

Technical report 15-038

# **Passenger-demands-oriented train scheduling for an urban rail transit network\***

Y. Wang, T. Tang, B. Ning, T.J.J. van den Boom, and B. De Schutter

*If you want to cite this report, please use the following reference instead:*

Y. Wang, T. Tang, B. Ning, T.J.J. van den Boom, and B. De Schutter, "Passenger-demands-oriented train scheduling for an urban rail transit network," *Transportation Research Part C*, vol. 60, pp. 1–23, Nov. 2015. doi:[10.1016/j.trc.2015.07.012](https://doi.org/10.1016/j.trc.2015.07.012)

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/15\\_038.html](https://pub.bartdeschutter.org/abs/15_038.html)

# Passenger-Demands-Oriented Train Scheduling for an Urban Rail Transit Network

Yihui Wang<sup>a,b,\*</sup>, Tao Tang<sup>a</sup>, Bin Ning<sup>a</sup>, Ton J.J. van den Boom<sup>b</sup>, Bart De Schutter<sup>b</sup>

<sup>a</sup>State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, P.R. China

<sup>b</sup>Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands

---

## Abstract

This paper considers the train scheduling problem for an urban rail transit network. We propose an event-driven model that involves three types of events, i.e., departure events, arrival events, and passenger arrival rates change events. The routing of the arriving passengers at transfer stations is also included in the train scheduling model. Moreover, the passenger transfer behavior (i.e., walking times and transfer times of passengers) is also taken into account in the model formulation. The resulting optimization problem is a real-valued nonlinear nonconvex problem. Nonlinear programming approaches (e.g., sequential quadratic programming) and evolutionary algorithms (e.g., genetic algorithms) can be used to solve this train scheduling problem. The effectiveness of the event-driven model is evaluated through a case study.

*Keywords:* train scheduling, passenger demands, event-driven, urban rail transit network

---

## 1. Introduction

Nowadays, urban rail transit systems play a key role in public transportation for big cities (e.g., Beijing, New York, Paris) since they combine high transport capacity and high efficiency. A safe, fast, energy-efficient, and comfortable urban rail system is important for the economic, environmental, and social objectives of big cities. The railway planning process is essential for urban rail transit operations and management, and in general it consists of five phases (Bussieck et al., 1997): demand analysis, line planning, train scheduling, rolling stock planning, and crew scheduling. The focus of this paper is on train scheduling for an urban rail transit network where the aim is to reduce the operation costs of the trains and to enhance passenger satisfaction. Passenger satisfaction can be characterized by the waiting times, the in-vehicle times, and the number of transfers, while the operation costs are determined by the number of train services and the energy consumption of the trains.

In most urban rail transit systems, the transit lines are separate from each other and each direction of a line has a separate rail track. Hence, trains usually do not overtake each other. In

---

\*Corresponding author. Tel:+86 10 5168 8607; Fax:+86 10 5168 8532. Y. Wang, T. Tang and B. Ning are also with Beijing Laboratory of Urban Mass Transit and Beijing Key Laboratory of Urban Mass Transit Automation and Control in Beijing Jiaotong University.

*Email addresses:* yihui.wang@tudelft.nl (Yihui Wang), ttang@bjtu.edu.cn (Tao Tang), bning@bjtu.edu.cn (Bin Ning), a.j.j.vandenboom@tudelft.nl (Ton J.J. van den Boom), b.deschutter@tudelft.nl (Bart De Schutter)

*Preprint submitted to Elsevier*

addition, passengers may need to make several interchanges between different lines to arrive at their destinations. Therefore, it is important to take passenger transfers into account in the train scheduling to shorten the total travel time of passengers. Hence, the train schedules for different lines should be coordinated in order to enable smooth passenger transfer and to minimize the total travel time of passengers. Moreover, the passenger demands may vary significantly along the day at different stations, i.e., the number of passengers getting on and getting off trains depends on the location of the stations and the time of day (e.g., morning-peak hours, afternoon-peak hours, and off-peak hours).

Train scheduling for *interurban* rail transit systems has been studied for decades (Szpigel, 1972; Petersen et al., 1986; Kraay et al., 1991; Higgins et al., 1996; Cordeau et al., 1998; Ghoseiri et al., 2004; D’Ariano et al., 2007). In interurban rail transit systems the available resources, e.g., the tracks and the crossings, are shared by trains with different origins and destinations. Thus, the trains may overtake and cross each other at some specific locations, such as sidings and crossings. In this paper, we concentrate however on urban rail transit systems, where the lines usually have double tracks, and train overtaking and crossing are normally not allowed during the operations.

Regular schedules with fixed headways are often used in practice for urban rail transit systems, e.g., every seven minutes there is a train entering a station. Kwan and Chang (2005) applied a heuristic-based evolutionary algorithm to optimize the frequency (or headway) between trains, where the operation costs and the passenger dissatisfaction are included in the performance index. Liebchen (2006, 2008) formulated the train scheduling problem as a periodic event-scheduling problem based on a graph model and obtained periodic schedules for the Berlin subway system using genetic algorithms and integer programming. Su et al. (2013) and Li and Lo (2014) optimized the cyclic train schedule together with the driving strategy to minimize the energy consumption through the utilization of regenerative energy. Regular schedules can reduce the passenger waiting time if the passenger arrival process at stations is a uniform process or a Poisson process (Niu and Zhou, 2013; Barrena et al., 2014). However, regular schedules may result in longer passenger waiting times and travel times under time-varying passenger demands.

Cury et al. (1980) obtained a nonperiodic train schedule aimed at minimizing passenger dissatisfaction and operation costs based on a model of the train movements and the passenger behavior. The resulting nonlinear scheduling problem was recast into several subproblems by Lagrangian relaxation and then solved in a hierarchical manner. The headway between trains in the optimal schedules obtained by Cury et al. (1980) varies with time instead of being a constant. Since the convergence rate of the hierarchical decomposition algorithm of Cury et al. (1980) can be quite poor in some cases, Assis and Milani (2004) proposed a model predictive control algorithm based on linear programming to optimize the train schedule. The algorithm proposed by Assis and Milani (2004) can effectively generate train schedules for the whole day. Furthermore, a demand-oriented timetable design has been proposed by Albrecht (2009), where the optimal train frequency and the capacity of trains are first determined and then the schedules of the trains are optimized. Niu and Zhou (2013) optimized the train schedules for an urban rail transit line with consideration of time-varying origin-destination passenger demands in heavily congested situations. In particular, a genetic algorithm was used to solve the resulting nonlinear programming problem. Furthermore, Niu et al. (2015) considered the train scheduling with time-dependent demand and skip-stop patterns to minimize the passenger waiting time. A branch-and-cut algorithm is presented by Barrena et al. (2014) to minimizing average passenger waiting time with consideration of a dynamic passenger demand. Furthermore, in (Wang et al., 2015) we proposed an iterative convex programming approach for train scheduling for urban rail

transit lines with the aim of minimizing the total travel time and the total energy consumption.

As mentioned before, passenger transfers are important when optimizing train schedules for an urban rail transit system. The passenger transfer behavior and transfer waiting times are considered by Wong et al. (2008), who present a mixed-integer programming optimization model to synchronize the train schedules for different urban rail transit lines in order to minimize the waiting times of the transfer passengers. In addition, Vázquez-Abad and Zubieta (2005) proposed a stochastic approximation approach to adjust the frequencies of different urban transit lines according to the observed variable passenger demands. However, the energy consumption of the trains and the dwell times at stations are not included in the model of Vázquez-Abad and Zubieta (2005).

The current paper extends the previous research in the following aspects:

- In (Wang et al., 2015), we have used a time-driven model for the train scheduling. However, in this paper, we present an event-driven model, which has a higher computational efficiency than the time-driven model. With this model, we can characterize the time-varying origin-destination passenger demand.
- In (Wang et al., 2015), we have considered a single urban transit line. In this paper, we consider an urban rail transit *network*. In addition, the passenger transfer behavior and the route choice of passengers at transfer stations are also included in the problem formulation.
- The passenger travel time is computed more accurately by including the walking time for passengers from entrances to platforms, the waiting time, the in-vehicle time, the transfer time, and the walking time for passengers from platforms to exit stations.

The rest of the paper is structured as follows. Section 2 introduces the three types of events and formulates an event-driven model for trains. Section 3 describes the performance criteria and constraints of the train scheduling problem. In addition, we also discuss the initial conditions and solution approaches in this section. In Section 4, the performance of the proposed event-driven model is evaluated via a case study. Finally, conclusions and recommendations are provided in Section 5.

## 2. Model formulation

In this section, we first discuss the assumptions and present notations in Section 2.1. Three types of events of the event-driven model, i.e., departure events, arrival events, passenger arrival rates change events, are then proposed in Section 2.2 to describe the departure and arrival of trains and the time-varying passenger demands. In Section 2.3, a passenger arrival rate query module is introduced to obtain the actual passenger arrival rates for each platform at any time. Finally, the state update equations for each type of events of the event-driven model are presented in Section 2.4 to calculate the number of onboard passengers, the number of waiting passengers, the number of transfer passengers, etc.

