Technical report 23-033

Modeling and efficient passenger-oriented control for urban rail transit networks[*]

X. Liu, A. Dabiri, Y. Wang, and B. De Schutter

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

[*] This report can also be downloaded via https://pub.bartdeschutter.org/abs/23_033.html

Modeling and Efficient Passenger-Oriented Control for Urban Rail Transit Networks

Xiaoyu Liu, Azita Dabiri, Yihui Wang, and Bart De Schutter, *Fellow, IEEE*

***Abstract*—Real-time timetable scheduling is an effective way to improve passenger satisfaction and to reduce operational costs in urban rail transit networks. In this paper, 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.**

***Index Terms*—Model predictive control, Urban rail transit, Real-time timetable scheduling, Time-dependent passenger origin-destination demand.**

I. 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 [1], [2], [3], 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 [4], train orders [5], conflict resolution [6], 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 paper, 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 [4], [7], speed profiles [8], and short-turning [9], but without detailed passenger origin-destination (OD) information. Another direction of studies addresses passenger OD demands on a single line [10], [11]. 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 [12] or urban rail networks [13], [14]; 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 [15], [16], [17]. 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 [18], [19] have inspired us to develop an efficient model for passenger-orient railway traffic networks and to develop efficient MPC methods for the real-time timetable scheduling problem.

The main contributions of the paper 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-

This work is supported by the National Natural Science Foundation of China under Grant 72071016, and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101018826 - CLariNet). The work of the first author is also supported by the China Scholarship Council under Grant 202007090003.

Xiaoyu Liu, Azita Dabiri, and Bart De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, 2628CD Delft, The Netherlands (email: x.liu-20@tudelft.nl, a.dabiri@tudelft.nl, b.deschutter@tudelft.nl).

Yihui Wang is with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (email: yihui.wang@bjtu.edu.cn).

dependent passenger origin-destination demands.

2) Thanks to the notion of cycle time introduced in this paper, 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 paper is structured as follows. Section II summarizes the literature related to this paper. In Section III, the passenger-oriented urban rail traffic model is proposed. In Section IV, the MPC controller is designed for the passenger-oriented timetable scheduling problem based on the proposed model. In Section V, the MPC optimization problem is solved with different methods, and an MILP-based approach is proposed. Section VI provides case studies to illustrate the accuracy of the model and the efficiency of the developed method. Finally, conclusions are given in Section VII.

II. State of The Art

A. 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 [2], 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 [20]. In [21], 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 [22], to optimize passengers' routes [9], and to integrate passenger reassignment and timetable scheduling [12]. Furthermore, max-plus models [23] and switching max-plus-linear models [1], [24] 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 [25], 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 [27], [28], 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 [5], 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.

B. 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 [29], 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 [30], 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 [10]. In [31], an MINLP problem was formulated to minimize passenger waiting time with consideration of time-varying passenger demands. In [11], 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 [32]; 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 [13], 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 [33]. In [34], 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 [14], 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.

C. *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 [1], 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 [24]. In [35], 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 [5]. 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 [36]. 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 paper 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.

III. 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 paper 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 paper 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 paper are introduced in Section III-A. Then, the simplified passenger flow model is proposed in Section III-B. In Section III-C, the train operation model related to the simplified passenger flow model is given.

A. *Notations*

Sets and Indices

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
$\mathrm{s}^{\mathrm{pla}}(p)$	Successor platform of platform p
$\mathrm{p}^{\mathrm{pla}}(p)$	Predecessor platform of platform p

Input Parameters

$c_p(k_p)$	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
$\tau_p^{\min}$	Minimum dwell time of train at platform p
$h_p^{\min}$	Minimum headway of platform p
$\lambda_{j,m}^{\mathrm{station}}(k_p)$	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
$\beta_{p,q,m}^{\mathrm{train}}$	Transfer rate of passengers from platform p to platform q with station m as their destination
$\theta_{p,q}^{\mathrm{trans}}$	Average walking time for passengers walking from platform p to platform q
$\theta_{p,q}^{\mathrm{duration}}$	Duration time for the transfer process from platform p to platform q

