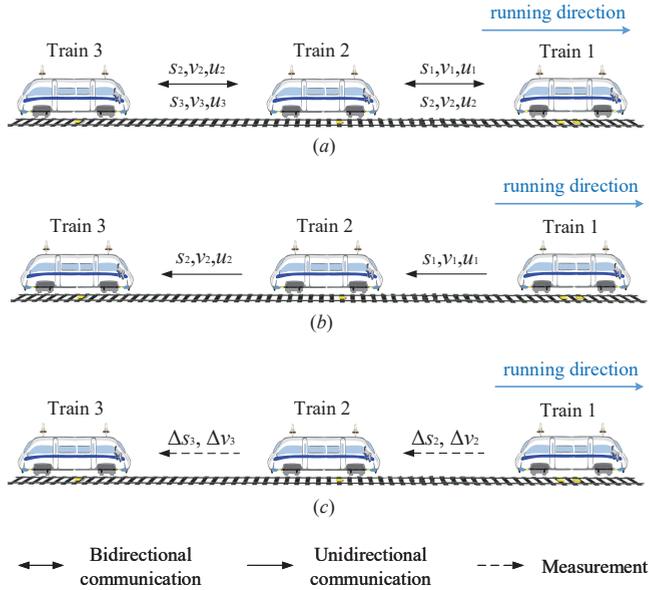


Figure 6.1: Illustration of train-to-train communication topologies for virtually coupled trains occurring in practice.

based on the real-time control input, speed, and position, received from its predecessor train, and then, the computed control input, speed, and position are sent to its successor train. In this context, each train only requires communicating with its neighbors once per control step.

Fig. 6.1(c) corresponds to the case when the communication between trains is lost, and thus a train cannot receive the real-time control input, speed, and position of its predecessor train. Then, to ensure safe operation, each train should compute control inputs based on the relative speed and position of its predecessor train measured by onboard sensors, e.g., radars or LiDARs, assuming the predecessor train may brake with the maximum braking force.

In this chapter, we consider the three communication topologies depicted in Fig. 6.1 and introduce different control approaches based on the three communication topologies.

Remark 6.1. Different from platoons of connected automated vehicles in road traffic, communication between non-adjacent trains is not considered due to the longer headway in railway systems compared to that in road traffic control systems, as communication over longer distances may become unreliable.

The virtual coupling train control problem aims at controlling trains operating with a relatively short headway while ensuring a safe and steady distance between two adjacent trains. The safety distance can be guaranteed by including hard constraints in the control problem. The steady distance between a train and its predecessor train is evaluated by local stability, while the steady distance between any two adjacent trains in the

platoon is ensured by the so-called string stability.

6.2.2. PRELIMINARIES

To introduce the concept of string stability, let us consider train i in the platoon, and the dynamic of train i is

$$x_{i,k+1} = f_i(x_{1,k}, \dots, x_{i,k}, \dots, x_{I,k}), \quad (6.1)$$

where $x_{i,k}$ represents the state of train i at time step k , I is the total number of trains in the platoon.

The definitions of local stability and string stability used in this chapter are introduced as follows.

Definition 6.2 (*Lyapunov Local Stability*) [58]: For a given system (6.1), the equilibrium point x_i^{eq} is said to be Lyapunov local stable if

$$\forall \epsilon > 0, \exists \delta > 0, \|x_{i,0} - x_i^{\text{eq}}\| < \delta \Rightarrow \|x_{i,k} - x_i^{\text{eq}}\| < \epsilon, \forall k \in \mathbb{N}_0. \quad (6.2)$$

In addition, the equilibrium point x_i^{eq} is said to be asymptotically Lyapunov locally stable if it is Lyapunov locally stable and $x_{i,k} \rightarrow x_i^{\text{eq}}$ as $k \rightarrow \infty$.

Let us further define the dynamic platoon of trains as

$$x_{k+1} = f(x_k), \quad (6.3)$$

where $x_k = [x_{1,k}, \dots, x_{I,k}]^T$.

Definition 6.3 (*Lyapunov String Stability*) [36; 122]: For a platoon of trains described by (6.3), the equilibrium point x^{eq} is said to be Lyapunov string stable if

$$\forall \epsilon > 0, \exists \delta > 0, \|x_0 - x^{\text{eq}}\| < \delta \Rightarrow \|x_k - x^{\text{eq}}\| < \epsilon, \forall k \in \mathbb{N}_0. \quad (6.4)$$

In addition, the equilibrium point x^{eq} is said to be asymptotically Lyapunov string stable if it is Lyapunov string stable and $x \rightarrow x^{\text{eq}}$ asymptotically.

Notation: A continuous function $h(\cdot) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is of class \mathcal{K} , if it is strictly increasing and $h(0) = 0$. A continuous function $h(\cdot) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is of class \mathcal{K}_∞ , if it is of class \mathcal{K} and $\lim_{u \rightarrow \infty} h(u) = \infty$. The quadratic norm corresponding to a positive definite symmetric matrix Q is $\|x\|_Q^2 = x^T Q x$. Given a set $\mathbb{X} \subseteq \mathbb{R}^n$, a scalar $a \in \mathbb{R}$, we define $a\mathbb{X} := \{ax | x \in \mathbb{X}\}$.

6.3. MATHEMATICAL MODEL FOR VIRTUALLY COUPLED TRAINS

6.3.1. TRAIN DYNAMIC MODEL

Although the dynamics of a train is continuous, the control input of the automatic train operation (ATO) system is typically implemented in a discrete-time manner due to the implementation of digital computers. Similar to [33; 70], the discrete-time model of longitudinal dynamics of a train can be described as

$$v_{i,k+1} = v_{i,k} + \frac{(u_{i,k} - r_i(v_{i,k}) - w_i(s_{i,k})) T}{M_{i,p}}, \quad (6.5a)$$

$$s_{i,k+1} = s_{i,k} + v_{i,k} T, \quad (6.5b)$$

where i is the train index, T represents the sampling time, $v_{i,k}$ and $s_{i,k}$ represent the speed and position of train i at time step k , respectively, $M_{i,p}$ denotes the total mass of train i from station p to its successor station with p being the station index. We assume that $M_{i,p}$ is a piecewise constant function whose value changes at the station in accordance with the variance of the passenger load. Moreover, $u_{i,k}$ is the control input, i.e., the traction/braking force; $r_i(v_{i,k})$ is the basic resistance that is related to the speed of train i ; $w_i(s_{i,k})$ denotes the additional resistance that is determined by the position of train i .

The total mass of train i changes when train i has arrived at a station and can be calculated by

$$M_{i,p} = m_i + n_{i,p} m_{pa}, \quad (6.6)$$

where m_i denotes the mass of train i itself; $n_{i,p}$ is the number of passengers on board the train at station p , and the value of $n_{i,p}$ changes when the train has arrived at a station; m_{pa} represents the average mass of a passenger.

The train basic resistance $r_i(v_{i,k})$ can be described by

$$r_i(v_{i,k}) = M_{i,p} \left(c_0 + c_1 v_{i,k} + c_2 v_{i,k}^2 \right), \quad (6.7)$$

where c_0 , c_1 , and c_2 are parameters that can be identified based on experiment data [29]. The train basic resistance considers the effects caused by the rotational inertia of wheelsets, the number of axles, the effective frontal cross-section, the air resistance, etc.

The additional resistance $w_i(s_{i,k})$ is related to the total mass of the train and can be approximated as a piecewise constant function of the train position:

$$w_i(s_{i,k}) = M_{i,p} g \sin \theta(s_{i,k}), \quad (6.8)$$

where $\theta(\cdot)$ is a function of train position representing slope at the corresponding position².

The decision variable $u_{i,k}$, the train speed $v_{i,k}$, and the train position $s_{i,k}$ should satisfy

$$-B_i^{\text{sb}} \leq u_{i,k} \leq U_i^{\text{max}}, \quad (6.9)$$

$$0 \leq v_{i,k} \leq v_{\text{lim}}(s_{i,k}), \quad (6.10)$$

$$s_{i,k} + d_i^{\text{safe}}(v_{i,k}, v_{i-1,k}) \leq s_{i-1,k}, \quad (6.11)$$

where B_i^{sb} and U_i^{max} are the maximum service braking force and the maximum traction force of train i , respectively, $v_{\text{lim}}(s_{i,k})$ is a piecewise constant function denoting the speed limit for train i at position $s_{i,k}$, $d_i^{\text{safe}}(v_{i,k}, v_{i-1,k})$ is the safety distance between train i and its predecessor train, which can be constrained by

$$d_i^{\text{safe}}(v_{i,k}, v_{i-1,k}) \geq d_i^{\text{sb}}(v_{i,k}) - d_{i-1}^{\text{eb}}(v_{i-1,k}) + L + D_{\text{safe}}, \quad (6.12a)$$

$$d_i^{\text{safe}}(v_{i,k}, v_{i-1,k}) \geq L + D_{\text{safe}}, \quad (6.12b)$$

²The additional resistance consists of the resistance caused by slope, curve, and tunnel. Note that the curve resistance and the tunnel resistance can be represented by $w_i(s_{i,k}) = M_{i,p} g \gamma(s_{i,k})$, with $0 \leq \gamma(s_{i,k}) \leq 1$; so they can be transformed into the form of (6.8).

where $d_i^{\text{sb}}(v_{i,k}) = \frac{v_{i,k}^2}{2a_i^{\text{sb}}}$ is the braking distance of train i with the service braking, i.e., when $a_i^{\text{sb}} = \frac{B_i^{\text{sb}}}{M_{i,p}}$, $d_{i-1}^{\text{eb}}(v_{i-1,k}) = \frac{v_{i-1,k}^2}{2a_{i-1}^{\text{eb}}}$ is the braking distance of train $i-1$ with emergency braking, i.e., $a_{i-1}^{\text{eb}} = \frac{B_{i-1}^{\text{eb}}}{M_{i-1,p}}$, where B_{i-1}^{eb} is the emergency braking force of train $i-1$, L denotes the length of a train, and D_{safe} is the safety distance applied to address the safety risk caused by modeling errors, positioning errors, communication delays, etc [81; 105].

6.3.2. DYNAMIC MODEL FOR VIRTUALLY COUPLED TRAINS

In a platoon of virtually coupled trains, a train is expected to follow its predecessor train at a certain distance. We consider that the relative distance between train i ($i > 1$) and its predecessor train is determined by the speeds of the two trains:

$$e_{i,k} = s_{i-1,k} - s_{i,k} - d_i^{\text{sb}}(v_{i,k}) + d_{i-1}^{\text{eb}}(v_{i-1,k}). \quad (6.13)$$

The first train ($i = 1$) tracks a desired speed profile with the speed and position represented by $v_{0,k}$ and $s_{0,k}$ respectively, and we define $e_{1,k} = s_{0,k} - s_{1,k}$. The illustration of calculating $e_{i,k}$ in (6.13) is shown in Fig. 6.2.

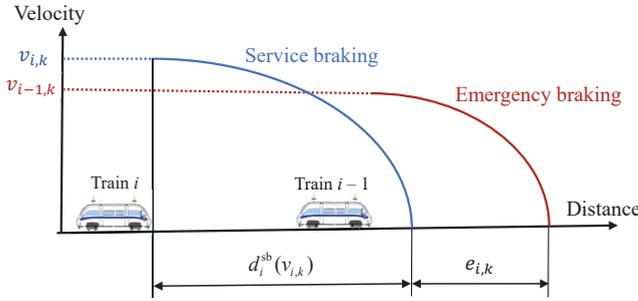


Figure 6.2: Illustration of calculating relative distance for train i ($i > 1$).

Let us define the state and input of train i as $x_{i,k} = [v_{i,k}, e_{i,k}]^\top$ and $\mu_{i,k} = \frac{1}{M_{i,p}} u_{i,k}$, respectively. Then, the evolution of $x_{i,k}$ can be expressed compactly as

$$x_{i,k+1} = f_i(x_{i,k}, \mu_{i,k}), \quad (6.14)$$

where $x_{i,k} \in \mathbb{X}_{i,k}$ and $\mu_{i,k} \in \mathbb{W}_{i,k}$, with $\mathbb{X}_{i,k}$ and $\mathbb{W}_{i,k}$ being the feasible sets of $x_{i,k}$ and $\mu_{i,k}$, respectively. If $u_{i,k} \in \mathbb{U}_{i,k}$, then $\mathbb{W}_{i,k} = \frac{1}{M_{i,p}} \mathbb{U}_{i,k}$. Note that, in (6.14), the states of train i implicitly depend on the position and speed of train $i-1$; hence we have coupled dynamics.

6.4. DISTRIBUTED MODEL PREDICTIVE CONTROL FOR VIRTUALLY COUPLED TRAINS

In this section, we apply different distributed MPC approaches for virtually coupled trains based on the nonlinear model in Section 6.3. We first provide the general non-

linear model predictive control problem formulation. Considering the possible communication structures introduced in Fig. 6.1, the computational complexity, and the model accuracy, we then develop the following six distributed MPC approaches:

- nonconvex cooperative distributed MPC: N-CDMPC;
- convex cooperative distributed MPC: C-CDMPC;
- nonconvex serial distributed MPC: N-SDMPC;
- convex serial distributed MPC: C-SDMPC;
- nonconvex decentralized MPC: N-DMPC;
- convex decentralized MPC: C-DMPC;

N-CDMPC, N-SDMPC, and N-DMPC are related to the bidirectional communication case, the unidirectional communication case, and the measurement case in Fig. 6.1, respectively. However, as the model (6.14) and constraints (6.11), (6.12), and (6.13) are nonlinear, the resulting MPC optimization problems of N-CDMPC, N-SDMPC, and N-DMPC are nonlinear and nonconvex, which may increase the computational burden of finding the optimal solution. Hence, we approximate these problems as convex problems to make a balanced trade-off between computational burden and accuracy, and the convex counterparts of the methods are labeled C-CDMPC, C-SDMPC, and C-DMPC, respectively. The details of the above approaches are provided as follows.