### 2.1. Notations and assumptions

Table 1 summarizes the parameters and variables used in this paper to describe the train scheduling problem. In addition, the symbols are also explained in text itself.

[Place Table 1 about here]

Consider an urban rail transit network with  $L$  lines and  $J$  stations. Let  $S_{\text{ln}}$  and  $S_{\text{sta}}$  be the sets of lines and station indices, respectively. In practice, a station could have several platforms and we denote the set of platforms as  $S_{\text{pla}}$ . Note that a physical line with two directions is defined as two separate lines in this paper. We make the following assumptions:

- A.1 There is no shared platform for different lines in the urban rail transit network.
- A.2 A platform can only accommodate one train at a time and no overtaking can occur at any point of the line.

Assumption A.1 is made for the sake of simplicity in the numbering of platforms. With Assumption 1, a platform is uniquely identified to a specific line, i.e., a line can be defined by a set of platforms. If passengers want to transfer from one line to the other, they need to walk from one platform to the other. Note that the assumption holds for most urban transit systems, e.g. the subway networks in Beijing, New York, Tokyo, Paris, and Rome. However, an island platform, which is common for passenger transfers from one line to another without walking a long distance and/or taking escalators or stairs, can be also modeled by considering it as two distinct platforms, where the average walking time for passenger transfers is relatively small. Assumption A.2 generally holds for most urban transit systems too. Furthermore, in practice, the trains of different lines are operated separately, which means that trains are not shared between different lines.

Platforms are uniquely defined in the network and therefore an urban rail transit line can be defined by a set of platforms. We denote the predecessor of platform  $p$  on a given line as  $\text{p}^{\text{pla}}(p)$  and the successor of platform  $p$  on that line as  $\text{s}^{\text{pla}}(p)$ . In order to distinguish the different running cycles of the physical trains, train services are introduced, where each train service has a unique service number that uniquely identifies a train and its current cycle. Let  $I_\ell$  be the total number of physical trains on transit line  $\ell$ ; then the total number of physical trains in the network is  $I_{\text{net}} = \sum_{\ell \in S_{\text{ln}}} I_\ell$ . So the physical trains in the network could be numbered as  $1, 2, \dots, I_{\text{net}}$ . The service number of trains can then be written as  $1, 2, \dots, I_{\text{net}}, I_{\text{net}}+1, I_{\text{net}}+2, \dots, 2I_{\text{net}}, \dots, N_{\text{cyc}}I_{\text{net}}$ , where  $N_{\text{cyc}}$  is the number of cycles for trains in the given scheduling period. Therefore, train service  $i$  corresponds to physical train  $[(i-1) \bmod I_{\text{net}}] + 1$ . For the sake of simplicity, we use “train  $i$ ” as a short-hand for “train service  $i$ ” from now on. In addition, the set of indices of all train services is denoted by  $S_{\text{tra}}$ . The predecessor and successor of train  $i$  on a given line are denoted as  $\text{p}^{\text{tra}}(i)$  and  $\text{s}^{\text{tra}}(i)$ , respectively. The start time and end time of the scheduling period are denoted as  $t_0$  and  $t_{\text{end}}$ .

## 2.2. Three types of events of the event-driven model

We model the train scheduling problem with consideration of passenger demands using three types of events:

- Departure events: representing the departure of a train at a station,
- Arrival events: representing the arrival of a train at a station,
- $\lambda$ -change events: representing the change of passenger arrival rates at a platform.

To describe the operation of trains, we propose an event-driven model consisting of a continuous part describing the movement of trains running from one station to another through the network,

and of the discrete events listed above. The  $k$ th event  $e_k$  occurring in the event-driven system is denoted as

$$e_k = (t_k, Y_{\text{type},k}, i_k, p_k), \quad (1)$$

where  $k$  is the event counter,  $t_k$  is the time instant at which event  $e_k$  occurs,  $Y_{\text{type},k}$  is the event type, which can have three possible values, i.e., ‘d’, ‘a’, or ‘ $\lambda$ ’ corresponding to a departure event, an arrival event, or a  $\lambda$ -change event,  $i_k$  is the train number, and  $p_k$  is the platform number.

[Place Figure 1 about here]

The model structure of the event-driven system is illustrated in Figure 1. The control inputs of the system are the departure times, the arrival times of trains, and the splitting rates of passenger flows at transfer stations. So the departure and arrival events at stations are the controlled events of the urban rail transit system since their event times are directly influenced by the inputs to the system. All the  $\lambda$ -change events are autonomous events and they are triggered by other events or by the environment; their event times cannot be controlled directly. We assume that the initial states, passenger arrival rates, line data, and train data, etc. are known or can be estimated via the available information. We denote the current time as  $t_{\text{current}}$  (see also Figure 2). Let  $\tau_{\text{process}}$  be the processing time for data preparation for the train scheduling. All the events that happened in the past, i.e., for which the event time is smaller than  $t_{\text{current}} + \tau_{\text{process}}$ , are known to the event-driven system and the set of these events is denoted as  $S_{\text{known}}$ . The set of events that will happen in the future is denoted as  $S_{\text{unknown}}$ . In addition, the departure events, arrival events, and the updates of the states should satisfy the constraints in the system, such as the headway constraints, train capacity constraints, and running time constraints. Furthermore, we introduce a global event list for the event-driven system. At any time, this list contains all the possible next events for all the trains and stations in the urban rail network. The next event of the system will be the event in the global event list with the smallest value of  $t_k$ , i.e. the event that will occur first, where ties are broken arbitrarily (The event-driven model structure is such that the order in which coinciding events are handled will not influence the state of the system since if an event triggers a next event there is always a minimum separation between the current event and the next event (see Section 2.4)). As a starting point, the global event list should be initialized based on the initial states of the system. Moreover, a  $\lambda$ -change query module is present to describe the passenger arrival rates for each platform (see Section 2.3 for more information). The details about the state updates and triggered events are given in Section 2.4.

[Place Figure 2 about here]

### 2.3. $\lambda$ -profile query module

The triggered  $\lambda$ -change events can be caused by the change of passenger arrivals at stations, the changes of splitting rates at transfer stations, and the passenger transfers at transfer stations. We introduce a  $\lambda$ -profile query module for each platform (see also Figure 3) to obtain the passenger arrival rate (Leemis, 1995). If platform  $p$  is not at a transfer station, then the query module only contains the base profile  $\lambda_{j,m}^{\text{station}}(\cdot)$  as explained in Section 2.3.1. However, if platform  $p$  is at a transfer station, then the query module for a platform stores the base profile and possibly additional update profiles due to splitting rate changes and passenger transfers, as explained in Section 2.3.2 and 2.3.3. [Place Figure 3 about here]

**Remark.** *The passenger splitting rate proposed in this paper is a fraction of the total passenger flow with the same destination that goes to a specific platform. The splitting rates are used to describe the route choices of passengers with the same destination. For example, at a transfer*

station passengers with the same destination could choose to stay on the line or to transfer to another line. In principle, passengers will make route choices based on the shortest route, or the minimum number of transfers, or their combination. These choices determine the splitting rates, which can be reconstructed based on online data estimation using offline data and/or passenger surveys.

### 2.3.1. Passenger arrivals at stations

The train scheduling model requires the real-time assessment of passenger arrival rates for different origins and destinations during the scheduling period. In the case of full state information, the actual passenger arrival rates can be obtained. However, this is not the case in practice, where we need to e.g. use the information collected by the advanced fare collection systems and estimate the passenger arrival rates based on the historical data and the current passenger flows (Wong and Tong, 1998). A typical profile for the passenger arrival rate at station  $j$  for passengers with destination  $m$  is given as the solid line in Figure 4, where the passenger arrival rate during the peak hours is much higher than that during the off-peak hours. A piecewise constant functions  $\lambda_{j,m}^{\text{station}}(\cdot)$  defined for  $t \in [t_0, t_{\text{end}}]$  is introduced to approximate the continuous passenger arrival rate at station  $j$  for passengers with destination  $m$ . [Place Figure 4 about here] These piecewise functions are the inputs to the event-driven model and we describe these piecewise constant functions via so-called base profiles. The base profiles are left-hand side continuous piecewise constant functions, which can be specified by a list of corner points as shown in Figure 5, where the corner points are marked with purple dots. Hence, the base profile shown in Figure 5 can be described by the following set of three corner points:

$$\{(t_1, \lambda_1), (t_2, \lambda_2), (t_3, \lambda_3)\}.$$

[Place Figure 5 about here]

### 2.3.2. Splitting rates changes at transfer stations

At a transfer station, passengers can choose to go to the platforms of different lines since there could be multiple routes available to go to their destination. The splitting of passenger flows at transfer stations can be influenced or controlled by the rail operator by providing route information and suggestions to passengers through information panels at the entrances of stations or through personal digital assistant (PDA) devices.