Decision variables

$a_p(k_p)$	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

Output variables

$r_p(k_p)$	Running time of train from platform p to its successor platform $\mathrm{s}^{\mathrm{pla}}(p)$ 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 $k_p c_p$
$n_{p,m}^{\mathrm{arrive,new}}(k_p)$	Number of passengers outside the urban rail network with destination m arriving at platform p at cycle k_p
$n_{p,m}^{\mathrm{arrive,trans}}(k_p)$	Number of transfer passengers with destination m arriving at platform p at cycle k_p
$n_{p,m}^{\mathrm{before}}(k_p)$	Number of passengers at platform p with station m as their destination immediately before the departure of train k_p
$n_{p,m}^{\mathrm{board}}(k_p)$	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

B. 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 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 platform.[1] 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

$$\lambda_{p,m}^{\text{original}}(t) = \beta_{j,p,m}\lambda_{j,m}^{\text{station}}(t), \forall p \in P_j, \forall m \in S, \tag{1}$$

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. 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:

$$\lambda_{p,m}(k_p) = \frac{1}{c_p(k_p)} \int_{L_p(k_p)}^{L_p(k_p)+c_p(k_p)} \lambda_{p,m}^{\text{original}}(t)\mathrm{d}t, \tag{2}$$

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.

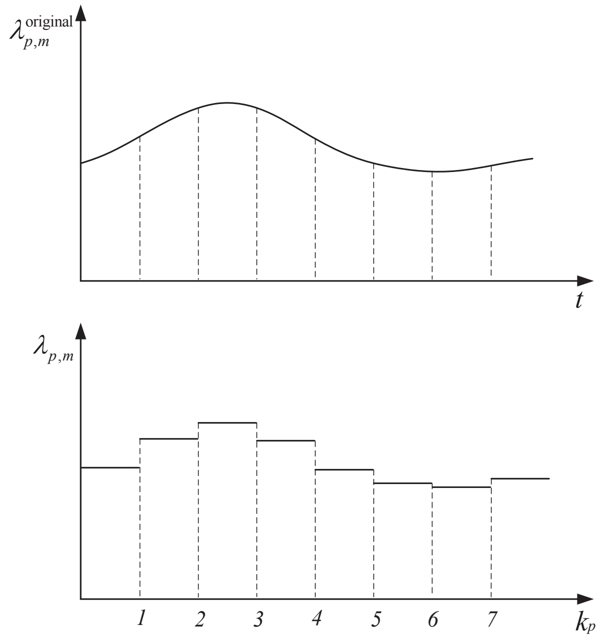

Fig. 1. Illustration of approximating passenger arrival rate.

According to the definition of cycle, only one train would visit platform p at cycle k_p; therefore, in this paper, 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

$$\begin{aligned} n_{p,m}(k_p+1) =& n_{p,m}(k_p) + n_{p,m}^{\text{arrive,new}}(k_p) \\ &+ n_{p,m}^{\text{arrive,trans}}(k_p) - n_{p,m}^{\text{board}}(k_p), \end{aligned} \tag{3}$$

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. 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:

$$n_{p,m}^{\text{arrive,new}}(k_p) = c_p(k_p)\lambda_{p,m}(k_p), \tag{4}$$

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,

[1] The 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 [37].

respectively. Then, we introduce a binary variable $y_{k_q,q,k_p,p}$ to represent the connection of trains at a transfer station:

$$y_{k_q,q,k_p,p} = \begin{cases} 1, \text{ if } \quad d_p(k_p-1) < a_q(k_q) + \theta^{\text{trans}}_{q,p} \leq d_p(k_p); \\ 0, \quad \text{otherwise}, \end{cases} \tag{5}$$

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 (5), the number of passengers $n^{\text{arrive,trans}}_{p,m}(k_p)$ transferring from other platforms of station j and arriving at platform p before the departure of train k_p can be calculated by

$$n^{\text{arrive,trans}}_{p,m}(k_p) = \sum_{q \in \text{plat}(p)} \sum_{k_q \in \mathcal{N}_q} y_{k_q,q,k_p,p} n^{\text{trans}}_{q,p,m}(k_q), \tag{6}$$

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

$$\begin{aligned} n^{\text{before}}_{p,m}(k_p) =& n_{p,m}(k_p) + (d_p(k_p) - L_p(k_p))\lambda_{p,m}(k_p) \\ &+ n^{\text{arrive,trans}}_{p,m}(k_p), \end{aligned} \tag{7}$$

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

$$n^{\text{before}}_{p}(k_p) = \sum_{m \in S} n^{\text{before}}_{p,m}(k_p). \tag{8}$$