6

6.4.1. GENERAL NONLINEAR MPC PROBLEM FORMULATION

To ensure that trains run with consistent speed and steady distance, we define the quadratic stage cost for train i at time step k as

$$\ell_i(x_{i,k}, \mu_{i,k}) = \|x_{i,k} - x_{i,k}^{\text{eq}}\|_Q^2 + \|\mu_{i,k}\|_R^2, \quad (6.15)$$

where $Q \in \mathbb{R}^{2 \times 2}$ is a positive symmetric matrix, and $R \in \mathbb{R}$. The first term in (6.15) defines the tracking error, while the second term corresponds to the energy consumption of train i . Virtual coupling aims to control trains running with consistent speed and steady relative distance. Thus, the equilibrium state of train i ($i > 1$) is defined as $x_{i,k}^{\text{eq}} = [v_{i,k}^{\text{eq}}, e_{i,k}^{\text{eq}}]^\top$ with $v_{i,k}^{\text{eq}} = v_{i-1,k}$ and $e_{i,k}^{\text{eq}} = L + D_{\text{des}}$, where L is the length of a train, D_{des} represents the desired distance between two trains. The equilibrium state of the first train ($i = 1$) is $x_{1,k}^{\text{eq}} = [v_{0,k}, 0]^\top$.

The nonlinear MPC optimization problem for train i is

$$\min_{\substack{x_{i,k_0} \\ \mu_{i,k_0}}} J_i(k_0) := \sum_{k=k_0}^{k_0+N-1} \ell_i(x_{i,k}, \mu_{i,k}) \quad (6.16a)$$

$$\text{s.t. } x_{i,k+1} = f(x_{i,k}, \mu_{i,k}), \quad k = k_0, \dots, k_0 + N - 1, \quad (6.16b)$$

$$x_{i,k} \in \mathbb{X}_{i,k}, \quad k = k_0, \dots, k_0 + N - 1, \quad (6.16c)$$

$$\mu_{i,k} \in \mathbb{W}_{i,k}, \quad k = k_0, \dots, k_0 + N - 1, \quad (6.16d)$$

where $\mathbf{x}_{i,k_0} = [x_{i,k_0}^\top, \dots, x_{i,k_0+N}^\top]^\top$ and $\boldsymbol{\mu}_{i,k_0} = [\mu_{i,k_0}, \dots, \mu_{i,k_0+N-1}]^\top$, $\mathbb{X}_{i,k}$ denotes the set defined by constraints (6.10)-(6.12), and $\mathbb{W}_{i,k}$ represents the set defined by constraint (6.9).

The optimization problem (6.16) is a nonlinear nonconvex optimization problem. Solving (6.16) at time step k_0 leads to the input sequence $\boldsymbol{\mu}_{i,k_0}^* = [\mu_{i,k_0}^*, \dots, \mu_{i,k_0+N-1}^*]$; only the first value μ_{i,k_0}^* is implemented in the system and the procedure is repeated under a moving horizon scheme.

From (6.16), we formulate a nonlinear model predictive controller, and the stability can be analyzed based on relaxed dynamic programming. The stability condition can be stated as follows.

Theorem 6.4 (*Lyapunov Stability*) [46]. Considering system (6.14) with $x_{i,k} \in \mathbb{X}_{i,k}$, let $\mathbb{X}_{i,k}$ be forward invariant³, and let $\pi_i(\cdot)$ be an admissible control law, i.e., $\pi_i(x_{i,k}) \in \mathbb{W}_{i,k}$, $\forall x_{i,k} \in \mathbb{X}_{i,k}$ such that $f_i(x_{i,k}, \pi_i(x_{i,k})) \in \mathbb{X}_{i,k+1}$. Then, the closed-loop system $x_{i,k+1} = f_i(x_{i,k}, \pi_i(x_{i,k}))$ is asymptotically stable on $\mathbb{X}_{i,k}$ with the equilibrium point $x_{i,k}^{\text{eq}}$ if for all $x_{i,k} \in \mathbb{X}_{i,k}$,

$$J_i^N(k) \geq \alpha \ell_i(x_{i,k}, \pi_i(x_{i,k})) + J_i^N(k+1), \quad (6.17a)$$

$$\beta_1(\|x_{i,k} - x_{i,k}^{\text{eq}}\|_2) \leq J_i^N(k) \leq \beta_2(\|x_{i,k} - x_{i,k}^{\text{eq}}\|_2), \quad (6.17b)$$

$$\ell_i(x_{i,k}, \pi_i(x_{i,k})) \geq \beta_3(\|x_{i,k} - x_{i,k}^{\text{eq}}\|_2), \quad (6.17c)$$

where $J_i^N(k)$ represents the optimized value of $J_i(k)$ at time step k with prediction horizon N , $\beta_1(\cdot)$, $\beta_2(\cdot)$, and $\beta_3(\cdot)$ are of class \mathcal{X}_∞ , and $\alpha \in (0, 1]$ is the relaxed dynamic programming parameter. Note that if $\alpha = 1$, then (6.17a) coincides with the general dynamic programming format.

Remark 6.5. Note that $J_i^N(k+1)$ in condition (6.17a) requires the control law in time step $k+1$, which is not available at time step k . In the MPC scheme, given the optimized control variables at time step k as $\boldsymbol{\mu}_{i,k}^* = [\mu_{i,k}^{*\top}, \dots, \mu_{i,k+N}^{*\top}]^\top$, we can directly build a sequence of feasible control variables for time step $k+1$ as

$$\tilde{\boldsymbol{\mu}}_{i,k+1} = [\tilde{\mu}_{i,k+1}^\top, \dots, \tilde{\mu}_{i,k+N}^\top, \tilde{\mu}_{i,k+N+1}^\top]^\top \quad (6.18)$$

$$= [\mu_{i,k+1}^{*\top}, \dots, \mu_{i,k+N}^{*\top}, \tilde{\mu}_{i,k+N+1}^\top]^\top, \quad (6.19)$$

where $\tilde{\mu}_{i,k+1}^\top, \dots, \tilde{\mu}_{i,k+N}^\top$ are the inputs in $\boldsymbol{\mu}_{i,k}^*$, and $\tilde{\mu}_{i,k+N+1}$ can be any admissible control law, e.g., $\mu_{i,k+N} = -\frac{B_{i,p}^{\max}}{M_{i,p}}$. The cost function for $\tilde{\boldsymbol{\mu}}_{i,k+1}$ is represented by $P_i^N(k+1)$. Thus, we can obtain optimized decision variables at time step $k+1$, such that $J_i^N(k+1) \leq P_i^N(k+1)$. Then, the implementable version of (6.17a) becomes

$$J_i^N(k) \geq \alpha \ell_i(x_{i,k}, \boldsymbol{\mu}_{i,k}^*) + P_i^N(k+1). \quad (6.20)$$

6.4.2. NONCONVEX COOPERATIVE DISTRIBUTED MPC

With the bidirectional communication as in Fig. 6.1(a), trains in a platoon of virtually coupled trains can compute control inputs in parallel and exchange information several

³A family of sets $\mathbb{X}_{i,k}$ is forward invariant if there exists $\mu_{i,k}$ such that $x_{i,k+1} = f_i(x_{i,k}, \mu_{i,k}) \in \mathbb{X}_{i,k+1}$ holds for all $x_{i,k} \in \mathbb{X}_{i,k}$.

times to achieve cooperative control. The alternating direction method of multipliers (ADMM) is an efficient distributed optimization approach for problems with coupled constraints [12]. Therefore, we adopt ADMM to solve the resulting distributed optimization problem in each step of distributed MPC.

For the MPC optimization problem of train i , (6.16c) and (6.16d) collect constraints for $x_{i,k}$ and $\mu_{i,k}$, and we can write (6.16c) and (6.16d) compactly as:

$$h_i(y_{i-1,k}, y_{i,k}, y_{i+1,k}) \leq E_{1,i,k}, \quad (6.21)$$

where $y_{i-1,k} = [x_{i-1,k}^\top, \mu_{i-1,k}^\top]^\top$, $y_{i,k} = [x_{i,k}^\top, \mu_{i,k}^\top]^\top$, $y_{i+1,k} = [x_{i+1,k}^\top, \mu_{i+1,k}^\top]^\top$, and $E_{1,i,k}$ is a constant. We can observe from (6.16) that different subproblems are coupled through constraint (6.21). The coupled constraints can be relaxed by introducing $z_{i,k} \geq 0$ as follows:

$$h_i(y_{i-1,k}, y_{i,k}, y_{i+1,k}) + z_{i,k} = E_{1,i,k}. \quad (6.22)$$

Then, in ADMM, the objective function for train i becomes

$$\begin{aligned} \mathcal{L}_i(k_0) = J_i(k_0) + \sum_{k=k_0}^{k_0+N-1} & \left(\lambda_{i,k}^\top (h_i(y_{i-1,k}, y_{i,k}, y_{i+1,k}) + z_{i,k} - E_{1,i,k}) + \right. \\ & \left. + \frac{\rho}{2} \|h_i(y_{i-1,k}, y_{i,k}, y_{i+1,k}) + z_{i,k} - E_{1,i,k}\|_2^2 \right), \end{aligned} \quad (6.23)$$

where $y_{i,k_0} = [y_{i,k_0}^\top, \dots, y_{i,k_0+N-1}^\top]^\top$, $\rho > 0$ is the augmented Lagrangian parameter, and $\lambda_{i,k}$ represents the Lagrangian multipliers, which are updated by

$$\lambda_{i,k}^{(q+1)} = \lambda_{i,k}^{(q)} + \rho \left(h_i(y_{i-1,k}^{(q+1)}, y_{i,k}^{(q)}, y_{i+1,k}^{(q)}) + z_{i,k} - E_{1,i,k} \right), \quad (6.24)$$

where q represents the iteration index, and $\lambda_{i,k}^{(q)}$ and $y_{i,k}^{(q)}$ are the values of $\lambda_{i,k}$ and $y_{i,k}$ after iteration q , respectively. For more details about ADMM, we refer the readers to [12; 48].

In each iteration, a nonlinear nonconvex optimization problem should be solved. We can use gradient-based approaches, e.g., sequential quadratic programming, to find a solution. ADMM is a distributed optimization approach, and a stopping criterion that can be applied in a distributed manner is required when implementing ADMM in the distributed control scheme.

Lemma 6.6. If $\mathcal{L}_i^N(k)$ represents the optimized value of $\mathcal{L}_i(k)$, one sufficient condition for (6.17a) in the distributed control scheme is

$$\mathcal{L}_i^N(k) \geq \alpha \ell_i(x_{i,k}, \mu_{i,k}^*) + P_i^N(k+1), \quad (6.25)$$

where $\alpha \in (0, 1]$.

Proof. Based on the weak duality theorem, we have

$$\mathcal{L}_i^N(k) \leq J_i^N(k). \quad (6.26)$$

Then, according to (6.20), we have

$$J_i^N(k) \geq \alpha \ell_i(x_{i,k}, \mu_{i,k}^*) + P_i^N(k+1). \quad (6.27)$$

Hence, we can conclude that (6.25) implies

$$J_i^N(k) \geq \alpha \ell_i(x_{i,k}, \mu_{i,k}^*) + J_i^N(k+1). \quad (6.28)$$

□

The iteration of ADMM for N-CDMPC stops when either the stability condition represented by (6.25) is satisfied, or the maximum number of iterations z_{\max} is reached. Based on the aforementioned stopping criteria, ADMM may terminate before reaching its (local) optimal solution. To ensure safe operations, the safety coupled constraints (6.11)-(6.12) can be directly incorporated as a constraint when optimizing (6.23), and the coupled constraint (6.13) is relaxed by (6.23).

Lemma 6.7 (Recursive Feasibility). If a feasible solution that satisfies the stopping criterion (6.25) is found at time step k , the feasibility for the optimization problem (6.16) of each agent at time step $k+1$ can be found.

Proof. The proof is based on finding a feasible solution for time step $k+1$. For a solution $\mu_{i,k}^*$ at time step k , a feasible solution at time step $k+1$ can be found as stated in Remark 6.5. □

Theorem 6.8 (Lyapunov String Stability). If a feasible solution that satisfies the stopping criterion (6.25) can be found, then the platoon of virtually coupled trains is Lyapunov string stable.

Proof. If a feasible solution that satisfies the stopping criterion (6.25) can be found, according to Theorem 6.4, we can show that the equilibrium point of each train is Lyapunov stable. Then, the Lyapunov string stability for the platoon of virtually coupled trains can be obtained following the procedure in [122]. □

Algorithm 4 elaborates the procedure for implementing the cooperative distributed MPC algorithm, where q is the iteration index, and $x_{i,k}^{(q)}$ and $\mu_{i,k}^{(q)}$ represent the values of $x_{i,k}$ and $\mu_{i,k}$ after iteration q , respectively.

6.4.3. CONVEX COOPERATIVE DISTRIBUTED MPC

The problem (6.16) formulated in Section 6.4.1 is a nonlinear nonconvex optimization problem. In the N-CDMPC approach developed in Section 6.4.2, we cannot ensure the convergence of ADMM and the optimal solution to the optimization problem easily. Moreover, solving nonlinear nonconvex optimization problems typically requires a larger computational burden than its convex counterpart.

There are two nonconvex components in the N-CDMPC formulation, i.e., the nonlinear model (6.16b) and constraints (6.22). By using Taylor expansion at the prior estimate state of the train, we can linearize $d_i^{\text{sb}}(v_{i,k})$ and $d_{i-1}^{\text{eb}}(v_{i-1,k})$ in (6.12) and (6.13). The prior estimate state of train i at time step $k+1$ can be calculated according to the current speed $v_{i,k+1}$, the current position $s_{i,k+1}$, and control inputs in (6.18) [33; 81]. The nonlinear model (6.16b) can also be linearized at each time step based on the prior estimate state by using Taylor expansion. Other settings are exactly the same as the N-CDMPC approach. Hence, we can simplify the N-CDMPC approach to develop a convex cooperative distributed MPC (C-CDMPC) approach for the platoon of virtually coupled trains.