Consider transfer station  $j$  and one of its platforms  $p$ . Let  $\beta_{p,m}^{\text{station}}(\cdot)$  denote the splitting rate profile of the passengers flows that arrive at station  $j$ , have destination  $m$ , and go to platform  $p$  (see Figure 6). Note that the platforms are uniquely defined in the urban rail transit network, so for simplicity we do not include the index  $\ell$  of the corresponding line in the variables. The function  $\beta_{p,m}^{\text{station}}(\cdot)$  is also a left-hand side continuous piecewise constant function. In order to provide a consistent service to the passengers, the splitting rate should not change too often, e.g., 15 minutes. The passenger arrival rate at the platforms of station  $j$  can be calculated as follows: [Place Figure 6 about here]

$$\lambda_{p,m}(t_k) = \beta_{p,m}^{\text{station}}(t_k) \lambda_{j,m}^{\text{station}}(t_k), \quad \forall p \in P_j, \forall m \in S_{\text{sta}}, \quad (2)$$

where  $t_k$  is one of the corner points of the base profile or of the splitting rate change profiles and  $P_j$  is the set of platforms at transfer station  $j$ . Furthermore, the sum of all the splitting rates at transfer station  $j$  is always equal to 1, i.e.,

$$\sum_{p \in P_j} \beta_{p,m}^{\text{station}}(t_k) = 1, \quad \forall m \in S_{\text{sta}}. \quad (3)$$

The splitting rates at the transfer stations are the control variables for the train scheduling problem. The change of the splitting rates of passenger flows results in  $\lambda$ -change events at the platforms of a transfer station. The  $\Delta\lambda$  profiles caused by splitting rate changes are also piecewise constant functions, which can be described by a list of corner points in a similar way as base profiles.

The average walking time for passengers from entrances of station  $j$  to platform  $p$  at time instant  $t$  can be calculated by

$$\theta_p^{\text{walk-in}}(t) = a_{0,p}^{\text{walk}} \left( \sum_{m \in S_{\text{sta}}} \beta_{p,m}^{\text{station}}(t) \lambda_{j,m}^{\text{station}}(t) \right) + b_{0,p}^{\text{walk}}, \quad (4)$$

where  $a_{0,p}^{\text{walk}}$  and  $b_{0,p}^{\text{walk}}$  are coefficients that depend on the layout of the station, the walking distance, etc., and that can e.g. be determined based on historical data. The total walking time for passengers from the entrances to platform  $p$  of station  $j$  during the scheduling time period  $[t_0, t_{\text{end}}]$  can be calculated by

$$t_p^{\text{walk-in}} = \sum_{c=1}^{C_p} \theta_p^{\text{walk-in}}(t_{p,c}^{\text{station}}) \left( \sum_{m \in S_{\text{sta}}} \beta_{p,m}^{\text{station}}(t_{p,c}^{\text{station}}) \lambda_{j,m}^{\text{station}}(t_{p,c}^{\text{station}}) (t_{p,c+1}^{\text{station}} - t_{p,c}^{\text{station}}) \right), \quad (5)$$

where  $t_{p,c}^{\text{station}}$  is the time instant at which PWA constant functions  $\beta_{p,m}^{\text{station}}(\cdot)$  and/or  $\lambda_{j,m}(\cdot)$  change for the  $c$ th time. Note that  $t_{p,1}^{\text{station}} = t_0$  and  $t_{p,C_p}^{\text{station}} = t_{\text{end}}$ .

### 2.3.3. Passenger transfers triggered by arrival events

If a train arrives at a transfer station, there could be several possible routes for the onboard passengers to arrive at their destinations. They could choose to stay on the train or to get off the train and transfer to a train on another line. At transfer station  $j$ , the splitting rate of the passengers that are on board of train  $i$  and have destination  $m$  and that go to platform  $p'$  of station  $j$  can be denoted as  $\beta_{i,p',m}^{\text{train}}$  for  $p' \in P_j$ . The splitting rates for passengers that stay on train  $i$  are then described by  $\beta_{i,p,m}^{\text{train}}$  with  $m \neq j$ . For train  $i$  that stops at platform  $p$  of transfer station  $j$ , the sum of all the splitting rates has to be equal to 1, i.e.,

$$\sum_{p' \in P_j} \beta_{i,p',m}^{\text{train}} = 1, \quad \forall i \in S_{\text{tra}}, \forall m \in S_{\text{sta}}. \quad (6)$$

Note that the passengers with destination  $j$ , i.e., the ones for which  $m = j$ , will not choose to transfer to other platforms but they will exit the transit network at station  $j$ ; so if train  $i$  arrives at platform  $p$ , we set  $\beta_{i,p,j}^{\text{train}} = 1$  with the platform  $p$  at which train  $i$  arrives at for and  $\beta_{i,p',j}^{\text{train}} = 0$  with  $p' \in P_j \setminus \{p\}$ . The total walking time for passengers from platform  $p$  to exits of station  $j$  can be calculated as

$$t_p^{\text{walk-out}} = \sum_{i \in S_{\text{tra}}^p} a_{p,0}^{\text{walk}} n_{i,p,j}^{\text{alight}} + b_{p,0}^{\text{walk}}, \quad (7)$$

where  $S_{\text{tra}}^p$  is the set of trains that stop at platform  $p$  during the scheduling period  $[t_0, t_{\text{end}}]$ ,  $n_{i,p,j}^{\text{alight}}$  is the number of passengers who get off train  $i$ , have destination  $j$ , and exit the urban rail network from platform  $p$ . The coefficients  $a_{p,0}^{\text{walk}}$  and  $b_{p,0}^{\text{walk}}$  can be determined in a similar way as  $a_{0,p}^{\text{walk}}$  and  $b_{0,p}^{\text{walk}}$ .

The walking time for transfer passengers depends on the walking distance between two platforms and on the number of transfer passengers. In practice, the walking time could be distributed

as shown by the solid line in Figure 7. For the sake of simplicity, we approximate the relationship between the passenger walking time and number of transfer passengers by a rectangular signal as represented by the dashed line in Figure 7. Hence, we can calculate the average walking time of the transfer passengers from platform  $p$  to the other platforms  $p' \in P_j \setminus \{p\}$  as [Place Figure 7 about here]

$$\theta_{i,p,p'}^{\text{walk}} = a_{p,p'}^{\text{walk}} n_{i,p,p'}^{\text{transf}} + b_{p,p'}^{\text{walk}}, \quad \forall i \in S_{\text{tra}}, \forall p' \in P_j \setminus \{p\}, \quad (8)$$

where  $n_{i,p,p'}^{\text{transf}}$  is the number of transfer passengers from train  $i$  to platform  $p'$  of line  $\ell'$ ,  $a_{p,p'}^{\text{walk}}$  and  $b_{p,p'}^{\text{walk}}$  are the coefficients for the average walking time, which depend on the layout of the transfer station, the walking distance, etc., and which can e.g. be determined based on historical data. The total transfer time  $t_{i,p}^{\text{transf}}$  for transferring passengers getting off from train  $i$  is

$$t_{i,p}^{\text{transf}} = \sum_{p' \in P_j \setminus \{p\}} \theta_{i,p,p'}^{\text{walk}} n_{i,p,p'}^{\text{transf}}. \quad (9)$$

Similar as the average walking time, the duration time for the transfer process can be approximated using

$$\theta_{i,p,p'}^{\text{duration}} = a_{p,p'}^{\text{duration}} n_{i,p,p'}^{\text{transf}} + b_{p,p'}^{\text{duration}}, \quad \forall i \in S_{\text{tra}}, \forall p' \in P_j \setminus \{p\}. \quad (10)$$

Similar to  $a_{p,p'}^{\text{walk}}$  and  $b_{p,p'}^{\text{walk}}$ ,  $a_{p,p'}^{\text{duration}}$  and  $b_{p,p'}^{\text{duration}}$  can be determined based on historical data. The updates for the  $\lambda$ -profile due to passenger transfers can be described by a list of corner points:

$$\{(a_{i,j}, 0), (a_{i,j} + \theta_{i,p,p'}^{\text{walk}}, \Delta\lambda_{i,p,p'}), (a_{i,j} + \theta_{i,p,p'}^{\text{walk}} + \theta_{i,p,p'}^{\text{duration}}, 0)\}, \quad (11)$$

where  $\Delta\lambda_{i,p,p'}$  is calculated by

$$\Delta\lambda_{i,p,p'} = \frac{n_{i,p,p'}^{\text{transf}}}{\theta_{i,p,p'}^{\text{duration}}}. \quad (12)$$