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

$$n^{\text{board}}_{p}(k_p) = \min\left(C_{\max,k_p} - n^{\text{remain}}_{p}(k_p),\ n^{\text{before}}_{p}(k_p)\right), \tag{9}$$

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

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

$$n^{\text{after}}_{p}(k_p) = n^{\text{before}}_{p}(k_p) - n^{\text{board}}_{p}(k_p). \tag{10}$$

If we define

$$\lambda_p(k_p) = \sum_{m \in S} \lambda_{p,m}(k_p), \tag{11}$$

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

$$n^{\text{after}}_{p,m}(k_p) = n^{\text{after}}_{p}(k_p) \frac{\lambda_{p,m}(k_p)}{\lambda_p(k_p)}, \tag{12}$$

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, $n^{\text{after}}_{p,m}(k_p)$ can be computed linearly.

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

$$n^{\text{board}}_{p,m}(k_p) = n^{\text{before}}_{p,m}(k_p) - n^{\text{after}}_{p,m}(k_p). \tag{13}$$

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

$$n^{\text{trans}}_{p,q,m}(k_p) = \beta^{\text{train}}_{p,q,m}\ n^{\text{depart}}_{\mathrm{p}^{\text{pla}}(p),m}(k_p), \forall q \in \text{plat}(p)/\{p\}, \tag{14}$$

where $n^{\text{depart}}_{\mathrm{p}^{\text{pla}}(p),m}(k_p)$ denotes the number of passengers with destination m on train k_p immediately after the train departure from the predecessor platform $\mathrm{p}^{\text{pla}}(p)$ of platform p, and $\beta^{\text{train}}_{p,q,m}$ 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

$$\sum_{q \in \text{plat}(p)} \beta^{\text{train}}_{p,q,m} = 1. \tag{15}$$

The transfer rate of passengers can be obtained based on the historical data or by a shortest path algorithm, e.g., Yen's algorithm [38], assuming that passengers select the platform corresponding to the shortest path to reach their destination.

Remark 2.1 It is worth noting that $\beta^{\text{train}}_{p,p,m}$ 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 $n^{\text{trans}}_{p,p,m}(k_p) = 0$. In particular, If the arrival station is not a transfer station, then $\beta^{\text{train}}_{p,p,m} = 1$.

Remark 2.2 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 $\beta^{\text{train}}_{p,p,j} = 1$ and $\beta^{\text{train}}_{p,q,j} = 0, \forall q \in \text{plat}(p)/\{p\}$, 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 $n^{\text{trans}}_{p,q,j}(k_p) = 0$, $\forall q \in \text{plat}(p)$.

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

$$n^{\text{remain}}_{p,m}(k_p) = \beta^{\text{train}}_{p,p,m}\ n^{\text{depart}}_{\mathrm{p}^{\text{pla}}(p),m}(k_p), \forall m \in S/\{\text{sta}(p)\}. \tag{16}$$

In other words, $n^{\text{remain}}_{p,m}(k_p)$ 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 $\text{sta}(p)$, $n^{\text{remain}}_{p,\text{sta}(p)}(k_p) = 0$.

Having (16), the total number of passengers $n^{\text{remain}}_{p}(k_p)$ remaining on train k_p at platform p after the alighting process can be calculated by

$$n^{\text{remain}}_{p}(k_p) = \sum_{m \in S} n^{\text{remain}}_{p,m}(k_p). \tag{17}$$

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

$$n_{p,m}^{\text{depart}}(k_p) = n_{p,m}^{\text{remain}}(k_p) + n_{p,m}^{\text{board}}(k_p). \tag{18}$$

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

$$n_p^{\text{depart}}(k_p) = \sum_{m \in S} n_{p,m}^{\text{depart}}(k_p). \tag{19}$$

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

$$n_p^{\text{alight}}(k_p) = n_{\text{p}^{\text{pla}}(p)}^{\text{depart}}(k_p) - n_p^{\text{remain}}(k_p), \tag{20}$$

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

C. Train Operation Model

In this paper, 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

$$L_p(k_p) < a_p(k_p) < d_p(k_p) \leq L_p(k_p) + c_p(k_p), \tag{21}$$

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

$$\tau_p(k_p) = d_p(k_p) - a_p(k_p), \tag{22}$$

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

$$\tau_p(k_p) \geq \tau_p^{\min}, \tag{23}$$

where $\tau_p^{\min}$ 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

$$a_p(k_p) \geq d_p(k_p - 1) + h_p^{\min}, \tag{24}$$

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 $\text{s}^{\text{pla}}(p)$ of platform p is

$$a_{\text{s}^{\text{pla}}(p)}(k_p) = d_p(k_p) + r_p(k_p), \tag{25}$$

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

$$r_p^{\min}(k_p) \leq r_p(k_p) \leq r_p^{\max}(k_p), \tag{26}$$

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 $\text{s}^{\text{pla}}(p)$, 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.