Algorithm 4 Cooperative Distributed MPC for Virtually Coupled Trains

Input: x_{i,k_0} , $M_{i,p}$, N , U^{\max} , B^{sb} , B^{eb} , I_{train} , D_{safe} , D_{des} , L , k_{end} , q_{max} , ρ , $\lambda_{i,k}^{(0)}$, α ; recommend speeds $v_{0,k}$, $s_{0,k}$;
Output: control input $\mu_{i,k}$

- 1: $k \leftarrow k_0$
- 2: **repeat**
- 3: $q \leftarrow 0$
- 4: **repeat**
- 5: **for** $i = 1, \dots, I_{\text{train}}$ **do**
- 6: minimize objective (6.23) subject to (6.5)-(6.12)
- 7: send obtained $x_{i,k}^{(q+1)}$ and $\mu_{i,k}^{(q+1)}$ to neighbours
- 8: update $\lambda_{i,k}^{(q+1)}$ subject to (6.24)
- 9: **end for**
- 10: $q \leftarrow q + 1$
- 11: **until** $q = q_{\text{max}}$ or (6.25) holds for each train i
- 12: apply control decision $\mu_{i,k}$ to each train i
- 13: $k \leftarrow k + 1$
- 14: **until** $k = k_{\text{end}}$

6

6.4.4. NONCONVEX SERIAL DISTRIBUTED MPC

For the unidirectional communication in Fig. 6.1(b), each train only communicates with its neighbors once in one control step. In this context, each train computes control inputs sequentially based on the information received from its predecessor train. Specifically, train i calculates control inputs based on the speed $\bar{v}_{i-1,k}$, position $\bar{s}_{i-1,k}$, and control input $\bar{\mu}_{i-1,k}$ received from train $i-1$, where $\bar{v}_{i-1,k}$, $\bar{s}_{i-1,k}$, and $\bar{\mu}_{i-1,k}$ are the results of the optimization problem in train $i-1$. Thus, the safety constraints in (6.12) are replaced by

$$d_i^{\text{safe}}(v_{i,k}, \bar{v}_{i-1,k}) \geq d_i^{\text{sb}}(v_{i,k}) - d_{i-1}^{\text{eb}}(\bar{v}_{i-1,k}) + L + D_{\text{safe}}, \quad (6.29a)$$

$$d_i^{\text{safe}}(v_{i,k}, \bar{v}_{i-1,k}) \geq L + D_{\text{safe}}. \quad (6.29b)$$

Furthermore, the relative distance with its predecessor train becomes

$$e_{i,k} = \bar{s}_{i-1,k} - s_{i,k} - d_i^{\text{sb}}(v_{i,k}) + d_{i-1}^{\text{eb}}(\bar{v}_{i-1,k}). \quad (6.30)$$

Then, the cost function becomes

$$\bar{\ell}_i(x_{i,k}, \mu_{i,k}) = \|x_{i,k} - \bar{x}_{i,k}^{\text{eq}}\|_{Q_i}^2 + \|\mu_{i,k}\|_{R_i}^2, \quad (6.31)$$

where $\bar{x}_{i,k}^{\text{eq}} = \left[\bar{v}_{i,k}^{\text{eq}}, \bar{e}_{i,k}^{\text{eq}} \right]^T$ is the equilibrium state of train i , with $\bar{v}_{i,k}^{\text{eq}} = \bar{v}_{i-1,k}$ and $\bar{e}_{i,k}^{\text{eq}} = L + D_{\text{des}}$.

Therefore, in nonconvex serial distributed MPC (N-SDMPC), each train solves the

MPC optimization problem as follows

$$\min_{\substack{x_{i,k_0} \\ \mu_{i,k_0}}} J_i(k_0) := \sum_{k=k_0}^{k_0+N-1} \bar{\ell}_i(x_{i,k}, \mu_{i,k}) \quad (6.32a)$$

$$\text{s.t. } x_{i,k+1} = f(x_{i,k}, \mu_{i,k}), \quad k = k_0, \dots, k_0 + N - 1, \quad (6.32b)$$

$$g_i(\bar{y}_{i-1,k}, y_{i,k}) \leq E_{2,i,k}, \quad k = k_0, \dots, k_0 + N - 1, \quad (6.32c)$$

where (6.32c) is the compact form of constraints corresponding to $y_{i,k} = [x_{i,k}^\top, \mu_{i,k}^\top]^\top$, i.e., constraints (6.9)-(6.11) and (6.29).

The MPC optimization problem (6.32) is a nonlinear nonconvex optimization problem, and we can use gradient-based approaches, e.g., sequential quadratic programming, to find a solution. At each MPC step of N-SDMPC, each train calculated its control input $\mu_{i,k}$ for implementation by solving (6.32) with received $\bar{x}_{i-1,k}$ and $\bar{\mu}_{i-1,k}$, and then send the obtained $x_{i,k}$ and $\mu_{i,k}$ to its succeeding train.

Remark 6.9. As each train only communicates with its neighbors once per control step in the unidirectional communication case, the global optimal solution to the overall problem cannot be guaranteed. The serial distributed MPC approach follows a first-come first-serve fashion for the coupled constraint (6.32c), i.e., the predecessor train calculates and sends states and control inputs to its follower train, and the follower train then calculates states and control inputs that satisfy the coupled constraint (6.32c) based on the received information.

6.4.5. CONVEX SERIAL DISTRIBUTED MPC

To reduce the computational burden of solving the nonlinear nonconvex optimization problem (6.32) for each train, (6.32) can be approximated to develop convex serial distributed MPC (C-SDMPC) for the platoon of virtually coupled trains based on the prior estimate state [81]. See also Section 6.4.3 for detailed information on the convex approximation using the prior estimate state. Then, we can obtain the convex counterpart of (6.32) by linearizing $d_i^{\text{sb}}(v_{i,k})$ and $d_{i-1}^{\text{eb}}(\bar{v}_{i-1,k})$ in (6.29a) and (6.30). Other settings of the C-SDMPC approach are exactly the same as the N-SDMPC approach in Section 6.4.4.

6.4.6. NONCONVEX DECENTRALIZED MPC

The virtually coupled train control approaches should be able to ensure safe operation when the communication between trains is lost, i.e., the case in Fig. 6.1(c). In this context, each train should compute control inputs based on the relative speed and position of its predecessor train measured by onboard sensors, e.g., radars or LiDARs, assuming the predecessor train brakes with the maximum braking force. This leads to a nonconvex decentralized MPC (N-DMPC) approach elaborated in this section.

For train i , the relative speed and position with respect to its predecessor train, i.e., train $i-1$, at time step k are represented by $\Delta v_{i,k}$ and $\Delta s_{i,k}$, respectively, which can be obtained by onboard sensors. At time step k , the estimated speed $\hat{v}_{i-1,k}$ and position $\hat{s}_{i-1,k}$ of train $i-1$ are

$$\hat{v}_{i-1,k} = v_{i,k} + \Delta v_{i,k}, \quad (6.33a)$$

$$\hat{s}_{i-1,k} = s_{i,k} + \Delta s_{i,k}. \quad (6.33b)$$

Then, the predicted state of train $i - 1$ is estimated by assuming the control value as $\hat{\mu}_{i-1,k} = \frac{-B_{i-1}^{\text{eb}}}{M_{i-1,p}}$. Thus, the safety constraints in (6.12) are replaced by

$$d_i^{\text{safe}}(v_{i,k}, \hat{v}_{i-1,k}) \geq d_i^{\text{sb}}(v_{i,k}) - d_{i-1}^{\text{eb}}(\hat{v}_{i-1,k}) + L + D_{\text{safe}}, \quad (6.34a)$$

$$d_i^{\text{safe}}(v_{i,k}, \hat{v}_{i-1,k}) \geq L + D_{\text{safe}}. \quad (6.34b)$$

Furthermore, the relative distance with its predecessor train becomes

$$e_{i,k} = \hat{s}_{i-1,k} - s_{i,k} - d_i^{\text{sb}}(v_{i,k}) + d_{i-1}^{\text{eb}}(\hat{v}_{i-1,k}). \quad (6.35)$$

To ensure the safety operation, train i should follow the desired state $\hat{x}_{i,k}^{\text{eq}} = [\hat{v}_{i,k}^{\text{eq}}, \hat{e}_{i,k}^{\text{eq}}]^\top$ with $\hat{v}_{i,k}^{\text{eq}} = \hat{v}_{i-1,k}$ and $\hat{e}_{i,k}^{\text{eq}} = L + D_{\text{des}}$. Then, the cost function for N-DMPC is

$$\hat{\ell}_i(x_{i,k}, \mu_{i,k}) = \|x_{i,k} - \hat{x}_{i,k}^{\text{eq}}\|_{Q_i}^2 + \|\mu_{i,k}\|_{R_i}^2. \quad (6.36)$$

Hence, the optimization problem of train i for N-DMPC becomes

$$\min_{\substack{x_{i,k_0} \\ \mu_{i,k_0}}} J_i(k_0) := \sum_{k=k_0}^{k_0+N-1} \hat{\ell}_i(x_{i,k}, \mu_{i,k}) \quad (6.37a)$$

$$\text{s.t. } x_{i,k+1} = f(x_{i,k}, \mu_{i,k}), \quad k = k_0, \dots, k_0 + N - 1, \quad (6.37b)$$

$$g_i(\hat{y}_{i-1,k}, y_{i,k}) \leq E_{3,i,k}, \quad k = k_0, \dots, k_0 + N - 1, \quad (6.37c)$$

where (6.37c) collects constraints (6.9)-(6.11) and (6.34). The optimization problem (6.37) is also a nonlinear nonconvex optimization problem. At each MPC step of N-DMPC, each train calculated its control input $\mu_{i,k}$ for implementation by solving (6.37) with estimated $\hat{x}_{i-1,k}$.

6.4.7. CONVEX DECENTRALIZED MPC

Similarly, we can obtain the convex counterpart of optimization problem (6.37), named as convex decentralized MPC (C-DMPC) by linearizing $d_i^{\text{sb}}(v_{i,k})$ and $d_{i-1}^{\text{eb}}(\hat{v}_{i-1,k})$ in (6.34a) and (6.35). Then, the nonlinear model (6.37b) can be linearized at each time step based on the prior estimate state. Other settings of the C-DMPC approach are exactly the same as the N-DMPC approach.

6.5. CASE STUDY

In this section, we conduct simulations to validate the developed distributed MPC approaches. We first introduce general settings for simulations. Then, we perform simulations for a platoon of trains with uniform masses. Finally, we explore simulations involving trains with varying masses.

6.5.1. GENERAL SETUP

The simulations are conducted based on the real-life train operation data of trains on the Beijing Yizhuang Line from Station YH to Station CQ. The values of the main parameters

are provided in Table 6.2. The value of ρ is set as 0.5, and the initial value of $\lambda_{i,k}^{(0)}$ is set as 1. The values of the safety distance and the desired distance are the same as those in papers [33; 35]. The distance from Station YH to Station CQ is 1398.6 m, and the slope and the speed limit information along the line are shown in Fig. 6.3. Model mismatches exist between the control model and the simulation model. The controller design considers the prediction model with the values of the maximum traction and braking forces U_i^{\max} , B_i^{sb} , and B_i^{eb} given in Table 6.2, while the assessment experiments use the simulation model considering the realistic traction and braking characteristics given in Fig. 6.4 (see also [121]).

Table 6.2: Parameters for the controller design

Parameter	Symbol	Numerical value
Prediction horizon	N	5
Sampling time	T	0.2 s
Number of trains	I_{train}	4
Average train mass	$M_{i,p}$	60 t
Resistance parameter	c_0	0.0078
Resistance parameter	c_1	0.00085
Resistance parameter	c_2	0.000076
Maximum traction force	U_i^{\max}	60000 N
Maximum service braking force	B_i^{sb}	48000 N
Emergency braking force	B_i^{eb}	60000 N
Safety distance	D_{safe}	5 m
Desired distance	D_{des}	10 m
Train Length	L	10 m
Weight of tracking error	Q_1	100
Weight of relative speed error	Q_2	1
Weight of control variable	R	1
RDP parameter	α	0.5
Maximum iterations	q_{\max}	5

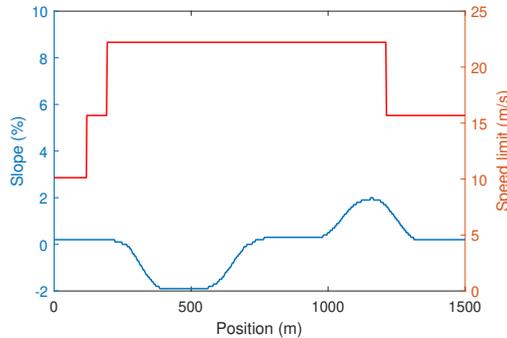


Figure 6.3: Line information from Station YH to Station CQ.

Sequential quadratic programming (SQP) is an efficient gradient-based algorithm for solving nonlinear programming problems [10] and has also been applied to solve the

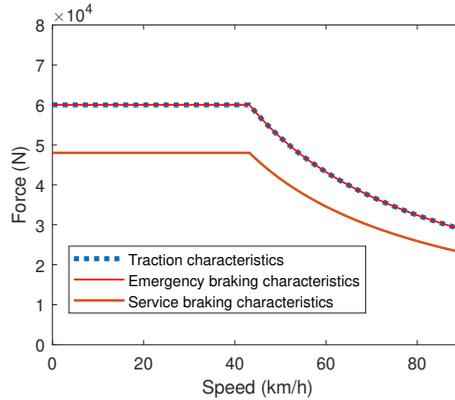


Figure 6.4: Traction and braking characteristics of the simulation model.

optimization problem of virtually coupled trains [81]. Similar to [81], in each MPC step, the resulting optimization problem is solved by SQP through the `fmincon` function in the MATLAB optimization toolbox. All simulations are implemented in MATLAB (R2019b) on a computer with an Intel Xeon W-2223 CPU and 8GB RAM.

6

6.5.2. CONTROL PERFORMANCE WITH UNIFORM TRAIN MASSES

This case study is conducted to evaluate the performance of the distributed MPC approaches in the case that trains in the platoon have the same mass. The parameters are provided in Section 6.5.1. We consider a platoon of 4 trains in the simulation, and all trains weigh 60 t. The simulation results are presented in Table 6.3, wherein the relative distance is calculated as defined in (6.13), and the speed difference represents the velocity error between a train and its predecessor train.

Note that the relative distance represents the distance between two trains, assuming that the predecessor train performs emergency braking and the succeeding train performs service braking. As the train length is set as 10 m and the safety distance D_{safe} is 5 m, the relative distance should be larger than 15 m to ensure safe operation. Furthermore, since the desired distance is 10 m, the ideal relative distance should be 20 m considering the length of the train (i.e., 10 m).

It can be observed from Table 6.3 that convex cooperative distributed MPC (C-CDMPC), convex serial distributed MPC (C-SDMPC), and convex decentralized MPC (C-DMPC) exhibit a performance that is comparable to that of their nonconvex counterparts in terms of the relative distance and the speed difference. The average CPU time is reduced when the underlying problem is convex, with a reduction of 64.25%, 17.86%, and 17.86% for C-CDMPC, C-SDMPC, and C-DMPC, respectively, compared with their corresponding original approaches, indicating that a computational burden reduction is achieved by transforming these problems to their corresponding convex problems. As the performance, in terms of the relative distance and the speed difference, of the original approaches is comparable with their corresponding convex counterparts, we will

focus on C-CDMPC, C-SDMPC, and C-DMPC to compare the performance of different distributed control schemes in the following for brevity.