#### 2.4. State updates of the event-driven model

When an event occurs, the state of the system should be updated and some other events may be triggered. For all the events occurring in the given system, the numbers of passengers waiting at platforms need to be updated. It is important to note that the passenger arrival rate stays the same between two subsequent events. Immediately before event  $e_k$  happens, the number of passengers  $w_{p_k,m}^{\text{wait,before}}(t_k)$  with destination  $m$  that are waiting at platform  $p_k$  is updated as follows (see Figure 8): [Place Figure 8 about here]

$$w_{p_k,m}^{\text{wait,before}}(t_k) = w_{p_k,m}^{\text{wait,after}}(t_{k'}) + \lambda_{p_k,m}(t_{k'})(t_k - t_{k'}), \quad (13)$$

where  $t_{k'}$  is the event time of the previous event  $e_{k'} = (t_{k'}, Y_{\text{type},k'}, i_{k'}, p_{k'})$  happening at platform  $p_k$  of line  $\ell_k$  (i.e.,  $p_{k'} = p_k$ ),  $w_{p_k,m}^{\text{wait,after}}(t_{k'})$  is the number of passengers at the platform immediately after event  $e_{k'}$ , and  $\lambda_{p_k,m}(t_{k'})(t_k - t_{k'})$  is the number of passengers that arrive at this platform between  $t_{k'}$  and  $t_k$ . The total number of waiting passengers  $w_{p_k}^{\text{wait,before}}(t_k)$  at platform  $p_k$  immediately before event  $e_k$  can be calculated as

$$w_{p_k}^{\text{wait,before}}(t_k) = \sum_{m \in S_{\text{sta}}} w_{p_k,m}^{\text{wait,before}}(t_k). \quad (14)$$

The waiting time of passengers at a platform is updated when an event occurs. We use  $t_{p_k}^{\text{wait}}(t_k)$  to denote the waiting time at the platform of the passengers that are at platform  $p_k$  when event  $e_k$  occurs, which can be calculated by

$$t_{p_k}^{\text{wait}}(t_k) = t_{p_k}^{\text{wait}}(t_{k'}) + \sum_{m \in \mathcal{S}_{\text{sta}}} \left( w_{p_k, m}^{\text{wait, after}}(t_{k'}) (t_k - t_{k'}) + \frac{1}{2} \lambda_{p_k, m}(t_{k'}) (t_k - t_{k'})^2 \right), \quad (15)$$

where  $t_{k'}$  is the event time of the previous event  $e_{k'}$  that occurred at platform  $p_k$ .

In general, the updates of other states and the triggered events caused by the current event depend on the event type of the current event. For  $\lambda$ -change events, only the number of waiting passengers at platforms and the waiting time of these passengers need to be updated. For departure events and arrival events, a detailed description of the updates of other states and the triggered events is given as follows.

#### 2.4.1. State updates and triggered events for departure events

When a departure event occurs, denoted as  $e_k = (d_{i_k, p_k}, 'd', i_k, p_k)$ , then train  $i_k$  departs from platform  $p_k$  at time  $d_{i_k, p_k}$ . The number of passengers boarding train  $i_k$  at platform  $p_k$  is equal to the minimum of the number of waiting passengers  $w_{p_k}^{\text{wait, before}}(t_k)$  and the remaining space  $n_{i_k, p_k}^{\text{remain}}$  on the train after the alighting process of passengers, i.e.

$$n_{i_k, p_k}^{\text{board}} = \min(n_{i_k, p_k}^{\text{remain}}, w_{p_k}^{\text{wait, before}}(t_k)). \quad (16)$$

The remaining space  $n_{i_k, p_k}^{\text{remain}}$  on train  $i_k$  for passengers is

$$n_{i_k, p_k}^{\text{remain}} = C_{\max, i_k} - n_{i_k, \text{p}^{\text{pla}}(p_k)} - n_{i_k, p_k}^{\text{alight}}, \quad (17)$$

where  $C_{\max, i_k}$  is the capacity of train  $i_k$ ,  $\text{p}^{\text{pla}}(p_k)$  is the predecessor platform<sup>1</sup> of platform  $p_k$ ,  $n_{i, p}$  is the number of passengers on train  $i$  when it departs from platform  $p$ , and  $n_{i, p}^{\text{alight}}$  is the number of passengers getting off train  $i$  at platform  $p$ . The calculation for  $n_{i_k, p_k}^{\text{alight}}$  will be given in (31) below.

At platform  $p_k$ , the number of passengers  $w_{p_k}^{\text{wait, after}}(t_k)$  who cannot get on train  $i_k$ , i.e., the number of passengers waiting at the platform immediately after the departure event  $e_k$  of train  $i_k$ , is

$$w_{p_k}^{\text{wait, after}}(t_k) = w_{p_k}^{\text{wait, before}}(t_k) - n_{i_k, p_k}^{\text{board}}. \quad (18)$$

In addition, if there are any passengers left by train  $i_k$ , i.e.,  $w_{p_k}^{\text{wait, after}}(t_k) > 0$ , we assume that the proportion of the passengers with different destinations with respect to the total number of waiting passengers does not change after the boarding process. This means that the passengers with different destinations have the same probability to be left by train  $i_k$ . So the number of passenger with destination  $m$  that are left by train  $i_k$  is proportional to the number of waiting passengers with destination  $m$ , i.e.,

$$w_{p_k, m}^{\text{wait, after}}(t_k) = w_{p_k}^{\text{wait, after}}(t_k) \frac{w_{p_k, m}^{\text{wait, before}}(t_k)}{w_{p_k}^{\text{wait, before}}(t_k)}. \quad (19)$$

<sup>1</sup>Recall that  $\text{p}^{\text{pla}}(p_k)$  is the previous platform on the line to which platform  $p_k$  belongs.

The number of passengers with destination  $m$  that board train  $i_k$  at platform  $p_k$  is

$$n_{i_k, p_k, m}^{\text{board}} = w_{p_k, m}^{\text{wait, before}}(t_k) - w_{p_k, m}^{\text{wait, after}}(t_k). \quad (20)$$

After the boarding process has completed, the number of passengers  $n_{i_k, p_k, m}^{\text{after}}$  with destination  $m$  that are on board of train  $i_k$  is updated as

$$n_{i_k, p_k, m}^{\text{after}} = n_{i_k, p_k, m}^{\text{before}} + n_{i_k, p_k, m}^{\text{board}}, \quad (21)$$

and the total number of passengers  $n_{i_k, p_k}^{\text{after}}$  on board of train  $i_k$  at platform  $p_k$  after the boarding process is

$$n_{i_k, p_k}^{\text{after}} = n_{i_k, p_k}^{\text{before}} + n_{i_k, p_k}^{\text{board}}, \quad (22)$$

where  $n_{i_k, p_k, m}^{\text{before}}$  and  $n_{i_k, p_k}^{\text{before}}$  are the number of passengers with destination  $m$  and the total number of passengers on board train  $i$  before the boarding process of passengers (see Section 2.4.2 below for more details).

The departure event  $e_k$  at platform  $p_k$  will generate an arrival event at the next platform of the line to which platform  $p_k$  belongs, which is described as follows:

$$(a_{i_k, \text{spla}(p_k)}, 'a', i_k, \text{spla}(p_k)),$$

where  $a_{i_k, \text{spla}(p_k)}$  is the arrival time of train  $i_k$  at platform  $\text{spla}(p_k)$ . The arrival time  $a_{i_k, \text{spla}(p_k)}$  can be calculated by

$$a_{i_k, \text{spla}(p_k)} = d_{i_k, p_k} + r_{i_k, p_k}, \quad (23)$$

where  $d_{i_k, j_k}$  is equal to  $t_k$  and  $r_{i_k, p_k}$  is the running time on the track segment between platform  $p_k$  and platform  $\text{spla}(p_k)$ . This arrival event should then be added to the global event list.

#### 2.4.2. State updates and triggered events for arrival events

When event  $e_k = (a_{i_k, p_k}, 'a', i_k, p_k)$  occurs, i.e., the arrival event of train  $i_k$  at platform  $p_k$ , the number of passengers  $w_{p_k}^{\text{wait, before}}(t_k)$  waiting at platform  $p_k$  immediately before this arrival event should be updated using (13) and (14).

The number of passengers getting off train  $i_k$  depends on platform  $p_k$ :

- If platform  $p_k$  is at the first station of the line, then there are no passengers getting off train  $i_k$ , i.e.

$$n_{i_k, p_k}^{\text{alight}} = 0. \quad (24)$$

In addition, the number of passengers  $n_{i_k, p_k, m}^{\text{before}}$  that have destination  $m$  and are on board of train  $i_k$  immediately before the boarding process is also equal to zero, i.e.

$$n_{i_k, p_k, m}^{\text{before}} = 0, \quad \forall m \in \mathcal{S}_{\text{sta}}. \quad (25)$$

The arrival event  $e_k$  at platform  $p_k$  will generate a departure event at this platform, which is described as follows:

$$(d_{i_k, p_k}, 'd', i_k, p_k), \quad (26)$$

where  $d_{i_k, p_k}$  is the departure time of train  $i_k$  at platform  $p_k$ .

- If the station  $j_k$  to which platform  $p_k$  belongs, is neither the first station, nor a transfer station, nor a terminal station, then the passengers with destination  $j_k$  will get off train  $i_k$ . The number of these passengers can be computed as follows:

$$n_{i_k, p_k}^{\text{alight}} = n_{i_k, \text{ppla}(p_k), j_k}^{\text{after}} \quad (27)$$

where  $n_{i_k, \text{ppla}(p_k), j_k}^{\text{after}}$  is the number of onboard passengers with destination  $j_k$  after the boarding process at predecessor platform  $\text{ppla}(p_k)$  has completed. Furthermore, we calculate the number of passengers  $n_{i_k, p_k, m}^{\text{before}}$  as follows:

$$n_{i_k, p_k, m}^{\text{before}} = n_{i_k, \text{ppla}(p_k), m}^{\text{after}}, \quad \forall m \in \mathcal{S}_{\text{sta}} \setminus \{j_k\}. \quad (28)$$

Therefore, the total number of passengers on board train  $i_k$  before the boarding process is

$$n_{i_k, p_k}^{\text{before}} = \sum_{m \in \mathcal{S}_{\text{sta}} \setminus \{j_k\}} n_{i_k, p_k, m}^{\text{before}}. \quad (29)$$

Moreover, the departure event given in (26) will be generated and included in the global event list.