IV. Model Predictive Control for Passenger-Oriented Timetable Scheduling

Model predictive control is a control method that repeatedly solves finite-horizon optimization problems and implements optimized decisions in a moving horizon manner [39]. 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 paper, 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 III can be used as the prediction model for the MPC controller. The model is a nonlinear model, and, for each cycle, it can be represented as follows:

$$n_{p,m}(k_p + 1) = f\left(n_{p,m}(k_p), n_{q,p,m}^{\text{trans}}(k_q), g_p(k_p)\right), \tag{27}$$

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; $n_{q,p,m}^{\text{trans}}(k_p)$ 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 paper, an objective function of the following form is considered:

$$J=\sum_{p\in P}\sum_{k_p\in\mathcal{N}_p(\kappa)}\left(n_p^{\text{before}}(k_p)c_p(k_p)+\xi n_p^{\text{after}}(k_p)c_p(k_p)\right), \tag{28}$$

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 (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

$$\left\{\begin{array}{l}\min\limits_{\mathbf{g}(\kappa)} J=\sum\limits_{p\in P}\sum\limits_{k_p\in\mathcal{N}_p(\kappa)}\left(n_p^{\text{before}}(k_p)+\xi n_p^{\text{after}}(k_p)\right)c_p(k_p),\\ \text{s.t.}\quad (1)-(14),\ (16)-(26),\end{array}\right. \tag{29}$$

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 (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. 2.

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 (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.

V. Solution Approaches

The resulting optimization problem in Section IV is a nonlinear non-convex problem because of (5), (6), and (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.

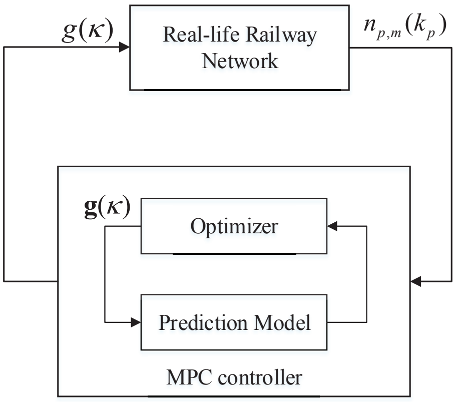

Fig. 2. MPC for passenger-oriented timetable scheduling.

A. 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 [40]. In SQP, a sequence of quadratic programming problems is solved to get descent directions of the original problem. The objective function and the constraints of the optimization problem should be continuously differentiable when applying the SQP algorithm. In this paper, the optimization problem has some points of non-smoothness due to the min function in (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 paper.

B. 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 (5), (6), and (9).

In order to transform (5) into a mixed logical dynamical (MLD) system [41], the time checking binary variable $x_{k_q,q,k_p,p}$ is introduced as

$$x_{k_q,q,k_p,p}=\left\{\begin{array}{ll}1, & \text{if}\quad a_q(k_q)+\theta_{q,p}^{\text{trans}}\le d_p(k_p);\\ 0, & \text{otherwise},\end{array}\right. \tag{30}$$

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 window[2]. Then, (30) is equivalent to

$$\begin{cases} a_q(k_q)+\theta^{\text{trans}}_{q,p}-d_p(k_p)\leq\left(1-x_{k_q,q,k_p,p}\right)\left(M_{\text{t}}-d_p(k_p)\right); \\ a_q(k_q)+\theta^{\text{trans}}_{q,p}-d_p(k_p)\geq\varepsilon+x_{k_q,q,k_p,p}\left(m_{\text{t}}-d_p(k_p)-\varepsilon\right), \end{cases} \tag{31}$$

where ε is a sufficient small number (generally the machine precision) [41]. Define

$$y_{k_q,q,k_p,p}=x_{k_q,q,k_p,p}-x_{k_q,q,k_p-1,p}. \tag{32}$$

Then, based on Lemma 5.1, (5) is equivalent to (31) and (32).