Table 6.3 shows that all approaches can ensure safe operation when trains have the same mass with a minimum relative distance larger than 15 m. The average relative distance of C-CDMPC and C-SDMPC is close to the ideal relative distance (20 m), while C-DMPC has the largest average relative distance. Furthermore, C-CDMPC exhibits the smallest fluctuation, with the relative distance fluctuating between [18.47 m, 22.62 m] and the speed difference fluctuating between [-1.1071 m/s, 1.3886 m/s].

Table 6.3: Simulation results for different approaches with uniform train masses

Approach	RDP	Total cost	Relative distance (m)			Speed difference (m/s)			CPU time (s)	
			max	average	min	max	average	min	max	average
N-CDMPC	yes	$2.1506 \cdot 10^4$	22.63	19.89	18.50	1.3927	0.0099	-1.1059	4.74	4.00
C-CDMPC	yes	$1.8412 \cdot 10^4$	22.62	19.91	18.47	1.3886	0.0083	-1.1071	1.62	1.43
N-SDMPC	no	$2.8826 \cdot 10^4$	27.64	20.02	19.35	1.6091	0.0025	-1.2273	0.40	0.28
C-SDMPC	no	$2.8825 \cdot 10^4$	27.64	20.02	19.35	1.6091	0.0025	-1.2273	0.30	0.23
N-DMPC	no	$9.1687 \cdot 10^5$	31.50	21.91	19.08	1.7582	0.0035	-1.3627	0.37	0.28
C-DMPC	no	$9.2349 \cdot 10^5$	31.64	21.92	19.08	1.7488	0.0035	-1.3778	0.30	0.23

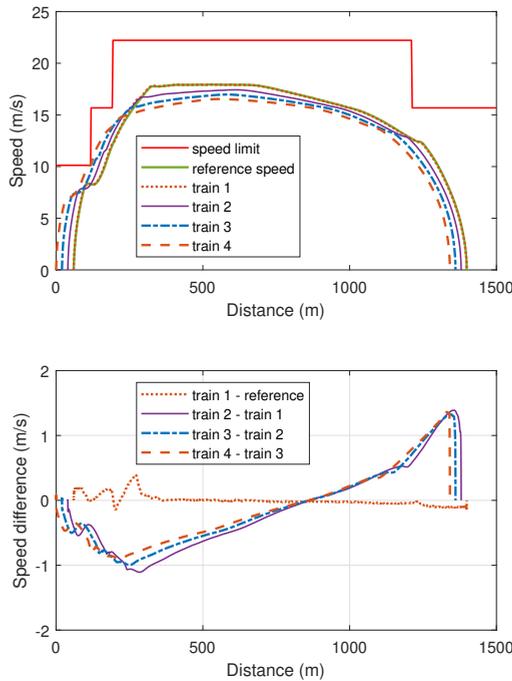


Figure 6.5: Speed profiles and speed difference of C-CDMPC (with the same mass).

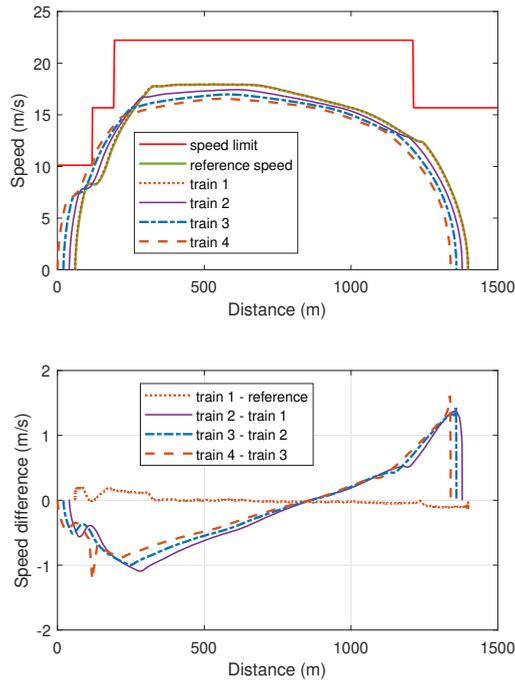


Figure 6.6: Speed profiles and speed difference of C-SDMPC (with the same mass).

For further demonstration, the speed profiles obtained by C-CDMPC, C-SDMPC, and C-DMPC are provided in Fig. 6.5, Fig. 6.6, and Fig. 6.7, respectively, where we include the speed difference between a train and its predecessor train. For the first train, the speed difference denotes the difference with the reference speed. It can be observed from Fig. 6.6 that due to the speed limit, train 4 cannot accelerate, causing a rapid change in speed difference. Thanks to the bidirectional communication as represented in Fig. 6.1(a), the rapid change is avoided in Fig. 6.5, i.e., by using C-CDMPC, a train can include the information of its follower train when calculating its control input, thereby achieving a more homogeneous speed profile via cooperative control. Table 6.3 and Fig. 6.7 show that C-DMPC exhibits the largest fluctuation in both relative distance and the speed difference. As a train cannot receive information from its predecessor train in Fig. 6.1(c), a train should always assume its predecessor train will perform emergency braking. The decentralized control scheme tends to be conservative; thus, the relative distance and the speed difference of C-DMPC are larger than those of C-CDMPC and C-SDMPC.

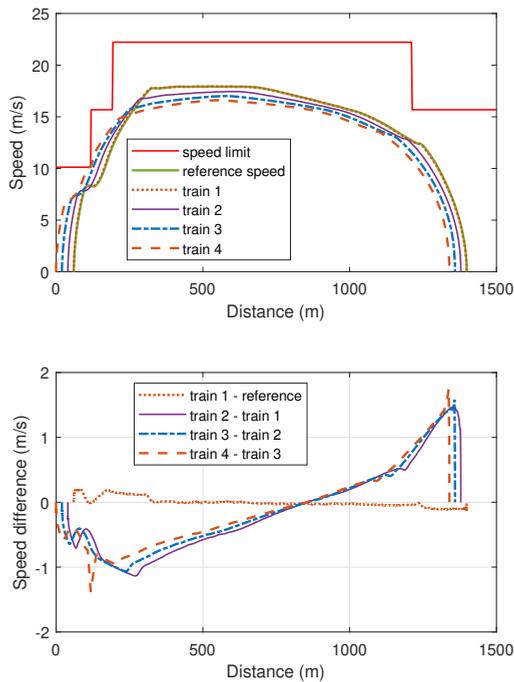


Figure 6.7: Speed profiles and speed difference of C-DMPC (with the same mass).

6.5.3. CONTROL PERFORMANCE WITH HETEROGENEOUS TRAIN MASS

In general, the masses of trains within a platoon are different due to variations in the total passenger loads on each train. The mass of a train influences the acceleration and deceleration (see (6.5a)), and determines the upper bound and the lower bound of the control input. Therefore, the mass inconsistency will influence the control performance of the platoon. In this case study, we consider a platoon of 4 trains, where the weights of the trains, from the leader train to the follower trains, are 60 t, 66 t, 57 t, and 66 t, respectively; so the heaviest train is more than 15% heavier than the lightest train.

In order to show the importance of incorporating the information on weights into the control design, we first conduct simulations with all trains assumed to have the same mass in the control design. Then, we compare the results with the true masses of the trains used in the control design. The simulation results are provided in Table 6.4.

Table 6.4: Simulation results for different approaches with different train masses

	Approach	RDP	Total cost	Relative distance (m)			Speed difference (m/s)			CPU time (s)	
				max	average	min	max	average	min	max	average
With all trains assumed to have the same mass	N-CDMPC	yes	$3.9359 \cdot 10^7$	44.26	16.34	-4.39	1.3943	0.0102	-1.1032	4.71	4.02
	C-CDMPC	yes	$3.9348 \cdot 10^7$	44.26	16.35	-4.39	1.3902	0.0087	-1.1049	1.78	1.41
	N-SDMPC	no	$3.9198 \cdot 10^7$	44.26	16.44	-4.40	1.5643	0.0030	-1.2313	0.55	0.26
	C-SDMPC	no	$3.9198 \cdot 10^7$	44.26	16.44	-4.40	1.5643	0.0030	-1.2313	0.30	0.22
	N-DMPC	no	$3.7318 \cdot 10^7$	46.31	18.35	-2.66	1.7237	0.0041	-1.3598	0.47	0.26
	C-DMPC	no	$3.7330 \cdot 10^7$	46.31	18.35	-2.67	1.7254	0.0041	-1.3747	0.27	0.22
With true masses of trains in control design	N-CDMPC	yes	$2.5846 \cdot 10^4$	21.96	19.86	17.86	2.2300	0.0126	-1.5705	5.04	4.35
	C-CDMPC	yes	$1.9643 \cdot 10^4$	21.92	19.88	18.16	2.2387	0.0106	-1.5715	1.82	1.39
	N-SDMPC	no	$3.0343 \cdot 10^4$	27.42	20.01	18.69	2.5732	0.0042	-1.6251	0.34	0.26
	C-SDMPC	no	$3.0342 \cdot 10^4$	27.42	20.01	18.69	2.5732	0.0042	-1.6251	0.31	0.23
	N-DMPC	no	$9.2421 \cdot 10^4$	31.33	21.87	18.18	2.7396	0.0057	-1.7608	0.35	0.27
	C-DMPC	no	$9.3093 \cdot 10^4$	31.47	21.88	18.17	2.7339	0.0057	-1.7748	0.30	0.23

From Table 6.4, we can find that all approaches have comparable performance in terms of the relative distance and speed difference with their convex counterpart under both cases. Therefore, in the following, we only use convex approaches for the comparison between the two cases in which the mass information is disregarded or included in the design.

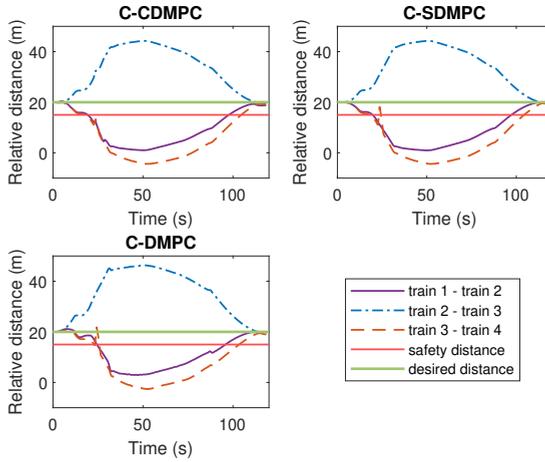


Figure 6.8: Relative distance of different approaches (with all trains assumed to have the same mass in the control design).

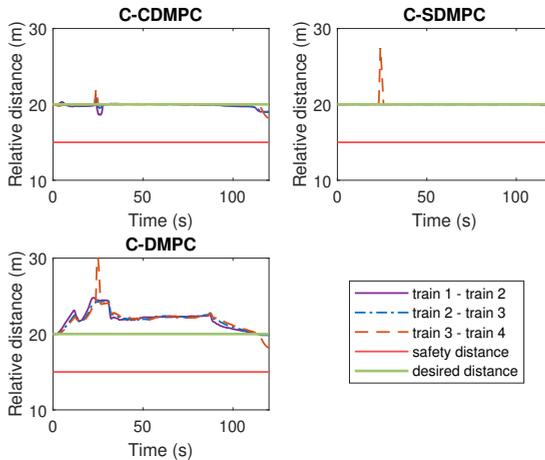


Figure 6.9: Relative distance of different approaches (with the true masses of the trains used in the control design).

In Table 6.4, if we assume all trains have the same mass in the control design, the minimum relative distance across all approaches is less than 0 m, implying the potential collision between trains, i.e., a train cannot ensure safety operation by using service braking when the predecessor train performs emergency braking. The relative distance between trains during the operation process of each approach when assuming all trains have the same mass in the control design is shown in Fig. 6.8. Fig. 6.8 shows that the relative distance between Train 1 and Train 2, Train 3 and Train 4 are lower than the given threshold. As the follower train has a larger inertia than its predecessor train, if the predecessor train starts to perform emergency braking, the follower train cannot perform braking with the same deceleration. Therefore, when a train is heavier than its predecessor train, the required safe tracking distance becomes difficult to ensure. Train 3 is lighter than Train 2; thus, the braking distance of Train 3 is shorter than expected and the safety distance between trains can be ensured. However, the relative distance between Train 2 and Train 3 is larger than the desired distance, with the maximum value being more than twice the desired distance, which is unnecessary and negatively influences the tracking performance.

The relative distance between trains of each approach when considering the true masses of the trains is shown in Fig. 6.9. From Table 6.4 and Fig. 6.9, we can find that by including train masses explicitly, the safety distance between trains can be ensured, and the relative distance between trains is comparable to the case of uniform masses in Table 6.3.

The speed profiles obtained by C-CDMPC, C-SDMPC, and C-DMPC considering the true masses of trains are provided in Fig. 6.10, Fig. 6.11, and Fig. 6.12, respectively. The C-CDMPC approach has the smallest fluctuation, with the relative distance fluctuating between [18.16 m, 21.92 m] and the speed difference fluctuates between [-1.5715 m/s, 2.2387 m/s]. In the cooperative control scheme, a subsystem can include the status of its neighbors and try to reach consistency with its neighbors regarding the relative distance and speed difference. Fig. 6.10, Fig. 6.11, and Fig. 6.12 show that for all three control methods, the speed difference between Train 2 and Train 3 is lower than the speed difference between Train 1 and Train 2, Train 3 and Train 4, implying that if the follower train is lighter than the predecessor train, the tracking performance would be better.

From the above simulations, we can conclude that the cooperative control approach has the best tracking performance while requiring ample communication and computation capabilities. Hence, C-CDMPC can be selected when sufficient communication bandwidth and computation power are available. The C-SDMPC approach can be selected in case of limited communication bandwidth and limited computation power. Moreover, in the worst case when two neighbor trains cannot communicate with each other, C-DMPC can be selected to control trains in a decentralized manner. Moreover, the simulation results also indicate that arranging heavier trains at the front of the platoon can help to improve the control performance of the virtually coupled trains.