- If the station  $j_k$  to which platform  $p_k$  belongs is a transfer station, then not only the passengers with destination  $j_k$  will get off train  $i_k$ , but the passengers with other destinations may also get off train  $i_k$ . The splitting rates for the passengers with destination  $m$  staying on or getting off train  $i_k$  are denoted as  $\beta_{i_k, p', m}^{\text{train}}$  for  $p' \in P_{j_k}$ . The number of passengers  $n_{i_k, p_k, m}^{\text{before}}$  that have destination  $m$  and are on board of train  $i_k$  immediately before the boarding process can be calculated by

$$n_{i_k, p_k, m}^{\text{before}} = \beta_{i_k, p_k, m}^{\text{train}} n_{i_k, \text{ppla}(p_k), m}^{\text{after}}, \quad \forall m \in \mathcal{S}_{\text{sta}}, \quad (30)$$

where  $n_{i_k, \text{ppla}(p_k), m}^{\text{after}}$  is the number of onboard passengers with destination  $m$  immediately after the boarding process at predecessor platform  $\text{ppla}(p_k)$  has completed.

As mentioned in Section 2.3.3, since the current station is station  $j_k$  the splitting rate  $\beta_{i_k, p_k, j_k}^{\text{train}}$  equals 1 for the passenger flow with destination  $j_k$ . All these passengers will get off the train and exit the network from station  $j_k$ . For the passengers with destination  $m$  with  $m \neq j_k$ , the passengers staying on the line to which platform  $p_k$  belongs, also stay on train  $i_k$ . Hence, the number of alighting passengers can be calculated by

$$n_{i_k, p_k}^{\text{alight}} = n_{i_k, \text{ppla}(p_k)}^{\text{after}} - \sum_{m \in \mathcal{S}_{\text{sta}} \setminus \{j_k\}} n_{i_k, p_k, m}^{\text{before}}. \quad (31)$$

where  $n_{i_k, \text{ppla}(p_k)}^{\text{after}}$  is in fact equal to the number of passengers on board of train  $i_k$  when it arrives at platform  $p_k$  and  $\sum_{m \in \mathcal{S}_{\text{sta}} \setminus \{j_k\}} n_{i_k, p_k, m}^{\text{before}}$  is the total number of passengers staying on train  $i_k$  after the alighting process. The number of transferring passengers  $n_{i_k, p_k, p', m}^{\text{transf}}$  that have destination  $m$  and transfer from platform  $p_k$  to some other platform  $p'$  can be calculated by

$$n_{i_k, p_k, p', m}^{\text{transf}} = \beta_{i_k, p', m}^{\text{train}} n_{i_k, \text{ppla}(p_k), m}^{\text{after}}, \quad \forall p' \in P_{j_k} \setminus \{p_k\}. \quad (32)$$

The total number of transfer passengers from train  $i_k$  is then

$$n_{i_k, p_k}^{\text{transf}} = \sum_{p' \in P_{j_k} \setminus \{p_k\}} \sum_{m \in S_{\text{sta}} \setminus \{j_k\}} n_{i_k, p_k, p', m}^{\text{transf}}. \quad (33)$$

The transfer process of passengers will trigger two  $\lambda$ -change events to increase and decrease the passenger arrival rates. These two events can be written as

$$(a_{i_k, p_k} + \theta_{i_k, p_k, p'}^{\text{walk}}, \lambda', -, p_k), \quad \forall p' \in P_{j_k} \setminus \{p_k\}, \quad (34)$$

$$(a_{i_k, p_k} + \theta_{i_k, p_k, p'}^{\text{walk}} + \theta_{i_k, p_k, p'}^{\text{duration}}, \lambda', -, p_k), \quad \forall p' \in P_{j_k} \setminus \{p_k\}, \quad (35)$$

where ‘-’ is a dummy place holder as there is no train included in these events. The above  $\lambda$ -change events should be added to the global event list. In addition, the departure event given in (26) will also be generated and included in the global event list.

- If the station  $j_k$  to which platform  $p_k$  belongs is a terminal station, there are no passengers getting on and getting off trains because we assume the terminal station to be a technical station. So the passenger arrival rate at a technical station is equal to zero and the number of passengers that have this technical station as destination is also equal to zero. Therefore, (24) and (25) hold for this case. The arrival event at a terminal station will also generate a departure event. However, this departure event will be different from that given in (26). Since the service number of a train will be augmented with the total number of trains in the network after its arrival at the terminal station, the generated departure event is as follows:

$$(d_{i_k + I_{\text{net}}, p_k}, d', i_k + I_{\text{net}}, p_k), \quad (36)$$

where  $I_{\text{net}}$  is the total number of trains in the network.

The passenger in-vehicle time for trains, denoted as  $t_{i_k, p_k}^{\text{in-vehicle}}$  should be updated when an arrival event happens. When arrival event  $e_k$  happens, the passenger in-vehicle time, including the running time of train  $i_k$  and the dwell time at platform  $p_k$ , can be calculated by

$$t_{i_k, p_k}^{\text{in-vehicle}} = n_{i_k, \text{ppla}(p_k)}^{\text{after}} r_{i_k, p_k} + (n_{i_k, \text{ppla}(p_k)}^{\text{after}} - n_{i_k, p_k}^{\text{alight}})(d_{i_k, p_k} - a_{i_k, p_k}), \quad (37)$$

where  $r_{i_k, p_k}$ ,  $d_{i_k, p_k}$ ,  $a_{i_k, p_k}$  are the running time, departure time, and arrival time of train  $i_k$  at platform  $p_k$ .

### 3. The train scheduling problem

Based on the model formulation of Section 2, we now formulate the train scheduling problem with consideration of the passenger demands. First, the performance criteria and the constraints of the train scheduling problem are formulated. Next, we also discuss how a rolling horizon approach can be adopted for the train scheduling problem and how the initial conditions are defined. Furthermore, some solution approaches for the train scheduling problem are presented.

### 3.1. Performance criteria

In this paper, we minimize the total travel time of all passengers and the energy consumption of the trains using a weighted sum strategy. Note, however, that we can accommodate other performance criteria all well, such as the operation cost of rail operators.

The total energy consumption for all  $I$  trains running with  $J$  stations can be formulated as

$$E_{\text{total}} = \sum_{i \in S_{\text{tra}}} \sum_{p \in S_{\text{pla}}} (E_{i,p}^{\text{acc}} + E_{i,p}^{\text{hold}} + E_{i,p}^{\text{dec}}), \quad (38)$$

where  $E_{i,p}^{\text{acc}}$ ,  $E_{i,p}^{\text{hold}}$ , and  $E_{i,p}^{\text{dec}}$  are the energy consumption of the acceleration, speed-holding, and deceleration phase, respectively. The acceleration and deceleration of train  $i$  between platform  $p$  and  $s^{\text{pla}}(p)$  is denoted by  $a_{i,p}^{\text{acc}}$  and  $a_{i,p}^{\text{dec}}$ , respectively. Moreover, the operation time for the acceleration phase, the speed-holding phase, and the deceleration phase are denoted as  $t_{i,p}^{\text{acc}}$ ,  $t_{i,p}^{\text{hold}}$ , and  $t_{i,p}^{\text{dec}}$ , respectively. The energy consumption for the acceleration phase of train  $i$  from platform  $p$  to successor platform  $s^{\text{pla}}(p)$  is

$$E_{i,p}^{\text{acc}} = \int_0^{t_{i,p}^{\text{acc}}} \left( (m_{e,i} + n_{i,p}^{\text{after}} m_{\text{pa}}) \left( a_{i,p}^{\text{acc}} + k_{1i} + k_{2i}v(t) + g \sin(\theta_p) \right) + k_{3i}v^2(t) \right) v(t) dt, \quad (39)$$

where  $m_{e,i}$  is the mass of train  $i$  itself,  $m_{\text{pa}}$  is the mass of one passenger,  $m_{e,i} + n_{i,p}^{\text{after}} m_{\text{pa}}$  is the mass of train  $i$  and the passengers on-board of train  $i$  at platform  $p$ ,  $k_{1i}$ ,  $k_{2i}$ , and  $k_{3i}$  are the resistance coefficients of train  $i$ ,  $v(t)$  is equal to  $a_{i,p}^{\text{acc}}t$ , and  $\theta_p$  is the average gradient between platform  $p$  and the successor  $s^{\text{pla}}(p)$ . The energy consumption for the speed holding phase of train  $i$  from platform  $p$  to platform  $s^{\text{pla}}(p)$  is

$$E_{i,p}^{\text{hold}} = \int_{t_{i,p}^{\text{acc}}}^{t_{i,p}^{\text{acc}} + t_{i,p}^{\text{hold}}} \left( (m_{e,i} + n_{i,p}^{\text{after}} m_{\text{pa}}) (k_{1i} + k_{2i}v_{i,j} + g \sin(\theta_p)) + k_{3i}v_{i,p}^2 \right) v_{i,p} dt. \quad (40)$$