Lemma 5.1 Given $y_{k_q,q,k_p,p}=x_{k_q,q,k_p,p}-x_{k_q,q,k_p-1,p}$, $d_p(k_p-1)<a_q(k_q)+\theta^{\text{trans}}_{q,p}\leq d_p(k_p)$ 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 (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 $a_q(k_q)+\theta^{\text{trans}}_{q,p}$:
if $a_q(k_q)+\theta^{\text{trans}}_{q,p}>d_p(k_p)$, 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 $d_p(k_p-1)<a_q(k_q)+\theta^{\text{trans}}_{q,p}\leq d_p(k_p)$, 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 $a_q(k_q)+\theta^{\text{trans}}_{q,p}\leq d_p(k_p-1)$, we have $x_{k_q,q,k_p,p}=1$ and $x_{k_q,q,k_p-1,p}=1$; then, $y_{k_q,q,k_p,p}=0$. ■

The min function in (9) can be handled by introducing the auxiliary binary variable $\delta^{\text{board}}_{k_p,p}$ and the auxiliary real variable $f_{k_p,p}$. Define

$$f_{k_p,p}=\left(C_{\max,k_p}-n^{\text{remain}}_p(k_p)\right)-n^{\text{before}}_p(k_p), \tag{33}$$

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

$$\begin{cases} f_{k_p,p}\leq M_{\text{p}}\left(1-\delta^{\text{board}}_{k_p,p}\right), \\ f_{k_p,p}\geq\varepsilon+\left(m_{\text{p}}-\varepsilon\right)\delta^{\text{board}}_{k_p,p}, \end{cases} \tag{34}$$

where M_{p} and m_{p} are the maximum value and the minimum value of $f_{k_p,p}$, respectively[3].
Having (34), the expression (9) is equivalent to

$$\begin{aligned} n^{\text{board}}_p(k_p)=&\,\delta^{\text{board}}_{k_p,p}\left(C_{\max,k_p}-n^{\text{remain}}_p(k_p)\right) \\ &+\left(1-\delta^{\text{board}}_{k_p,p}\right)n^{\text{before}}_p(k_p). \end{aligned} \tag{35}$$

After introducing auxiliary variables in (30) and (34), we still have nonlinear terms, i.e., the product of binary variables and real variables in (6), (31), and (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 [41], [42]. The details of the transformation are given in Appendix A.

In summary, we introduce three equivalence transformations, i.e., (5) with (31)-(32), (9) with (33)-(35), and (37) with (38) in Appendix A. The proof for "(5) is equivalent to (31)-(32)" is provided in Lemma 5.1. The equivalence of "(9) and (33)-(35)" and "(37) and (38)" can be found in [41], [42]. Based on the above transformations, we can finally obtain an MILP problem that is exactly equivalent to the original optimization problem.

C. Simplified Mixed-Integer Linear Programming Approach

In Section V-B, several auxiliary variables and constraints are introduced to handle the train capacity constraints in (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 (9), and hence the constraints (33), (34), and (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., (9) will be replaced with:

$$n^{\text{board}}_p(k_p)=n^{\text{before}}_p(k_p). \tag{36}$$

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

As mentioned in Section V-A, 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.

VI. 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 [13], [33] 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.

A. 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 [13], [33]. Therefore, in this paper, we define the model in [13], [33] 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. 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

[2] The value of M_{t} can be the length of the planning time window, i.e., $M_{\text{t}}=t_{\text{end}}$, and m_{t} can be equal to 0.

[3] The value of M_{p} can be a very large value related to train capacities, i.e., $M_{\text{p}}=10\cdot C_{\max,k_p}$, and m_{t} can be a small value, i.e., $m_{\text{p}}=-10\cdot C_{\max,k_p}$.

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 8GB RAM. The main parameters associated with the simulation are listed in Table I. 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.

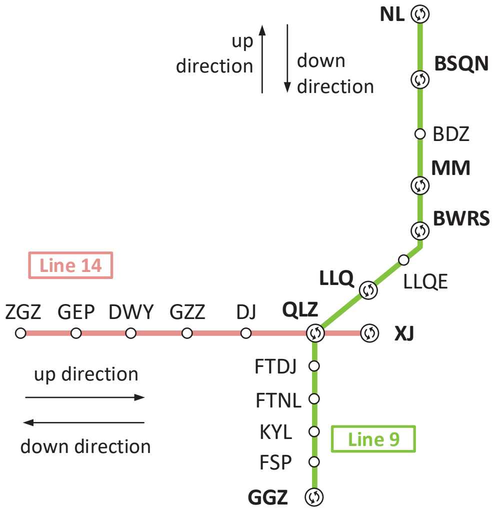

Fig. 3. Real-life network of 2 lines from Beijing subway.