6.5.4. HIGHLIGHTS OF RESULTS

CONVEX APPROXIMATION

In Sections 6.5.2 and 6.5.3, we have conducted simulations for cooperative distributed MPC, serial distributed MPC, and decentralized MPC under both the cases of uniform

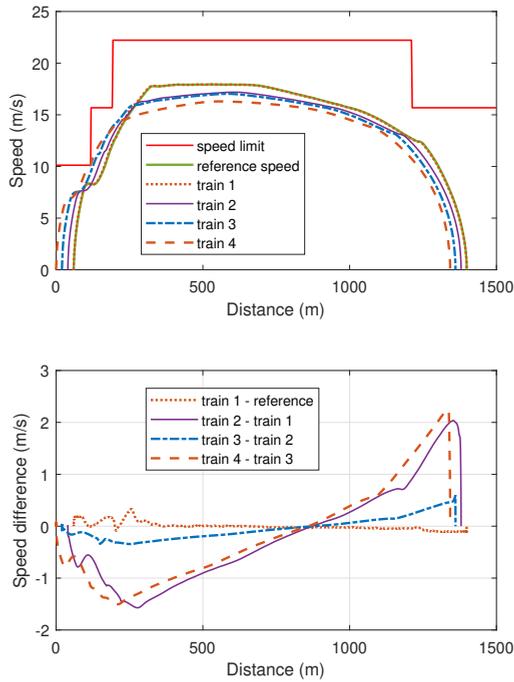


Figure 6.10: Speed profiles and speed difference of C-CDMPC (with the true masses of the trains used in the control design).

masses and heterogeneous masses. For all approaches and cases, we have tested non-linear MPC approaches and their convex approximations. The simulation results indicate that MPC with convex approximation can achieve a speed tracking accuracy that is comparable to that of the original nonconvex counterpart, while significantly reducing the computation time. Therefore, using convex approximation is an effective way to improve the computational efficiency of MPC in virtually coupled trains.

RELAXED DYNAMIC PROGRAMMING (RDP)

Sections 6.5.2 and 6.5.3 provide case studies for uniform masses and heterogeneous masses, respectively. The simulations indicate that cooperative distributed MPC, when accompanied by RDP, can achieve better performance with lower speed and distance tracking differences. By using RDP, we can develop a stopping criterion for the string stability of the platoon, which, in general, cannot be achieved with serial distributed MPC and decentralized MPC. Overall, RDP is an effective approach to analyze the stability of MPC approaches. Moreover, sufficient computational capacity should be ensured to support the efficient implementation of the RDP-based stopping criterion developed in this chapter.

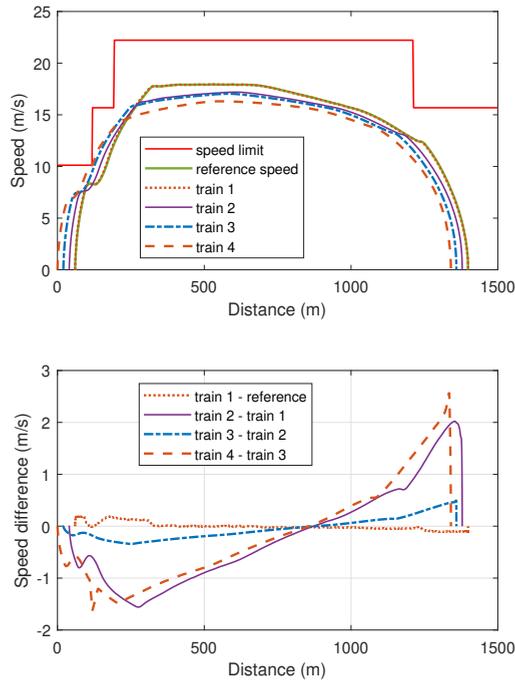


Figure 6.11: Speed profiles and speed difference of C-SDMPC (with the true masses of the trains used in the control design).

HETEROGENEOUS MASSES

Train masses influence the dynamics of trains and should be considered explicitly in the controller design to improve control performance. In Section 6.5.3, we have conducted simulations for cases with and without true masses of trains. The simulation results indicate that incorporating the true masses of trains in the controller design ensures safety and achieves the desired tracking performance while significantly reducing the total costs for all the mentioned MPC approaches. In this context, we conclude that, in general, detailed train information should be included to improve control performance when designing control approaches for virtually coupled heterogeneous trains.

Table 6.5: Characteristics of including different elements

Elements	Characteristics
Convex approximation	Reduce computational burden while maintaining tracking accuracy
RDP	Incorporate a stopping criterion into cooperative distributed MPC for string stability
Heterogeneous masses	Improve tracking accuracy and ensure safety

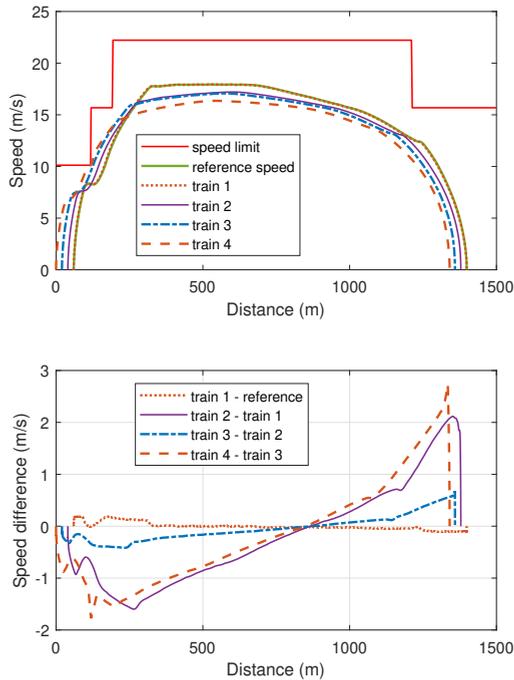


Figure 6.12: Speed profiles and speed difference of C-DMPC (with the true masses of the trains used in the control design).

To summarize, the advantages of considering convex approximation, relaxed dynamic programming, and heterogeneous masses are listed in Table 6.5. In this chapter, we consider communication between two consecutive trains (as stated in Fig. 6.1). In this context, each train only needs to consider the status of its preceding and succeeding trains when calculating its control decision. Thus, the approaches can be extended to larger train platoons without increasing the computational burden for each individual train.

6.6. CONCLUSIONS

In this chapter, cooperative distributed MPC, serial distributed MPC, and decentralized MPC have been compared and assessed for controlling virtually coupled trains, considering the nonlinear train model and changes in the masses of trains. We introduced the relaxed dynamic programming approach into the train control field, and a distributed stopping criterion with a stability guarantee has been developed for the cooperative distributed MPC approach. We have also proposed and assessed convex approximations of the above control approaches to make a balanced trade-off between computational

burden and accuracy. The three control approaches and their convex counterparts have been evaluated considering their distance tracking error, speed tracking error, and CPU time. Simulation results indicate that: 1) the convex approaches can achieve a performance that is comparable to that of their original nonconvex version, while the computational burden is reduced; 2) the cooperative control approach has the best tracking performance while requiring ample communication and computation capabilities; 3) by considering heterogeneous train masses explicitly, the safety distance between trains and the desired tracking performance can be ensured while the total objective function value is significantly reduced.

Future research could explore uncertainties related to resistances and train dynamics to enhance the performance of the control methods. Additionally, distributed control under conditions of intermittent communication is also promising, which can be achieved by designing appropriate self-triggered or event-triggered control strategies to address communication latency. Furthermore, future work could involve extending the research into other types of rail transportation modes, such as freight and heavy haul trains.

7

CONCLUSIONS AND RECOMMENDATIONS

In this thesis, we have investigated traffic management for urban rail transit networks by developing several efficient MPC approaches, including bi-level MPC, distributed MPC, and learning-based MPC. In this final chapter, the main conclusions, the impact of this thesis, and the recommendations for future research are provided.

7.1. CONCLUSIONS

This thesis answers the main research question (*How can model predictive control benefit flexible, highly efficient, and passenger-oriented urban rail transit network operations?*) by proposing a macroscopic passenger absorption model and a microscope passenger-oriented train scheduling model, and by developing several control approaches (including bi-level MPC, distributed MPC, mixed-integer linear programming (MILP) based MPC, and learning-based MPC) for efficient train scheduling in urban rail transit networks. In addition, a cooperative distributed MPC approach is developed for efficient operations of virtually coupled trains considering heterogeneous passenger loads. The results highlight the benefits of MPC in train scheduling and train operation problems, as it provides practical and innovative solutions for flexible, highly efficient, and passenger-oriented urban rail transit network operations.

The main results of this thesis can be summarized as follows:

- **Efficient model and control approach for the integration of timetables, passenger flows, and train speeds**

In Chapter 2, we have developed a bi-level framework for the integration of passenger flows, timetables, and train speeds. To deal with time-dependent passenger OD demands in urban rail transit networks, a passenger absorption model is developed to optimize train departure frequencies and rolling stock circulation plans. The passenger absorption model has been a macroscopic model that determines the maximum transport capacity of each line while making a balanced

trade-off between model accuracy and computational efficiency. To implement the generated departure frequencies, a train schedule model has been developed at the lower level considering detailed timetables, detailed rolling stock circulation, train speed profiles, and train orders. A bi-level MPC approach has been proposed for the bi-level framework, and the MPC optimization problems in both levels have been transformed into small-scale MILP problems, which enables solving them with existing MILP solvers. In this way, efficient real-time scheduling can be achieved for urban rail transit networks. (This addresses subquestion 1, i.e., integration of passenger flows, timetables, and train speeds, and subquestion 2, i.e., application of MPC.)

- **Scenario-based distributed MPC for optimizing train departure frequency**

In Chapter 3, we have developed a scenario-based distributed MPC approach for optimizing the train departure frequency of urban rail transit networks with uncertain passenger flows. To handle the computational complexity and communication restrictions in practical urban rail transit networks, a distributed-knowledgeable-reduced horizon algorithm has been proposed by considering different lines as different subsystems. Furthermore, to deal with uncertain passenger flows, we have implemented a scenario-based distributed control scheme and developed a scenario-based distributed-knowledgeable-reduced-horizon algorithm is developed. The developed algorithm can handle the computational complexity issues arising from large-scale networks. By incorporating a scenario-based distributed control scheme, the control performance of the proposed approach can be improved when considering uncertain passenger flows. (This addresses subquestion 3, i.e., train scheduling in large-scale networks.)

- **Learning-based MPC for timetable and train composition adjustment**

In Chapter 4, we have proposed a passenger flow model for passenger-oriented timetable scheduling. We have proposed a centralized MPC framework for real-time timetable scheduling. The MPC optimization problem has been transformed into a mixed-integer linear programming problem, which can be solved efficiently by existing MILP solvers. To further reduce the online computational burden, in Chapter 5, we have developed a learning-based MPC approach for timetable rescheduling problems. In the developed learning-based MPC approach, the integer variables of the resulting mixed-integer programming problem are obtained by deep learning, and the continuous variables are then solved by optimization. By applying a long short-term memory network to train the agent, the dynamic interdependencies within train schedules are captured to ensure effective adaptation and learning in response to evolving temporal dynamics. The developed approach can significantly reduce the computation time while achieving comparable performance to results obtained by the exact optimization-based MPC approach. (This addresses subquestion 4, i.e., train scheduling with time-varying passenger demands, and subquestion 5, i.e., efficient online computation of MPC.)

- **Cooperative distributed MPC for virtually coupled trains**

In Chapter 6, we have introduced the relaxed dynamic programming approach

into the train control field, and we have developed a distributed stopping criterion with a stability guarantee for the cooperative distributed MPC approach. In this context, the string stability of a train platoon can be ensured without using the MPC terminal conditions. To investigate the effect of train masses on the dynamics and control of the virtually coupled trains, we have explicitly included the changes in train masses when designing distributed MPC approaches. Considering typical communication topologies, we have compared and assessed the performance of cooperative distributed MPC, serial distributed MPC, and decentralized MPC for the control problem of virtually coupled trains. The simulation results indicate that the cooperative control approach exhibits the best tracking performance while requiring ample communication and computation capabilities. (This addresses subquestion 6, i.e., control of virtually coupled heterogeneous trains.)

7.2. IMPACTS OF THIS THESIS

7.2.1. SOCIAL IMPACTS

The thesis contributes to several positive social impacts, including the following:

- **Enhanced passenger convenience**

The primary objective of urban rail transit networks is to provide satisfactory services to passengers. The developed passenger-oriented traffic management approaches in Chapters 2-4 take into account the total passenger travel time while Chapter 5 considers the total passenger waiting time in urban rail transit networks. These approaches are instrumental in enhancing passenger satisfaction and ensuring the service quality of urban rail transit networks.

- **Improved efficiency of urban rail transit networks**

The current traffic management of urban rail transit networks suffers from a high computational burden due to the continuously expanding network scales and increasing passenger demands. This thesis has developed efficient traffic management and train control approaches to enhance the system-wide efficiency of urban rail transit networks from several aspects. In particular, Chapters 3-5 benefit efficient decision-making by developing efficient traffic management approaches that can encompass large-scale networks and time-dependent passenger demands. Furthermore, Chapter 6 provides several train speed approaches that can help to achieve efficient control for virtually coupled trains. By leveraging these approaches, significant contributions can be made toward efficient decision making for urban rail transit networks.

- **Sustainable urban mobility**

The approaches developed in Chapters 2, 3, and 5 explicitly incorporate the energy consumption of trains into the traffic management problem. The approaches enable the scheduling of fewer trains when passenger demands decrease, thereby facilitating energy-efficient traffic management. As a result, this thesis contributes to the sustainable mobility of urban rail transit networks and thus can help to achieve climate objectives.

7.2.2. SCIENTIFIC AND TECHNICAL IMPACTS

This thesis also makes contributions to advancing the state-of-the-art MPC methodologies that have been verified to be effective for urban rail transit networks.

- **Distributed MPC frameworks with reduced computation costs**

The implementation of MPC approaches is typically restricted by communication and system scaling issues. Shortening the prediction horizon can reduce the computational burden; however, a short prediction horizon may negatively affect the performance of the controller as less future information can be included in the decision-making process. The approach developed in Chapter 3 extends the existing approaches of designing cost-to-go functions. To deal with large-scale networks, solving the problem in a distributed manner is a natural choice. The distributed optimization approaches have a significant impact on distributed MPC. The research in Chapter 6 extends the direction of distributed MPC in reducing iterations while applying distributed optimization. By investigating the relation between distributed optimization and distributed control, significant improvement can be made in distributed MPC in terms of computational efficiency and control performance.

- **Learning-based MPC frameworks for mixed logical dynamical systems**

As an optimization-based approach, MPC can ensure safe control by explicitly including hard constraints. However, solving the optimization problem online for mixed logical dynamical systems may not be computationally affordable. Reinforcement learning and deep learning can provide efficient control decisions through a well-trained agent; however, ensuring safety in learning-based approaches is still challenging. The learning-based MPC approach developed in Chapter 5 leverages deep learning to improve solution efficiency while using MPC to ensure constraint satisfaction. The developed learning-based MPC approach expands the direction of control mixed logical dynamical systems where a trade-off between the control performance and real-time realizability is necessary.