In the deceleration phase, electric motors work as electric generators to generate energy for the urban rail transit system. So The energy consumption of the deceleration phase of train  $i$  on the segment between platform  $p$  and  $s^{\text{pla}}(p)$  may become negative in the braking process of trains, which is calculated by

$$E_{i,p}^{\text{dec}} = \eta_{i,p} \int_{t_{i,p}^{\text{acc}} + t_{i,p}^{\text{hold}}}^{t_{i,p}^{\text{dec}}} \left( (m_{e,i} + n_{i,p}^{\text{after}} m_{\text{pa}}) \left( a_{i,p}^{\text{dec}} + k_{1i} + k_{2i}v(t) + g \sin(\theta_p) \right) + k_{3i}v_{i,p}^2 \right) v_{i,p} dt, \quad (41)$$

where  $\eta_{i,p}$  is the energy recovery rate in the deceleration phase of train  $i$  on segment between platform  $p$  and  $s^{\text{pla}}(p)$ .

The total travel time of all passengers includes the passenger waiting time, the passenger in-vehicle time, and the passenger transfer time

$$t_{\text{total}} = \sum_{p \in S_{\text{pla}}} \left( t_p^{\text{wait}} + t_p^{\text{walk-in}} + t_p^{\text{walk-out}} \right) + \sum_{i \in S_{\text{tra}}} \sum_{p \in S_{\text{pla}}} \left( t_{i,p}^{\text{transf}} + t_{i,p}^{\text{in-vehicle}} \right), \quad (42)$$

where  $t_p^{\text{wait}}$ ,  $t_p^{\text{walk-in}}$ ,  $t_p^{\text{walk-out}}$ ,  $t_{i,p}^{\text{transf}}$ , and  $t_{i,p}^{\text{in-vehicle}}$  can be calculated by (15), (5), (7), (9), and (37).

In order to make the trains transport as many passengers as possible, we add a penalty term for the waiting time of the passengers left by the last train during the scheduling period:

$$f_{\text{penalty}} = \sum_{p \in S_{\text{pla}}} \left( t_p^{\text{wait}}(t_{\text{end}}) - t_p^{\text{wait}}(t_{k'}) \right), \quad (43)$$

where  $t_{\text{end}}$  is the end time of this scheduling period and  $t_{k'}$  is the time instant of the last departure event on platform  $p$ .

We apply a weighted sum strategy to solve the multi-objective train scheduling problem, i.e., we consider

$$f_{\text{opt}} = \frac{E_{\text{total}}}{E_{\text{total,nom}}} + \zeta_1 \frac{t_{\text{total}}}{t_{\text{total,nom}}} + \zeta_2 \frac{f_{\text{penalty}}}{f_{\text{penalty,nom}}}, \quad (44)$$

where  $\zeta_1$  and  $\zeta_2$  are non-negative weights, and the normalization factors  $E_{\text{total,nom}}$ ,  $t_{\text{total,nom}}$ , and  $f_{\text{penalty,nom}}$  are the nominal values of the total energy consumption, the total travel time of passengers, and the waiting time of the passengers left by the last train on each line, respectively. These nominal values can e.g. be determined using some typical feasible schedules.

### 3.2. Constraints

The constraints of the train scheduling problem consist of the event time constraints, the headway constraints, the passenger flow constraints, and the train capacity constraints. The passenger flow and train capacity constraints are (2)-(3), (6), (13)-(14), (16)-(22), and (24)-(33) as given in Section 2. The scheduling period in this paper is denoted as  $[t_0, t_{\text{end}}]$ ; so the event times  $t_k$  of the controlled events, i.e., the departure events and the arrival events, should satisfy

$$t_0 \leq t_k \leq t_{\text{end}}, \quad (45)$$

The event times of the autonomous events, i.e., the  $\lambda$ -change events, could be larger than  $t_{\text{end}}$ , in which case they will be handled in the next scheduling period. In addition, the event times of the departure events and the arrival events should satisfy the operational constraints as follows. For an arrival event  $e_k$ , the arrival time  $a_{i_k, p_k}$  of train  $i_k$  at platform  $p_k$  should satisfy the headway constraints:

$$a_{i_k, p_k} - d_{\text{ptra}(i_k), p_k} \geq h_{p_k, \text{min}}, \quad (46)$$

where  $d_{\text{ptra}(i_k), p_k}$  is the departure time of the previous train at platform  $p_k$  and  $h_{p_k, \text{min}}$  is the minimum headway at platform  $p_k$  to ensure the safe operation of trains. Note that since all the platforms are uniquely defined, the turnaround operation is equivalent to the operation to go from one platform to another platform. Hence, the constraint caused by the turnaround operation for a cyclic line can be included in the headway constraints. For a transversal line, the turnaround constraints can be written as follows:

$$d_{\text{spla}(i_k), p_k} - d_{i_k, p_k} \geq h_{\text{min,turn}}, \quad (47)$$

where  $p_k$  is the platform where the turnaround operation starts, platform  $p_k$  and successor platform  $\text{spla}(p_k)$  are at the same station, and  $h_{\text{min,turn}}$  is the minimum turnaround headway. Furthermore, for a departure event  $e_k$ , the departure time  $d_{i_k, p_k}$  of train  $i_k$  at platform  $p_k$ , should satisfy

$$\begin{aligned} d_{i_k, p_k} &\geq a_{i_k, p_k} + \tau_{i_k, p_k, \text{min}}, \\ d_{i_k, p_k} &\leq a_{i_k, p_k} + \tau_{i_k, p_k, \text{max}}, \end{aligned} \quad (48)$$

where  $\tau_{i_k, p_k, \text{min}}$  and  $\tau_{i_k, p_k, \text{max}}$  are the minimal and maximal dwell time for train  $i_k$  at platform  $p_k$ . The minimal dwell time is affected by the number of passengers getting off and getting on the train, which can be calculated as

$$\tau_{i_k, p_k, \text{min}} = \min \left( \tilde{\tau}_{\text{min}}, \alpha_{1,d} + \alpha_{2,d} n_{i_k, p_k}^{\text{alight}} + \alpha_{3,d} n_{i_k, p_k}^{\text{board}} + \alpha_{4,d} \left( \frac{w_{p_k}^{\text{wait}}(t_k)}{n^{\text{door}}} \right)^3 n_{i_k, p_k}^{\text{board}} \right), \quad (49)$$

where  $\tilde{\tau}_{\min}$  is the minimum dwell time predefined by railway operator,  $\alpha_{1,d}$ ,  $\alpha_{2,d}$ ,  $\alpha_{3,d}$ , and  $\alpha_{4,d}$  are coefficients that can e.g. be estimated based on historical data,  $n^{\text{door}}$  is the number of doors of the train,  $n_{i_k,p_k}^{\text{alight}} \in [0, C_{\max,i_k}]$  is the number of passengers that get off train  $i_k$  at platform  $p_k$ ,  $n_{i_k,p_k}^{\text{board}} \in [0, C_{\max,i_k}]$  is the number of passengers that board train  $i_k$  at platform  $p_k$ ,  $w_{i_k,p_k}^{\text{wait}} \geq 0$  is the number of passengers waiting for train  $i_k$  at platform  $p_k$ , and  $w_{i_k,p_k}^{\text{wait}}/n^{\text{door}}$  is the number of passengers waiting at each door. Note that Since we do not consider the capacity of the platform in this paper, we do not have an upper bound for  $w_{i_k,p_k}^{\text{wait}}$ . Moreover, the departure time  $d_{i_k,p_k}$  should satisfy the headway constraint as follows:

$$d_{i_k,p_k} - d_{p^{\text{tra}}(i_k),p_k} \leq h_{p_k,\max}, \quad (50)$$

where  $h_{p_k,\max}$  is the maximum departure-departure headway between trains at platform  $p_k$ , which is introduced to limit passenger dissatisfaction.

The running time  $r_{i,p}$  should satisfy

$$r_{i_k,p_k,\min} \leq r_{i_k,p_k} \leq r_{i_k,p_k,\max}, \quad (51)$$

where  $r_{i_k,p_k,\min}$  and  $r_{i_k,p_k,\max}$  are the minimal and maximal running time of train  $i_k$  between platform  $p_k$  and successor platform  $s^{\text{pla}}(p_k)$ , respectively. The minimum running time is limited by the train characteristics and the condition of the line. The maximum running time is introduced to limit passenger dissatisfaction since if trains run too slow, the passengers may complain. The maximum running time could be decided based on a passenger survey or on the expertise of the railway operator.

### 3.3. Rolling horizon approach and initial conditions

In this section, we discuss the rolling horizon approach in detail and we define the corresponding initial conditions.