TABLE I
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

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 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 II.

TABLE II
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%

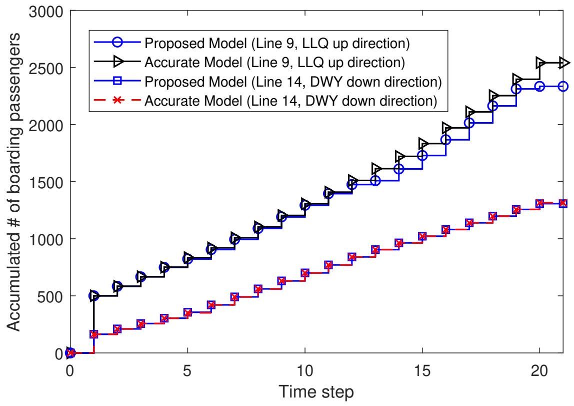

Fig. 4. Accumulated number of boarding passengers at platforms.

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.

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. 5).

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. 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

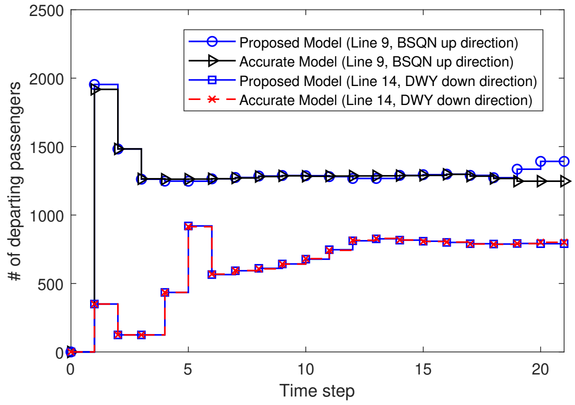

Fig. 5. Number of departing passengers at each time step.

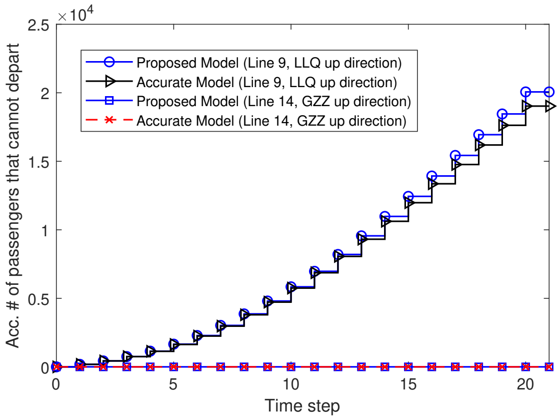

Fig. 6. Accumulated number of passengers that cannot board at platforms.

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.

B. 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 V, which can reflect the effectiveness and the real-time feasibility of the developed MPC controller. The model in [13], [33] 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 VI-A, and the parameters for train operation constraints are listed in Table III, 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 III
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 VI-A. 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 IV, where the performance is the value of the objective function in (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 IV
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.

C. Closed-Loop Control for Real-Time Timetable Scheduling

In Section VI-A and Section VI-B, 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 [13], [33]) and the newly developed model.

The urban rail network is shown in Fig. 3, and all settings related to the numerical experiment are identical with Section VI-B. The simulation is conducted for 15 time steps and the prediction horizon of MPC is 5 (i.e. $5 \cdot T_{\mathrm{ctrl}}$). In the developed model, variables related to the number of passengers are updated every time step.

TABLE V
PERFORMANCE OF MPC IN REAL-TIME TIMETABLE SCHEDULING

	Prediction model	Performance	CPU time (s)	
			t_{avrg}	$t_{\max}$
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 VI-B that MILP-based formulation performs best among the optimization approaches provided in Section IV; 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 VI-B.

As Table V 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.