- **Extension to other applications**

The methodologies developed in this thesis can be applied to the management and control problems of various other networks, including road traffic networks and power systems. The bi-level MPC approach of Chapter 2 extends the directions of controlling large-scale systems by reducing the computational burden by dividing the problem into macroscopic and microscopic levels. The developed distributed MPC approaches in Chapters 3 and 6 divide the overall problem into several subproblems according to the physical structure of the system and the research extends the direction of application of distributed control. The learning-based MPC approach of Chapter 5 extends the direction of improving the control efficiency of hybrid systems by separating discrete and continuous variables. The above approaches can be introduced to various application fields, and significant improvements can be made towards realizing more reliable and efficient control.

7.3. RECOMMENDATIONS FOR FUTURE RESEARCH

In this section, we outline some potential future research directions based on the findings of this thesis, which can be categorized into application topics and theory topics.

7.3.1. RECOMMENDATIONS FOR APPLICATION TOPICS

- **Passenger route choices**

In this thesis, the time-dependent passenger origin-destination demands are included in the traffic management problem of urban rail transit networks. However, when timetables are changed, passengers may also change the route for their travel. In this context, the dynamic interactions between departure frequencies and passenger route choices still ask for further research.

- **Disturbances and disruptions**

The real-time operations of the urban rail transit networks are inevitably influenced by many unexpected factors, such as bad weather, infrastructure failures, and equipment failures. These interferences may lead to serious delays and must be prevented where possible; otherwise, their effects should be minimized. These problems can be divided into disturbances and disruptions. Disturbances are defined as short delays caused by perturbations while disruptions refer to long delays that can lead to large decreases in network capacity. Extending the developed approaches to tackle disturbances and disruptions is also an imperative research direction.

- **Heterogeneous train platoons**

Safety is crucial for virtually coupled trains when reducing spacing between consecutive trains. The trains in the platoon typically have different conditions, including the variation of passenger load and difference of train types, yielding a heterogeneous train platoon. Developing efficient approaches to control heterogeneous train platoons can further improve the effectiveness and safety of the virtually coupled trains. Furthermore, including traction and braking saturation can also help to enhance the practical applicability of the corresponding approaches.

7.3.2. RECOMMENDATIONS FOR THEORY TOPICS

- **Distributed MPC under inexact distributed optimization**

Distributed optimization is typically applied to control of multi-agent systems; however, due to communication and computational power limitations, many real-time applications of distributed optimization terminate their iterations before reaching the optimal solution, thereby resulting in inexact minimization. Such early termination can result in constraint violation issues, as some distributed optimization approaches (e.g., ADMM) cannot ensure constraint satisfaction during iterations. Extending the distributed MPC approaches to deal with constraint violation issues with early termination by using tightened constraints [1; 41] and control barrier functions [123] can be a promising topic.

- **Multi-agent learning-based MPC**

Chapter 5 in this thesis focuses on centralized learning-based MPC, where a single

learning agent is incorporated into the centralized MPC scheme. The developed approach is limited to relatively small-scale cases (e.g., a bi-directional urban rail transit line considered in Chapter 5). A natural extension is to extend the work to multi-agent systems, so as to deal with the case of large-scale networks, where the centralized approach may encounter scaling and communication issues. To develop such multi-agent learning-based MPC approaches, the interaction between neighbor agents should be integrated, and the interaction can be investigated as cooperative or noncooperative games [37; 83; 101]. Furthermore, the safety guarantee of each agent should be considered when developing the multi-agent learning-based MPC approach.

- **Theoretical analysis of learning-based control**

Learning-based control approaches have shown promising performance in many application fields. In general, the agent should be well-trained before applying it for real-time control. However, there is no commonly accepted rule for the training process, and the optimal values of the hyperparameters may vary from case to case, thereby significantly reducing the applicability of learning-based approaches in different cases. Theoretical analysis of learning-based approaches is essential for understanding their underlying principles, strengths, and limitations in diverse application domains, which will improve the applicability of learning-based approaches in various cases. To improve the applicability, the transfer learning approaches, e.g., inductive transfer learning [144] and transductive transfer learning [4; 95], can be incorporated to design the learning-based control approaches.

BIBLIOGRAPHY

- [1] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE, 2019.
- [2] J. Aoun, E. Quaglietta, and R. M. Goverde. Investigating market potentials and operational scenarios of virtual coupling railway signaling. *Transportation Research Record*, 2674(8):799–812, 2020.
- [3] J. Aoun, E. Quaglietta, and R. M. Goverde. Roadmap development for the deployment of virtual coupling in railway signalling. *Technological Forecasting and Social Change*, 189:122263, 2023.
- [4] A. Arnold, R. Nallapati, and W. W. Cohen. A comparative study of methods for transductive transfer learning. In *Seventh IEEE international conference on data mining workshops (ICDMW 2007)*, pages 77–82. IEEE, 2007.
- [5] W. O. Assis and B. E. Milani. Generation of optimal schedules for metro lines using model predictive control. *Automatica*, 40(8):1397–1404, 2004.
- [6] G. Basile, D. G. Lui, A. Petrillo, and S. Santini. Deep deterministic policy gradient virtual coupling control for the coordination and manoeuvring of heterogeneous uncertain nonlinear high-speed trains. *Engineering Applications of Artificial Intelligence*, 133:108120, 2024.
- [7] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.
- [8] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.
- [9] N. Bešinović, Y. Wang, S. Zhu, E. Quaglietta, T. Tang, and R. M. Goverde. A matheuristic for the integrated disruption management of traffic, passengers and stations in urban railway lines. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):10380–10394, 2022.
- [10] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.
- [11] R. M. Borges and E. Quaglietta. Assessing hyperloop transport capacity under moving-block and virtual coupling operations. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):12612–12621, 2021.

- [12] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- [13] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37, 2014.
- [14] V. Cacchiani, J. Qi, and L. Yang. Robust optimization models for integrated train stop planning and timetabling with passenger demand uncertainty. *Transportation Research Part B: Methodological*, 136:1–29, 2020.
- [15] G. Caimi, M. Fuchsberger, M. Laumanns, and M. Lüthi. A model predictive control approach for discrete-time rescheduling in complex central railway station areas. *Computers & Operations Research*, 39(11):2578–2593, 2012.
- [16] G. C. Calafiore and M. C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.
- [17] M. C. Campi, S. Garatti, and F. A. Ramponi. A general scenario theory for nonconvex optimization and decision making. *IEEE Transactions on Automatic Control*, 63(12):4067–4078, 2018.
- [18] D. Canca, E. Barrena, E. Algaba, and A. Zarzo. Design and analysis of demand-adapted railway timetables. *Journal of Advanced Transportation*, 48(2):119–137, 2014.
- [19] D. Canca, E. Barrena, A. De-Los-Santos, and J. L. Andrade-Pineda. Setting lines frequency and capacity in dense railway rapid transit networks with simultaneous passenger assignment. *Transportation Research Part B: Methodological*, 93:251–267, 2016.
- [20] Y. Cao, J. Wen, and L. Ma. Tracking and collision avoidance of virtual coupling train control system. *Future Generation Computer Systems*, 120:76–90, 2021.
- [21] A. Cauligi, A. Chakrabarty, S. Di Cairano, and R. Quirynen. PRISM: Recurrent neural networks and presolve methods for fast mixed-integer optimal control. In *Learning for Dynamics and Control Conference*, pages 34–46. PMLR, 2022.
- [22] G. Cavone, T. van den Boom, L. Blenkers, M. Dotoli, C. Seatzu, and B. De Schutter. An MPC-based rescheduling algorithm for disruptions and disturbances in large-scale railway networks. *IEEE Transactions on Automation Science and Engineering*, 19(1):99–112, 2022.
- [23] F. Corman. Interactions and equilibrium between rescheduling train traffic and routing passengers in microscopic delay management: A game theoretical study. *Transportation Science*, 54(3):785–822, 2020.

- [24] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological*, 44(1):175–192, 2010.
- [25] F. Corman and L. Meng. A review of online dynamic models and algorithms for railway traffic management. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1274–1284, 2014.
- [26] F. Corman and E. Quaglietta. Closing the loop in real-time railway control: Framework design and impacts on operations. *Transportation Research Part C: Emerging Technologies*, 54:15–39, 2015.
- [27] J. Cury, F. Gomide, and M. Mendes. A methodology for generation of optimal schedules for an underground railway system. *IEEE Transactions on Automatic Control*, 25(2):217–222, 1980.
- [28] A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2):643–657, 2007.
- [29] W. J. Davis. *The tractive resistance of electric locomotives and cars*. General Electric, 1926.
- [30] A. De-Los-Santos, G. Laporte, J. A. Mesa, and F. Perea. Simultaneous frequency and capacity setting in uncapacitated metro lines in presence of a competing mode. *Transportation Research Procedia*, 3:289–298, 2014.
- [31] B. De Schutter, T. Van den Boom, and A. Hegyi. Model predictive control approach for recovery from delays in railway systems. *Transportation Research Record*, 1793(1):15–20, 2002.
- [32] C. Di Meo, M. Di Vaio, F. Flammini, R. Nardone, S. Santini, and V. Vittorini. Ertms/etcs virtual coupling: Proof of concept and numerical analysis. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2545–2556, 2019.
- [33] J. Felez, Y. Kim, and F. Borrelli. A model predictive control approach for virtual coupling in railways. *IEEE Transactions on Intelligent Transportation Systems*, 20(7):2728–2739, 2019.
- [34] J. Felez and M. A. Vaquero-Serrano. Virtual coupling in railways: A comprehensive review. *Machines*, 11(5):521, 2023.
- [35] J. Felez, M. A. Vaquero-Serrano, and J. de Dios Sanz. A robust model predictive control for virtual coupling in train sets. *Actuators*, 11(12):372, 2022.
- [36] S. Feng, Y. Zhang, S. E. Li, Z. Cao, H. X. Liu, and L. Li. String stability for vehicular platoon control: Definitions and analysis methods. *Annual Reviews in Control*, 47:81–97, 2019.

- [37] P. Frihauf, M. Krstic, and T. Basar. Nash equilibrium seeking in noncooperative games. *IEEE Transactions on Automatic Control*, 57(5):1192–1207, 2011.
- [38] J. Fu, A. Nunez, and B. De Schutter. Real-time UAV routing strategy for monitoring and inspection for postdisaster restoration of distribution networks. *IEEE Transactions on Industrial Informatics*, 18(4):2582–2592, 2021.
- [39] J. Fu, D. Sun, S. Peyghami, and F. Blaabjerg. A novel reinforcement-learning-based compensation strategy for dmpc-based day-ahead energy management of ship-board power systems. *IEEE Transactions on Smart Grid*, 2024.
- [40] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.
- [41] P. Giselsson and A. Rantzer. On feasibility, stability and performance in distributed model predictive control. *IEEE Transactions on Automatic Control*, 59(4):1031–1036, 2013.
- [42] K. Gkiotsalitis and O. Cats. Optimal frequency setting of metro services in the age of covid-19 distancing measures. *Transportmetrica A: Transport Science*, 18(3):807–827, 2022.
- [43] C. Gong, J. Shi, Y. Wang, H. Zhou, L. Yang, D. Chen, and H. Pan. Train timetabling with dynamic and random passenger demand: A stochastic optimization method. *Transportation Research Part C: Emerging Technologies*, 123:102963, 2021.
- [44] R. M. Goverde. Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B: Methodological*, 41(2):179–201, 2007.
- [45] N. Growe-Kuska, H. Heitsch, and W. Romisch. Scenario reduction and scenario tree construction for power management problems. In *2003 IEEE Bologna Power Tech Conference Proceedings*, volume 3, pages 1 – 7. IEEE, 2003.
- [46] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*. Springer, 2017.
- [47] J. T. Haahr, J. C. Wagenaar, L. P. Veelenturf, and L. G. Kroon. A comparison of two exact methods for passenger railway rolling stock (re) scheduling. *Transportation Research Part E: Logistics and Transportation Review*, 91:15–32, 2016.
- [48] D. Han and X. Yuan. A note on the alternating direction method of multipliers. *Journal of Optimization Theory and Applications*, 155:227–238, 2012.
- [49] A. Higgins and E. Kozan. Modeling train delays in urban networks. *Transportation Science*, 32(4):346–357, 1998.
- [50] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

- [51] Z. Hou, H. Dong, S. Gao, G. Nicholson, L. Chen, and C. Roberts. Energy-saving metro train timetable rescheduling model considering ato profiles and dynamic passenger flow. *IEEE Transactions on Intelligent Transportation Systems*, 20(7):2774–2785, 2019.
- [52] J. Jeschke, D. Sun, A. Jamshidnejad, and B. De Schutter. Grammatical-evolution-based parameterized model predictive control for urban traffic networks. *Control Engineering Practice*, 132:105431, 2023.
- [53] Z. Jiang, J. Gu, W. Fan, W. Liu, and B. Zhu. Q-learning approach to coordinated optimization of passenger inflow control with train skip-stopping on a urban rail transit line. *Computers & Industrial Engineering*, 127:1131–1142, 2019.
- [54] B. Kersbergen, J. Rudan, T. van den Boom, and B. De Schutter. Towards railway traffic management using switching max-plus-linear systems. *Discrete Event Dynamic Systems*, 26(2):183–223, 2016.
- [55] B. Kersbergen, T. van den Boom, and B. De Schutter. Distributed model predictive control for railway traffic management. *Transportation Research Part C: Emerging Technologies*, 68:462–489, 2016.
- [56] T. Keviczky, F. Borrelli, and G. J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, 2006.
- [57] H. Khadilkar. A scalable reinforcement learning algorithm for scheduling railway lines. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):727–736, 2019.
- [58] H. K. Khalil. *Nonlinear Systems (Third Edition)*. Patience Hall, 2002.
- [59] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [60] P. Kuppusamy, S. Venkatraman, C. Rishikeshan, and Y. P. Reddy. Deep learning based energy efficient optimal timetable rescheduling model for intelligent metro transportation systems. *Physical Communication*, 42:101131, 2020.
- [61] Y. Kuwata and J. P. How. Cooperative distributed robust trajectory optimization using receding horizon MILP. *IEEE Transactions on Control Systems Technology*, 19(2):423–431, 2010.
- [62] Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How. Distributed robust receding horizon control for multivehicle guidance. *IEEE Transactions on Control Systems Technology*, 15(4):627–641, 2007.
- [63] J. M. Latorre, S. Cerisola, and A. Ramos. Clustering algorithms for scenario tree generation: Application to natural hydro inflows. *European Journal of Operational Research*, 181(3):1339–1353, 2007.