Since passenger demands vary with time in a daily operation, the train scheduling problem can be solved in a rolling horizon way, by solving the scheduling problem, e.g., every half an hour, so as to adapt the train schedule to passenger demands in real time. This works as follows. First, the train scheduling problem is solved for some period  $[t_0, t_{\text{end}}]$  and the trains will be operated according to the resulting optimal schedule. After some period of time  $t_p$ , e.g., half an hour, we will run the optimization process again, but now for the period  $[t_0 + t_p, t_{\text{end}} + t_p]$  using the known, measured, or estimated states of the system at time  $t_0 + t_p$ . We compute the new optimal schedule and execute it for the next  $t_p$  time units, and then the whole process is repeated again for the period  $[t_0 + 2t_p, t_{\text{end}} + 2t_p]$  and so on, until the end of the daily operation of the urban rail transit system.

When solving the train scheduling problem in a rolling horizon way, some of the variables will no longer be free variables but will have fixed, known values. Assuming that  $t_0$  is the start time instant of the scheduling period, the number of passengers on all trains and the number of passengers waiting at all platforms at time  $t_0$  are known. we now discuss the fixed departure times and arrival times for an urban rail line:

- If train  $i$  is in the terminal station at time  $t_0$ , i.e., the arrival time  $a_{i-I_{\text{net}},0}$  of train  $i - I_{\text{net}}$  at the terminal station will be a known value with  $a_{i-I_{\text{net}},0} < t_0$ . So  $a_{i-I_{\text{net}},0}$  is no longer an unknown variable.

- If train  $i$  is at a platform of a station at time  $t_0$ , we use  $p_{i,t_0}$  to denote that platform. The arrival time  $a_{i,p_{i,t_0}}$  of train  $i$  at platform  $p_{i,t_0}$  is known. In addition, the departure times, the arrival times, and the running times of train  $i$  before platform  $p_{i,t_0}$  are also known.
- If train  $i$  is running on a segment at  $t_0$ , we use  $p_{i,t_0}$  to denote the segment on which train  $i$  is running on at  $t_0$ . Note that we use the platform index  $p$  to denote the segment between platform  $p$  and  $s^{\text{pla}}(p)$ . The departure time  $d_{i,p_{i,t_0}}$  of train  $i$  at platform  $p_{i,t_0}$  has a known value with  $d_{i,p_{i,t_0}} < t_0$ . In addition, all the departure times, arrival times, and running times before segment  $p_{i,t_0}$  are known. Furthermore, the running time  $r_{i,p_{i,t_0}}$  on segment  $p_{i,t_0}$  is also fixed since we assume that the schedule of a train can only be changed at platforms. Therefore, the arrival time of train  $i$  at platform  $s^{\text{pla}}(p_{i,t_0})$  is also known.

### 3.4. Solution approaches

The train scheduling problem for an urban rail transit network is a non-smooth non-convex programming problem with objective function (44) and constraints (2)-(3), (6), (13)-(14), (16)-(22), (24)-(33), and (45)-(51), where the non-smoothness is caused by the min function in (16) and (49), and the non-convexity is due to the nonlinear non-convex objective function and the non-convex set defined by constraints. Several approaches, e.g., pattern search (Hooke and Jeeves, 1961), sequential quadratic programming<sup>2</sup>(SQP) (Boggs and Tolle, 1995), mixed-integer (non)linear programming (Wang et al., 2013), iterative convex programming (Wang et al., 2015), and evolutionary algorithms (Bocharnikov et al., 2007; Ding et al., 2009; Yang et al., 2013) can be applied to solve this train scheduling problem.

## 4. Case study

In this section, we illustrate the proposed event-driven model for urban rail transit networks via a case study. The performance of the schedules obtained by SQP and a genetic algorithm based on this event-driven model is compared with that of a fixed headway schedule.

### 4.1. Set-up

A small network with two cyclic lines as shown in Figure 9 is considered in this case study. **[Place Figure 9 about here]** Line 1, represented by the solid line, has 1 terminal station (station 1) and 5 normal stations; line 2, represented by the dashed line, has 1 terminal station (station 7) and 7 normal stations. The line data of these two lines are given in Table 2. **[Place Table 2 about here]** The minimum running time in Table 2 is calculated by taking a fixed acceleration of  $0.8 \text{ m/s}^2$  and a fixed deceleration of  $-0.8 \text{ m/s}^2$ ; furthermore, when calculating the minimum running time the trains are assumed to run at the maximum speed of  $22.2 \text{ m/s}$  during the holding phase. The maximum running time is assumed to be  $r_{i,j,\text{max}} = \zeta r_{i,j,\text{min}}$ , where we have chosen  $\zeta$  as 1.2 to ensure that the passengers do not complain that the train is too slow. The weight variables  $\zeta_1$  and  $\zeta_2$  given in (44) are chosen as 2 and 3, respectively. For each cyclic line, there are 5 physical trains and the number of train services considered in the train scheduling problem

---

<sup>2</sup>When the SQP algorithm is applied to a nonlinear programming problem with a non-differentiable objective function, it might get stuck in a local solution. In the nonlinear programming problem proposed in this paper, the minimum value of the objective function is usually not obtained at the points where the objective function is non-differentiable, so in practice the SQP algorithm will jump over the points in which the function is non differentiable. Therefore, the SQP approach with multiple initial points does work well in this case.

is taken as 7. The parameters of trains and passengers are chosen as in Table 3. [Place Table 3 about here] The passenger arrival rates at stations are given in Table 4, where the passenger arrival rates are piecewise constant functions and the passenger arrival rates at terminals, i.e., station 1 and station 7, are equal to 0. In addition, since we only consider one direction of the cyclic lines, no passengers are arriving at the last stations of these two lines, i.e., station 6 and station 12. [Place Table 4 about here]

**Remark.** *Note that the event-driven model presented in this paper can also be used to model cyclic lines. More specifically, for the case study example without terminal stations 1 and 7 and with non-zero passenger arrival rates at stations 6 and 12, the event-driven model formulated in Section 2 could be easily modified to present this case and the state updates of the arrival events at the first and last platform of the line could be changed accordingly.*

At time  $t_0$  (chosen as 2500 s), the initial states of trains for line 1 are as follows: train 1 and train 2 are running from station 4 and 2 to station 5 and station 3, respectively, and their arrival times are fixed, at 2530 s and 2550 s respectively. The number of passengers on train 1 and 2 at time  $t_0$  is given in Table 5 and the numbers of passengers waiting at the platforms of line 1 are shown in Table 6. [Place Table 5 about here] [Place Table 6 about here] For line 2, the initial states at time  $t_0$  are as follows: train 6, 7, and 8 are running from station 11, 9, and 8 to station 5, 10, and 3, respectively and their arrival times are 2520 s, 2540 s, and 2560 s. The number of passengers on these trains at  $t_0$  is given in Table 5 and the numbers of passengers waiting at platforms of line 2 are shown in Table 6. In addition, there are 3 and 2 trains stopped at terminal stations 1 and 7, respectively. We choose the end of the scheduling period  $t_{\text{end}}$  as 5000 s, where the schedule of 7 train services will be optimized for each line. To limit passenger dissatisfaction, the maximum departure-departure headway at stations is chosen as 400 s. The nominal values for the total travel time, the energy consumption, and the waiting time for the passengers who did not travel in the scheduling period are calculated based on a random feasible schedule, which yields  $1.454 \cdot 10^7$  s,  $3.436 \cdot 10^9$  J, and  $7.434 \cdot 10^6$  s respectively.

The model formulation in Section 2 distinguishes between the splitting rates  $\beta_{p,m}^{\text{station}}$  of the passengers just entering the rail network and the splitting rates  $\beta_{i,p,m}^{\text{train}}$  of the passenger arriving at the transfer stations by trains. For this case study, we simplify the train scheduling model by making<sup>3</sup>  $\beta_{i,p,m}^{\text{train}}$  equal to  $\beta_{p,m}^{\text{station}}(a_{i,j})$  with  $a_{i,j}$  the arrival time of train  $i$  at station  $j$ ; this reduces the number of decision variables, especially for cases with a large number of trains. Our approach is not suitable at all if the number of variables is large. In addition, the walk-in and walk-out time for passengers in (42) is also taken as zero for the sake of simplicity. Two solution approaches are adopted to solve the train scheduling problem, i.e., a multistart SQP approach and a genetic algorithm. The SQP method as implemented by the `fmincon` function of the Matlab optimization toolbox is employed and we choose 10 feasible initial points to solve the optimization problem. For the genetic algorithm, the `ga` function of the global optimization toolbox of Matlab is used.