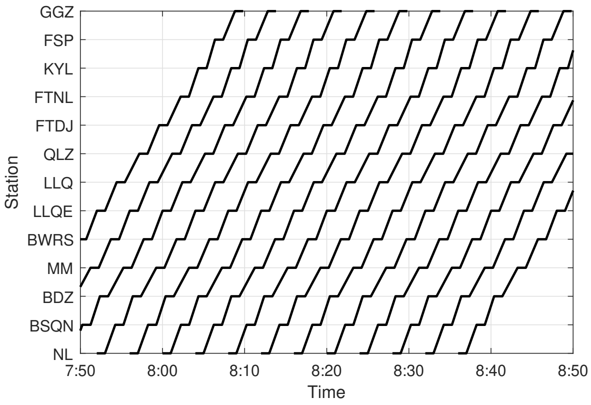

Fig. 7. Basic timetable.

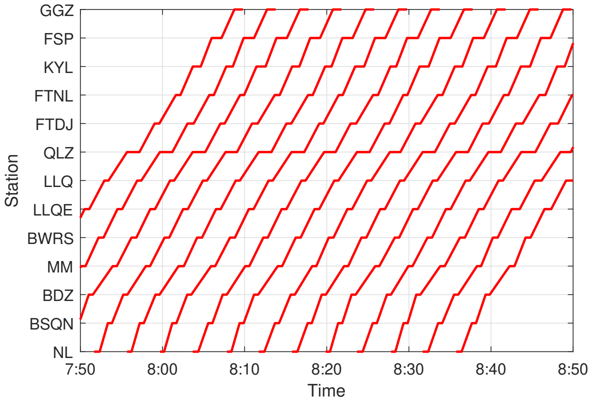

Fig. 8. Timetable generated by SQP-based MPC.

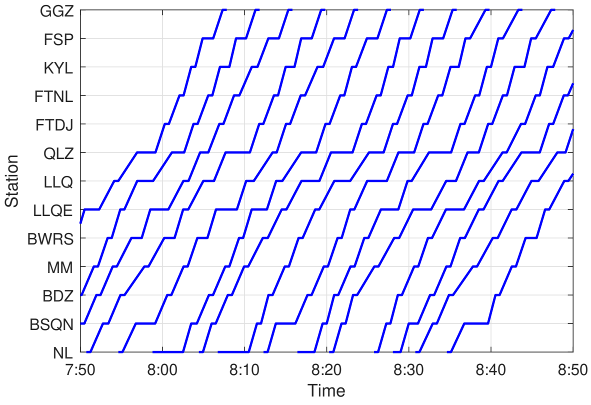

Fig. 9. Timetable generated by MILP-based MPC.

The basic timetable, the timetable generated by SQP-based MPC, and the timetable generated by MILP-based MPC are shown in Fig. 7, Fig. 8, and Fig. 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.

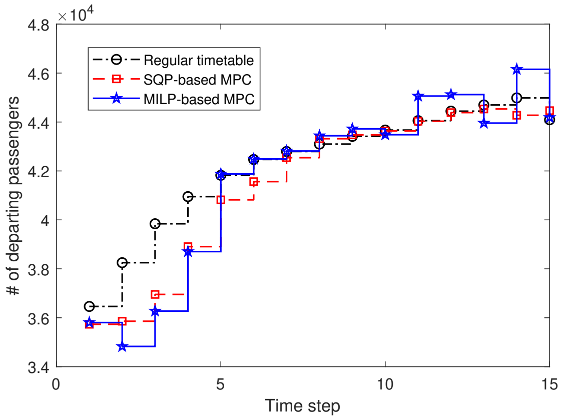

Fig. 10. Total number of departing passengers at each time step.

The total number of departing passengers for all lines and all platforms is depicted in Fig. 10. The timetable obtained from the MILP-based MPC approach results in more boarding and departing passengers, which means the resulting timetable can make better use of the available trains.

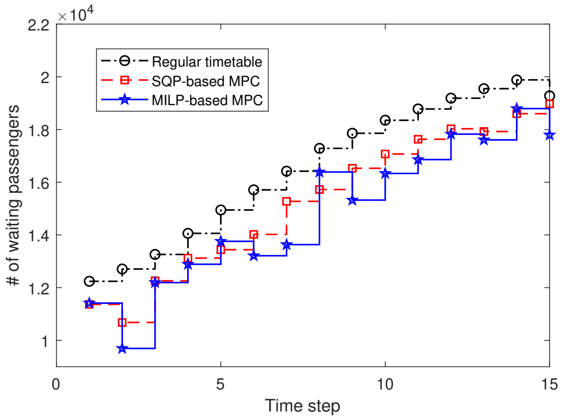

Fig. 11. Total number of waiting passengers at each time step.