- [64] E. L. Lawler and D. E. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14(4):699–719, 1966.
- [65] F. Leurent, E. Chandakas, and A. Poulhès. A traffic assignment model for passenger transit on a capacitated network: Bi-layer framework, line sub-models and large-scale application. *Transportation Research Part C: Emerging Technologies*, 47:3–27, 2014.
- [66] C. Li, J. Ma, T. H. Luan, X. Zhou, and L. Xiong. An incentive-based optimizing strategy of service frequency for an urban rail transit system. *Transportation Research Part E: Logistics and Transportation Review*, 118:106–122, 2018.
- [67] Q. Li, Z. Chen, and X. Li. A review of connected and automated vehicle platoon merging and splitting operations. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):22790–22806, 2022.
- [68] S. Li, B. De Schutter, L. Yang, and Z. Gao. Robust model predictive control for train regulation in underground railway transportation. *IEEE Transactions on Control Systems Technology*, 24(3):1075–1083, 2016.
- [69] S. Li, M. M. Dessouky, L. Yang, and Z. Gao. Joint optimal train regulation and passenger flow control strategy for high-frequency metro lines. *Transportation Research Part B: Methodological*, 99:113–137, 2017.
- [70] S. Li, L. Yang, and Z. Gao. Distributed optimal control for multiple high-speed train movement: An alternating direction method of multipliers. *Automatica*, 112:108646, 2020.
- [71] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Fast model predictive control for urban road networks via milp. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):846–856, 2011.
- [72] S. Lin, B. De Schutter, Y. Xi, and H. Hellendoorn. Efficient network-wide model-based predictive control for urban traffic networks. *Transportation Research Part C: Emerging Technologies*, 24:122–140, 2012.
- [73] S. Liu, A. Sadowska, and B. De Schutter. A scenario-based distributed model predictive control approach for freeway networks. *Transportation Research Part C: Emerging Technologies*, 136:103261, 2022.
- [74] X. Liu, A. Dabiri, and B. De Schutter. Timetable scheduling for passenger-centric urban rail networks: Model predictive control based on a novel absorption model. In *2022 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1147–1152. IEEE, 2022.
- [75] X. Liu, A. Dabiri, and B. De Schutter. Scenario-based mpc for real-time passenger-centric timetable scheduling of urban rail transit networks. *IFAC-PapersOnLine*, 56(2):2347–2352, 2023.

- [76] X. Liu, A. Dabiri, Y. Wang, and B. De Schutter. Modeling and efficient passenger-oriented control for urban rail transit networks. *IEEE Transactions on Intelligent Transportation Systems*, 24(3):3325–3338, 2023.
- [77] X. Liu, A. Dabiri, Y. Wang, and B. De Schutter. Real-time train scheduling with uncertain passenger flows: A scenario-based distributed model predictive control approach. *IEEE Transactions on Intelligent Transportation Systems*, 25(5):4219–4232, 2024.
- [78] X. Liu, A. Dabiri, Y. Wang, J. Xun, and B. De Schutter. Distributed model predictive control for virtually coupled trains: Comparison and assessment. 2024. submitted to journal.
- [79] X. Liu, A. Dabiri, J. Xun, and B. De Schutter. Bi-level model predictive control for metro networks: Integration of timetables, passenger flows, and train speed profiles. *Transportation Research Part E: Logistics and Transportation Review*, 180:103339, 2023.
- [80] X. Liu, C. Fabio Oliveira da Silva, A. Dabiri, Y. Wang, and B. De Schutter. Learning-based model predictive control for passenger-oriented train rescheduling with flexible train composition. 2024. submitted to journal.
- [81] Y. Liu, R. Liu, C. Wei, J. Xun, and T. Tang. Distributed model predictive control strategy for constrained high-speed virtually coupled train set. *IEEE Transactions on Vehicular Technology*, 71(1):171–183, 2021.
- [82] Y. Liu, Y. Zhou, S. Su, J. Xun, and T. Tang. An analytical optimal control approach for virtually coupled high-speed trains with local and string stability. *Transportation Research Part C: Emerging Technologies*, 125:102886, 2021.
- [83] S. Lozano, P. Moreno, B. Adenso-Díaz, and E. Algaba. Cooperative game theory approach to allocating benefits of horizontal cooperation. *European Journal of Operational Research*, 229(2):444–452, 2013.
- [84] X. Luan and F. Corman. Passenger-oriented traffic control for rail networks: An optimization model considering crowding effects on passenger choices and train operations. *Transportation Research Part B: Methodological*, 158:239–272, 2022.
- [85] X. Luan, B. De Schutter, L. Meng, and F. Corman. Decomposition and distributed optimization of real-time traffic management for large-scale railway networks. *Transportation Research Part B: Methodological*, 141:72–97, 2020.
- [86] X. Luan, Y. Wang, B. De Schutter, L. Meng, G. Lodewijks, and F. Corman. Integration of real-time traffic management and train control for rail networks-part 1: Optimization problems and solution approaches. *Transportation Research Part B: Methodological*, 115:41–71, 2018.
- [87] X. Luan, Y. Wang, B. De Schutter, L. Meng, G. Lodewijks, and F. Corman. Integration of real-time traffic management and train control for rail networks-part

- 2: Extensions towards energy-efficient train operations. *Transportation Research Part B: Methodological*, 115:72–94, 2018.
- [88] X. Luo, T. Tang, J. Yin, and H. Liu. A robust MPC approach with controller tuning for close following operation of virtually coupled train set. *Transportation Research Part C: Emerging Technologies*, 151:104116, 2023.
- [89] M. Lüthi. *Improving the efficiency of heavily used railway networks through integrated real-time rescheduling*. PhD thesis, ETH Zurich, 2009.
- [90] J. M. Maestre, R. R. Negenborn, et al. *Distributed Model Predictive Control Made Easy*, volume 69. Springer, 2014.
- [91] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.
- [92] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- [93] P. Mo, L. Yang, A. D’Ariano, J. Yin, Y. Yao, and Z. Gao. Energy-efficient train scheduling and rolling stock circulation planning in a metro line: a linear programming approach. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3621–3633, 2020.
- [94] P. Mo, L. Yang, A. D’Ariano, J. Yin, Y. Yao, and Z. Gao. Energy-efficient train scheduling and rolling stock circulation planning in a metro line: a linear programming approach. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3621–3633, 2020.
- [95] A. Moreo, A. Esuli, and F. Sebastiani. Lost in transduction: Transductive transfer learning in text classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(1):1–21, 2021.
- [96] B. Ning, T. Tang, K. Qiu, and C. Gao. CBTC (communication based train control): System and development. *WIT Transactions on State-of-the-art in Science and Engineering*, 46, 2010.
- [97] H. Niu, X. Zhou, and R. Gao. Train scheduling for minimizing passenger waiting time with time-dependent demand and skip-stop patterns: Nonlinear integer programming models with linear constraints. *Transportation Research Part B: Methodological*, 76:117–135, 2015.
- [98] P. Noursalehi, H. N. Koutsopoulos, and J. Zhao. Dynamic origin-destination prediction in urban rail systems: A multi-resolution spatio-temporal deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5106–5115, 2021.
- [99] H. Pan, L. Yang, and Z. Liang. Demand-oriented integration optimization of train timetabling and rolling stock circulation planning with flexible train compositions: A column-generation-based approach. *European Journal of Operational Research*, 305(1):184–206, 2023.

- [100] J. Park, B.-H. Lee, and Y. Eun. Virtual coupling of railway vehicles: Gap reference for merge and separation, robust control, and position measurement. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):1085–1096, 2020.
- [101] B. Peleg and P. Sudhölter. *Introduction to the theory of cooperative games*, volume 34. Springer Science & Business Media, 2007.
- [102] S. Pu and S. Zhan. Two-stage robust railway line-planning approach with passenger demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 152:102372, 2021.
- [103] J. Qi, V. Cacchiani, L. Yang, C. Zhang, and Z. Di. An integer linear programming model for integrated train stop planning and timetabling with time-dependent passenger demand. *Computers & Operations Research*, 136:105484, 2021.
- [104] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.
- [105] E. Quaglietta, P. Spartalis, M. Wang, R. M. Goverde, and P. van Koningsbruggen. Modelling and analysis of virtual coupling with dynamic safety margin considering risk factors in railway operations. *Journal of Rail Transport Planning & Management*, 22:100313, 2022.
- [106] E. Quaglietta, M. Wang, and R. M. Goverde. A multi-state train-following model for the analysis of virtual coupling railway operations. *Journal of Rail Transport Planning & Management*, 15:100195, 2020.
- [107] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.
- [108] L. Russo, S. H. Nair, L. Glielmo, and F. Borrelli. Learning for online mixed-integer model predictive control with parametric optimality certificates. *IEEE Control Systems Letters*, 7:2215–2220, 2023.
- [109] R. Sattiraju, J. Siemons, M. Soliman, W. Alshrafi, F. Rein, and H. D. Schotten. Design of a highly reliable wireless module for ultra-low-latency short range applications. In *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, pages 1–4. IEEE, 2017.
- [110] D. Šemrov, R. Marsetič, M. Žura, L. Todorovski, and A. Srdic. Reinforcement learning approach for train rescheduling on a single-track railway. *Transportation Research Part B: Methodological*, 86:250–267, 2016.
- [111] J. Shi, T. Qin, L. Yang, X. Xiao, J. Guo, Y. Shen, and H. Zhou. Flexible train capacity allocation for an overcrowded metro line: A new passenger flow control approach. *Transportation Research Part C: Emerging Technologies*, 140:103676, 2022.

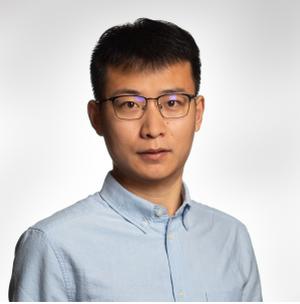
- [112] J. Shi, L. Yang, J. Yang, and Z. Gao. Service-oriented train timetabling with collaborative passenger flow control on an oversaturated metro line: An integer linear optimization approach. *Transportation Research Part B: Methodological*, 110:26–59, 2018.
- [113] M. Soliman, J. Siemons, J. Kochems, W. Alshrafi, J. Shamshoom, D. Heberling, and A. Dekorsy. Automatic train coupling: Challenges and key enablers. *IEEE Communications Magazine*, 57(9):32–38, 2019.
- [114] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [115] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460–469, 2010.
- [116] B. T. Stewart, S. J. Wright, and J. B. Rawlings. Cooperative distributed model predictive control for nonlinear systems. *Journal of Process Control*, 21(5):698–704, 2011.
- [117] S. Su, X. Li, T. Tang, and Z. Gao. A subway train timetable optimization approach based on energy-efficient operation strategy. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):883–893, 2013.
- [118] S. Su, J. She, K. Li, X. Wang, and Y. Zhou. A nonlinear safety equilibrium spacing-based model predictive control for virtually coupled train set over gradient terrains. *IEEE Transactions on Transportation Electrification*, 8(2):2810–2824, 2021.
- [119] S. Su, J. She, D. Wang, S. Gong, and Y. Zhou. A stabilized virtual coupling scheme for a train set with heterogeneous braking dynamics capability. *Transportation Research Part C: Emerging Technologies*, 146:103947, 2023.
- [120] D. Sun, A. Jamshidnejad, and B. De Schutter. A novel framework combining mpc and deep reinforcement learning with application to freeway traffic control. *IEEE Transactions on Intelligent Transportation Systems*, 2024.
- [121] X. Sun, Z. Yao, C. Dong, and D. Clarke. Optimal control strategies for metro trains to use the regenerative braking energy: A speed profile adjustment approach. *IEEE Transactions on Intelligent Transportation Systems*, 24(6):5883–5894, 2023.
- [122] D. Swaroop and J. K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, 1996.
- [123] X. Tan and D. V. Dimarogonas. Distributed implementation of control barrier functions for multi-agent systems. *IEEE Control Systems Letters*, 6:1879–1884, 2022.
- [124] R. Tang, L. De Donato, N. Besinović, F. Flammini, R. M. Goverde, Z. Lin, R. Liu, T. Tang, V. Vittorini, and Z. Wang. A literature review of artificial intelligence applications in railway systems. *Transportation Research Part C: Emerging Technologies*, 140:103679, 2022.

- [125] T. J. van den Boom and B. De Schutter. Modelling and control of discrete event systems using switching max-plus-linear systems. *Control Engineering Practice*, 14(10):1199–1211, 2006.
- [126] M. A. Vaquero-Serrano and J. Felez. A decentralized robust control approach for virtually coupled train sets. *Computer-Aided Civil and Infrastructure Engineering*, 38(14):1896–1915, 2023.
- [127] H. Wang, Q. Zhao, S. Lin, D. Cui, C. Luo, L. Zhu, X. Wang, and T. Tang. A reinforcement learning empowered cooperative control approach for IIoT-based virtually coupled train sets. *IEEE Transactions on Industrial Informatics*, 17(7):4935–4945, 2020.
- [128] P. Wang, L. Ma, R. M. Goverde, and Q. Wang. Rescheduling trains using petri nets and heuristic search. *IEEE Transactions on Intelligent Transportation Systems*, 17(3):726–735, 2016.
- [129] P. Wang, X. Xiao, and N. Nießen. Flexible rolling stock composition strategy in urban rail transit lines: The influences of various train units and station capacities. *Transportation Research Part C: Emerging Technologies*, 162:104609, 2024.
- [130] X. Wang, S. Li, T. Tang, and L. Yang. Event-triggered predictive control for automatic train regulation and passenger flow in metro rail systems. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):1782–1795, 2022.
- [131] X. Wang, L. Liu, T. Tang, and W. Sun. Enhancing communication-based train control systems through train-to-train communications. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1544–1561, 2019.
- [132] Y. Wang. *Optimal trajectory planning and train scheduling for railway systems*. PhD thesis, Delft University of Technology, 2014.
- [133] Y. Wang, A. D’Ariano, J. Yin, L. Meng, T. Tang, and B. Ning. Passenger demand oriented train scheduling and rolling stock circulation planning for an urban rail transit line. *Transportation Research Part B: Methodological*, 118:193–227, 2018.
- [134] Y. Wang, B. De Schutter, T. J. van den Boom, B. Ning, and T. Tang. Efficient bilevel approach for urban rail transit operation with stop-skipping. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2658–2670, 2014.
- [135] Y. Wang, B. Ning, T. Tang, T. J. van den Boom, and B. De Schutter. Efficient real-time train scheduling for urban rail transit systems using iterative convex programming. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3337–3352, 2015.
- [136] Y. Wang, T. Tang, B. Ning, T. J. van den Boom, and B. De Schutter. Passenger-demands-oriented train scheduling for an urban rail transit network. *Transportation Research Part C: Emerging Technologies*, 60:1–23, 2015.