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. 11 and Fig. 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.

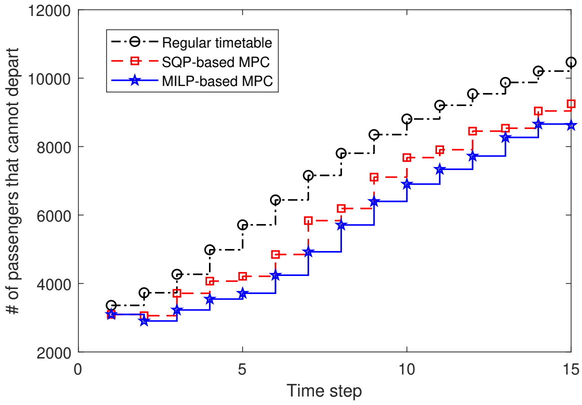

Fig. 12. Total number of passengers that cannot depart at each time step.

VII. Conclusions

In this paper, 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.

Appendix A

The product of real-valued variable f and binary variable δ can be transformed into linear inequalities by introducing an auxiliary real-valued variable z using the method in [41], [42], with

$$z = \delta \cdot f. \tag{37}$$

Then, $z = \delta \cdot f$ is equivalent to

$$\begin{cases} z \leq M_f \, \delta, \\ z \geq m_f \, \delta, \\ z \leq f - m_f(1-\delta), \\ z \geq f - M_f(1-\delta), \end{cases} \tag{38}$$

where M_f and m_f denote the maximum and minimum value of f, respectively.

References

[1] 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*, vol. 14, no. 10, pp. 1199–1211, 2006.

[2] 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*, vol. 183, no. 2, pp. 643–657, 2007.

[3] 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*, 2022.

[4] 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*, vol. 118, pp. 193–227, 2018.

[5] 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*, vol. 19, no. 1, pp. 99–112, 2022.

[6] P. Wang, L. Ma, R. M. Goverde, and Q. Wang, "Rescheduling trains using petri nets and heuristic search," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 726–735, 2016.

[7] 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*, vol. 21, no. 9, pp. 3621–3633, 2020.

[8] 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*, vol. 20, no. 7, pp. 2774–2785, 2019.

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

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

[11] 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*, vol. 97, pp. 182–213, 2017.

[12] Y. Zhu and R. Goverde, "Integrated timetable rescheduling and passenger reassignment during railway disruptions," *Transportation Research Part B: Methodological*, vol. 140, pp. 282–314, 2020.

[13] 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*, vol. 60, pp. 1–23, 2015.

[14] 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*, vol. 295, no. 1, pp. 183–202, 2021.

[15] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[16] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[17] 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*, vol. 18, no. 4, pp. 2582–2592, 2021.

[18] 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*, vol. 12, no. 3, pp. 846–856, 2011.

[19] 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*, vol. 24, pp. 122–140, 2012.

[20] 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*, vol. 44, no. 1, pp. 175–192, 2010.

[21] 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*, vol. 2607, no. 1, pp. 82–92, 2017.

[22] 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*, vol. 78, pp. 182–201, 2015.

[23] R. M. Goverde, "Railway timetable stability analysis using max-plus system theory," *Transportation Research Part B: Methodological*, vol. 41, no. 2, pp. 179–201, 2007.

[24] 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*, vol. 26, no. 2, pp. 183–223, 2016.

[25] 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*, vol. 14, no. 2, pp. 883–893, 2013.

[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*, vol. 54, pp. 15–39, 2015.

[27] 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*, vol. 115, pp. 41–71, 2018.

[28] 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*, vol. 115, pp. 72–94, 2018.

[29] D. Canca, E. Barrena, E. Algaba, and A. Zarzo, "Design and analysis of demand-adapted railway timetables," *Journal of Advanced Transportation*, vol. 48, no. 2, pp. 119–137, 2014.

[30] 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*, vol. 15, no. 6, pp. 2658–2670, 2014.

[31] 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*, vol. 76, pp. 117–135, 2015.

[32] 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*, vol. 110, pp. 26–59, 2018.

[33] 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*, 2021.

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

[35] 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*, vol. 39, no. 11, pp. 2578–2593, 2012.

[36] 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*, 2022.

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

[38] J. Y. Yen, "Finding the K shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[39] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[40] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4, pp. 1–51, 1995.

[41] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

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