- [137] Y. Wang, S. Zhu, A. D'Ariano, J. Yin, J. Miao, and L. Meng. Energy-efficient timetabling and rolling stock circulation planning based on automatic train operation levels for metro lines. *Transportation Research Part C: Emerging Technologies*, 129:103209, 2021.
- [138] Y. Wang, S. Zhu, S. Li, L. Yang, and B. De Schutter. Hierarchical model predictive control for on-line high-speed railway delay management and train control in a dynamic operations environment. *IEEE Transactions on Control Systems Technology*, 30(6):2344–2359, 2022.
- [139] H. P. Williams. *Model Building in Mathematical Programming*. John Wiley & Sons, 2013.
- [140] Q. Wu, X. Ge, Q.-L. Han, and Y. Liu. Railway virtual coupling: A survey of emerging control techniques. *IEEE Transactions on Intelligent Vehicles*, 8(5):3239–3255, 2023.
- [141] X. Wu, H. Dong, and K. T. Chi. Multi-objective timetabling optimization for a two-way metro line under dynamic passenger demand. *IEEE Transactions on Intelligent Transportation Systems*, 22(8):4853–4863, 2021.
- [142] X. Wu, M. Yang, W. Lian, M. Zhou, H. Wang, and H. Dong. Cascading delays for the high-speed rail network under different emergencies: A double layer network approach. *IEEE/CAA Journal of Automatica Sinica*, 10(10):2014–2025, 2023.
- [143] P. Xu, F. Corman, Q. Peng, and X. Luan. A timetable rescheduling approach and transition phases for high-speed railway traffic during disruptions. *Transportation Research Record*, 2607(1):82–92, 2017.
- [144] L. Xuhong, Y. Grandvalet, and F. Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pages 2825–2834. PMLR, 2018.
- [145] J. Xun, M. Chen, Y. Liu, and F. Liu. An overspeed protection mechanism for virtual coupling in railway. *IEEE Access*, 8:187400–187410, 2020.
- [146] J. Xun, Y. Li, R. Liu, Y. Li, and Y. Liu. A survey on control methods for virtual coupling in railway operation. *IEEE Open Journal of Intelligent Transportation Systems*, 3:838–855, 2022.
- [147] L. Yang, Y. Gao, A. D'Ariano, and S. Xu. Integrated optimization of train timetable and train unit circulation for a y-type urban rail transit system with flexible train composition mode. *Omega*, 122:102968, 2024.
- [148] L. Yang, Y. Zhang, S. Li, and Y. Gao. A two-stage stochastic optimization model for the transfer activity choice in metro networks. *Transportation Research Part B: Methodological*, 83:271–297, 2016.
- [149] J. Y. Yen. Finding the shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.

- [150] J. Yin, A. D'Ariano, Y. Wang, L. Yang, and T. Tang. Timetable coordination in a rail transit network with time-dependent passenger demand. *European Journal of Operational Research*, 295(1):183–202, 2021.
- [151] J. Yin, T. Tang, L. Yang, Z. Gao, and B. Ran. Energy-efficient metro train rescheduling with uncertain time-variant passenger demands: An approximate dynamic programming approach. *Transportation Research Part B: Methodological*, 91:178–210, 2016.
- [152] J. Yin, L. Yang, T. Tang, Z. Gao, and B. Ran. Dynamic passenger demand oriented metro train scheduling with energy-efficiency and waiting time minimization: Mixed-integer linear programming approaches. *Transportation Research Part B: Methodological*, 97:182–213, 2017.
- [153] C.-S. Ying, A. H. Chow, Y.-H. Wang, and K.-S. Chin. Adaptive metro service schedule and train composition with a proximal policy optimization approach based on deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6895–6906, 2021.
- [154] S. Zhan, L. G. Kroon, L. P. Veelenturf, and J. C. Wagenaar. Real-time high-speed train rescheduling in case of a complete blockage. *Transportation Research Part B: Methodological*, 78:182–201, 2015.
- [155] Z. Zhang, H. Song, H. Wang, X. Wang, and H. Dong. Cooperative multi-scenario departure control for virtual coupling trains: A fixed-time approach. *IEEE Transactions on Vehicular Technology*, 70(9):8545–8555, 2021.
- [156] Y. Zhao, D. Li, Y. Yin, and X. Zhao. Integrated optimization of demand-driven timetable, train formation plan and rolling stock circulation with variable running times and dwell times. *Transportation Research Part E: Logistics and Transportation Review*, 171:103035, 2023.
- [157] H. Zhou, J. Qi, L. Yang, J. Shi, and P. Mo. Joint optimization of train scheduling and rolling stock circulation planning with passenger flow control on tidal overcrowded metro lines. *Transportation Research Part C: Emerging Technologies*, 140:103708, 2022.
- [158] Y. Zhu and R. M. Goverde. Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions. *Transportation Research Part B: Methodological*, 123:149–181, 2019.
- [159] Y. Zhu and R. M. Goverde. Integrated timetable rescheduling and passenger re-assignment during railway disruptions. *Transportation Research Part B: Methodological*, 140:282–314, 2020.

CURRICULUM VITAE



Xiaoyu Liu was born on November 4, 1995, in Linyi, Shandong Province, China. He obtained his B.Sc. and M.Sc. degrees from the School of Electronic and Information Engineering at Beijing Jiaotong University, Beijing, China, in 2017 and 2020, respectively.

In November 2020, he became a Ph.D. candidate at the Delft Center for Systems and Control, Faculty of Mechanical Engineering, Delft University of Technology, Delft, the Netherlands, under the supervision of Prof.dr.ir. Bart De Schutter and Dr. Azita Dabiri. From July to September 2024, he conducted a research visit to the Division of Decision and Control Systems,

School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden, hosted by Prof. Dimos V. Dimarogonas.

In his Ph.D. project, he worked on distributed and learning-based model predictive control for urban rail transit networks. His research interests include (distributed) model predictive control, control of hybrid systems, reinforcement learning, railway operations, and intelligent transportation systems.

LIST OF PUBLICATIONS

JOURNAL PAPERS

7. **X. Liu**, D. V. Dimarogonas, C. Liu, A. Dabiri, B. De Schutter. Distributed model predictive control with non-terminal costs under inexact distributed optimization, *in preparation for journal*.
6. **X. Liu**, B. Wentges, A. Dabiri, J. Zhao, B. De Schutter. Railway timetable rescheduling with passenger origin-destination demands and rolling stock circulation during disruptions, *to be submitted to journal*.
5. **X. Liu**, C.F. Oliveira da Silva, A. Dabiri, Y. Wang, B. De Schutter. Learning-based model predictive control for passenger-oriented train rescheduling with flexible train composition, *submitted to journal*. *X. Liu and C.F. Oliveira da Silva contributed equally*.
4. **X. Liu**, A. Dabiri, Y. Wang, J. Xun, B. De Schutter. Distributed model predictive control for virtually coupled heterogeneous trains: Comparison and assessment, *IEEE Transactions on Intelligent Transportation Systems*, 2024, doi: 10.1109/TITS.2024.3458169.
3. **X. Liu**, A. Dabiri, Y. Wang, B. De Schutter. Real-time train scheduling with uncertain passenger flows: A scenario-based distributed model predictive control approach, *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 4219-4232, 2024.
2. **X. Liu**, A. Dabiri, J. Xun, B. De Schutter. Bi-level model predictive control for metro networks: Integration of timetables, passenger flows, and train speed profiles, *Transportation Research Part E: Logistics and Transportation Review*, vol. 180, pp. 103339, 2023.
1. **X. Liu**, A. Dabiri, Y. Wang, B. De Schutter. Modeling and efficient passenger-oriented control for urban rail transit networks, *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3325-3338, 2023.

CONFERENCE PROCEEDINGS

5. H. Zhang, **X. Liu**, D. Sun, A. Dabiri, B. De Schutter. Integrated reinforcement learning and optimization for railway timetable rescheduling, *Proceedings of the 17th IFAC Symposium on Control in Transportation Systems (CTS 2024)*, Ayia Napa, Cyprus, July 2024.
4. A. Daman, **X. Liu**, A. Dabiri, B. De Schutter. Benders decomposition-based optimization of train departure frequencies in metro networks, *Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC 2023)*, Bilbao, Spain, pp. 5371-5376, September 2023.
3. R. Guo, **X. Liu**, S. Bhatnagar, M. Vallati. Real-time routing and scheduling of on-demand autonomous customized bus systems, *Proceedings of the 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC 2023)*, Bilbao, Spain, pp. 5371-5376, September 2023.

2. **X. Liu**, A. Dabiri, B. De Schutter. Scenario-based MPC for real-time passenger-centric timetable scheduling of urban rail transit networks, *Proceedings of the 2023 IFAC World Congress (IFAC 2023)*, Yokohama, Japan, pp. 2646-2651, July 2023.
1. **X. Liu**, A. Dabiri, B. De Schutter. Timetable scheduling for passenger-centric urban rail networks: model predictive control based on a novel absorption model, *Proceedings of the 2022 IEEE Conference on Control Technology and Applications (CCTA 2022)*, Trieste, Italy, pp. 1147-1152, August 2022.

TRAIL THESIS SERIES

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 400 titles, see the TRAIL website: www.rsTRAIL.nl. The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Liu, X., Distributed and Learning-based Model Predictive Control for Urban Rail Transit Networks, T2024/10, October 2024, Thesis Series, the Netherlands

Clercq, G. K. de, On the Mobility Effects of Future Transport Modes, T2024/9, October 2024, Thesis Series, the Netherlands

Dreischerf, A.J., From Caveats to Catalyst: Accelerating urban freight transport sustainability through public initiatives, T2024/8, September 2024, TRAIL Thesis Series, the Netherlands

Zohoori, B., Model-based Risk Analysis of Supply Chains for Supporting Resilience, T2024/7, October 2024, TRAIL Thesis Series, the Netherlands

Poelman, M.C., Predictive Traffic Signal Control under Uncertainty: Analyzing and Reducing the Impact of Prediction Errors, T2024/6, October 2024, TRAIL Thesis Series, the Netherlands

Berge, S.H., Cycling in the age of automation :Enhancing cyclist interaction with automated vehicles through human-machine interfaces, T2024/5, September 2024, TRAIL Thesis Series, the Netherlands

Wu, K., Decision-Making and Coordination in Green Supply Chains with Asymmetric Information, T2024/4, July 2024, TRAIL Thesis Series, the Netherlands

Wijnen, W., Road Safety and Welfare, T2024/3, May 2024, TRAIL Thesis Series, the Netherlands

Caiati, V., Understanding and Modelling Individual Preferences for Mobility as a Service, T2024/2, March 2024, TRAIL Thesis Series, the Netherlands

Vos, J., Drivers' Behaviour on Freeway Curve Approach, T2024/1, February 2024, TRAIL Thesis Series, the Netherlands

Geržinič, N., The Impact of Public Transport Disruptors on Travel Behaviour, T2023/20, December 2023, TRAIL Thesis Series, the Netherlands

Dubey, S., A Flexible Behavioral Framework to Model Mobilityon-Demand Service Choice Preference, T2023/19, November 2023, TRAIL Thesis Series, the Netherlands

Sharma, S., On-trip Behavior of Truck Drivers on Freeways: New mathematical models and control methods, T2023/18, October 2023, TRAIL Thesis Series, the Netherlands

- Ashkrof, P., Supply-side Behavioural Dynamics and Operations of Ride-sourcing Platforms, T2023/17, October 2023, TRAIL Thesis Series, the Netherlands
- Sun, D., Multi-level and Learning-based Model Predictive Control for Traffic Management, T2023/16, October 2023, TRAIL Thesis Series, the Netherlands
- Brederode, L.J.N., Incorporating Congestion Phenomena into Large Scale Strategic Transport Model Systems, T2023/15, October 2023, TRAIL Thesis Series, the Netherlands
- Hernandez, J.I., Data-driven Methods to study Individual Choice Behaviour: with applications to discrete choice experiments and Participatory Value Evaluation experiments, T2023/14, October 2023, TRAIL Thesis Series, the Netherlands
- Aoun, J., Impact Assessment of Train-Centric Rail Signaling Technologies, T2023/13, October 2023, TRAIL Thesis Series, the Netherlands
- Pot, F.J., The Extra Mile: Perceived accessibility in rural areas, T2023/12, September 2023, TRAIL Thesis Series, the Netherlands
- Nikghadam, S., Cooperation between Vessel Service Providers for Port Call Performance Improvement, T2023/11, July 2023, TRAIL Thesis Series, the Netherlands
- Li, M., Towards Closed-loop Maintenance Logistics for Offshore Wind Farms: Approaches for strategic and tactical decision-making, T2023/10, July 2023, TRAIL Thesis Series, the Netherlands
- Berg, T. van den, Moral Values, Behaviour, and the Self: An empirical and conceptual analysis, T2023/9, May 2023, TRAIL Thesis Series, the Netherlands
- Shelat, S., Route Choice Behaviour under Uncertainty in Public Transport Networks: Stated and revealed preference analyses, T2023/8, June 2023, TRAIL Thesis Series, the Netherlands
- Zhang, Y., Flexible, Dynamic, and Collaborative Synchromodal Transport Planning Considering Preferences, T2023/7, June 2023, TRAIL Thesis Series, the Netherlands
- Kapetanović, M., Improving Environmental Sustainability of Regional Railway Services, T2023/6, June 2023, TRAIL Thesis Series, the Netherlands
- Li, G., Uncertainty Quantification and Predictability Analysis for Traffic Forecasting at Multiple Scales, T2023/5, April 2023, TRAIL Thesis Series, the Netherlands
- Harter, C., Vulnerability through Vertical Collaboration in Transportation: A complex networks approach, T2023/4, March 2023, TRAIL Thesis Series, the Netherlands
- Razmi Rad, S., Design and Evaluation of Dedicated Lanes for Connected and Automated Vehicles, T2023/3, March 2023, TRAIL Thesis Series, the Netherlands
- Eikenbroek, O., Variations in Urban Traffic, T2023/2, February 2023, TRAIL Thesis Series, the Netherlands