#### 4.2. Results

In order to compare the performance of the schedules obtained by the SQP approach and the genetic algorithm, two reference schedules with a fixed departure headway are considered. In fixed headway schedule 1, dwell times at stations have the same value, i.e., 60 s, and the fixed

---

<sup>3</sup>Recall that the splitting rates  $\beta_{i,p,m}^{\text{train}}$  for trains are numbers and that the splitting rates  $\beta_{p,m}^{\text{station}}$  are piecewise constant functions that have the time as their argument.

headways between trains for line 1 and line 2 are 340.0 s and 382.6 s, respectively. In fixed headway schedule 2, dwell times at normal stations and transfer stations are 60 s and 90 s, respectively. The headways between trains for line 1 and line 2 are 327.7 s and 368.0 s respectively in fixed headway schedule 2. The headways of the reference schedule are obtained by minimizing the objective function (44). The train schedules obtained by the fixed-headway approach, the SQP method, and genetic algorithm are given in Figure 10. [\[Place Figure 10 about here\]](#) The train schedules with fixed headway for line 1 and line 2 are given in Figures 10a and 10b. The train schedules obtained by the SQP method for line 1 and line 2 are given in Figures 10c and 10d. Moreover, the train schedules obtained by the genetic algorithm for line 1 and line 2 are given in Figures 10e and 10f. In addition, the number of passengers on board of trains for line 1 and line 2 obtained by the different solution approaches is given in Figure 11. Since there are no passenger arrivals at terminal stations (station 1 and station 7 for line 1 and line 2), the number of passengers on board of trains should be equal to 0 when trains depart or arrive at terminal stations, which is illustrated in Figure 11. For example, Figure 11c shows that when train 3 departs from station 2, the number of onboard passengers has already reached the maximum capacity, i.e., 1500 passengers. Similarly, train 2 and train 4 reach their maximum capacity at station 4 and station 3, respectively. [\[Place Figure 11 about here\]](#)

Station 3 and station 5 are transfer stations in the small rail network shown in Figure 9. Some passengers need to transfer to arrive at their destinations, e.g., passengers that enter the network at station 2 but have destination 9, 10, or 11 need to transfer at station 3. The number of transfer passengers at stations 3 and 5 is given in Figure 12. [\[Place Figure 12 about here\]](#) For the platform at transfer station 3 of line 1, the number of onboard passengers with different destinations is shown in Figure 12a, where the number of onboard passengers with destination 1, 2, 7, 8 is equal to 0. The passengers with destination 3 will get off the train at this station and the passengers with destination 9, 10, and 11 will also get off the train and transfer to line 2. The other passengers will stay on board. It is noted that the passengers with destination 12 choose to stay on the train instead of transferring to line 2 at station 3. This is because these passengers can also transfer at station 5, as shown in Figure 12b, and this will lead to a shorter travel time. Furthermore, the number of passengers at station 3 for train 1 is 0 in Figure 12a, since train 1 has already passed station 3 at time  $t_0$ . Similarly, the number of onboard passengers with different destinations at station 3 and 5 of line 2 is shown in Figures 12c and 12d.

A comparison of the performance of the three approaches, i.e., the fixed-headway approach, the SQP method and the genetic algorithm, is illustrated in Table 7, where the values of the objective function, the computation time of the solution approaches, the energy consumption of trains, the number and the travel time of the passengers that finished their trip, and the number and the waiting time of the passengers that did not travel are listed. [\[Place Table 7 about here\]](#) Additionally, the variance of the solutions obtained by 10 runs of different solution approaches are provided via the standard deviation ( $\sigma$ ) and the maximum and minimum values. For the computation for these three approaches we have used Matlab on a 64-bit linux operation system running on 1.8 GHz Intel Core2 Duo CPU. It is observed that the computation time of the fixed-headway approach is the smallest and that the SQP method yields the best performance among these three approaches for this case study. In particular, the objective value of the SQP method is about 10% smaller than that of the schedule with fixed headway and it is about 4% smaller than that obtained by the genetic algorithm. The train schedule obtained by the SQP method has a lower energy consumption and more passengers arrive at their destination within the period  $[t_0, t_{\text{end}}]$ . However, the travel time for passengers that finished their trip obtained by the genetic algorithm is smaller than that obtained within the period  $[t_0, t_{\text{end}}]$  by the SQP method. The reason

for this is that the number of passengers that finished their trips in the schedule obtained by the genetic algorithm is smaller than that of the SQP method and the fixed-headway approach. In addition, the number of passengers that did not finish their trip at  $t_{\text{end}}$  of the SQP approach is the smallest among these three approaches. These passengers still left at stations at  $t_{\text{end}}$  will be picked up by the train services of the next scheduling period.

A simple result of the train scheduling problem for a longer scheduling period, i.e., 3600 s, is also given in this paper, where train schedules of the fixed-headway approach and the SQP approach are given in Figure 13 and their performance comparison is given in Table 8. [Place Table 8 about here] As we can observe from Figure 13, the headway between trains in the schedule obtained from SQP approach is changing with the number of waiting passengers and the passenger arrival rates at stations. The passenger demand-oriented train schedule yields a better performance than the fixed headway train schedule (see Table 8). So a similar conclusion to the one presented above can also be obtained via train scheduling over the longer scheduling period.

[Place Figure 13 about here]

#### 4.3. Discussion

As we can observe for Figure 10, there is a small difference for the end times of the last train for the different approaches, which is normal because we do not set the end times of trains as hard constraints but optimize them through the objective function. Based on the results of the fixed-headway approach 1 and the SQP method, the end times of the last train are close to each other, e.g., the end times in Figure 10a and Figure 10c are 4989 s and 4967 s, respectively. From a theoretical point of view, the non-fixed headway schedule is more flexible than the fixed headway schedule especially when the passenger demands vary during the scheduling period because the headway of trains can become smaller so as to reduce the waiting time of passengers if many passengers are waiting at a certain station, and vice versa. In addition, if many passengers are on board of trains, the running time can become smaller to reduce the passenger onboard time. However, the smaller running time would increase the energy consumption. So the running times are optimized to achieve a good trade-off between energy consumption and the passenger travel time. Therefore, the flexibility of the non-fixed headway schedule will benefit both the passengers and the train operators.

In addition, the longer dwell times at some stations mean the running times will become shorter, while the shorter running times results in a larger energy consumption in general. The dwell times obtained by the event-driven model in Figure 10c-10f and Figure 13c-13d are calculated based on the number of alighting and boarding passengers. So in principle the dwell times obtained by the event-driven model can largely reduce the train delays introduced by the boarding and alighting process of passengers. In the fixed-headway approach, the dwell times at stations are predefined without considering the detailed information about the passenger flows. On the one hand, if the predefined dwell time is much longer than the time needed for the alighting and boarding process, then the total travel time will increase. This is the reason that the objective value of the fixed-headway approach 2 is higher than that of the fixed-headway approach 1. On the other hand, if there are many passengers getting on and getting off trains at a certain station, the predefined dwell times will not be enough for the alighting and boarding process of passengers, and train delays will be generated. So the longer dwell times in non-fixed headway schedules could reduce the delays caused by the passenger boarding and alighting process and could increase the robustness of the schedules.

The non-fixed schedules obtained by the SQP method and the genetic algorithm have a better performance than the fixed-headway approach because in this paper the objective function of the

train scheduling problem is a trade-off between the total passenger travel time and the energy consumption of trains. In addition, the passenger demands are different at different stations in the network and are changing with time. In order to reach a good balance between the energy consumption and the passenger travel time, the running times and headways are optimized in the train scheduling problem proposed in this paper. In addition, the dwell times of the trains in the non-fixed headway schedules also satisfy the passenger alighting and boarding constraints. So the passenger-demands-oriented train schedules obtained by the SQP method and genetic algorithm yield a better performance than the fixed headway schedule.

## 5. Conclusions and Future Work

In this paper, the train scheduling problem for an urban rail transit network has been investigated, where the origin-destination passenger demands are taken into account. We have developed an event-driven model with three types of events, i.e., departure events, arrival events, and passenger arrival rate change events, to describe the operation of the trains and the calculation of the number of passengers onboard of trains and waiting at platforms in the presence of the time-varying passenger demands. Furthermore, the passenger transfer process and the splitting of passenger flows at transfer stations are also included in the event-driven model. The corresponding scheduling problem is a non-convex nonlinear optimization problem, which can be solved using e.g. multi-start SQP or a genetic algorithm. For the given case study, the SQP method provides a better trade-off between control performance and computational complexity than the genetic algorithm. In addition, the performance of the non-fixed headway train schedule obtained by the SQP method is also better than the traditional fixed-headway train schedule; however, the computation time of the SQP method is much higher than that of the fixed-headway approach.

An extensive comparison and assessment of the SQP method, the genetic algorithm, and other solution approaches for different scenarios will be a topic for future work. Since the passenger demand in both directions is not symmetric in general, the optimal headways of these two directions are not equal. The event-driven model presented in this paper can easily be extended to the two-directional case. Hence, in our future work, we will consider the train scheduling problem for more complex cases with both directions of cyclic lines and transversal lines and also include the change of the number of physical trains in the problem formulation. Note that we have focused on the generation of non-fixed headway train schedules in this paper. However, in practice the uncertainties in passenger demands and in the operation of the trains are important for the feasibility and robustness of non-fixed headway train schedules. In future work, we will therefore include these uncertainties in the event-driven model, and perform extensive tests to assess the robustness of non-fixed headway train schedules and to compare their performance with that of fixed-headway train schedules. For the cases with multiple lines and a large number of stations and trains, multi-level and/or distributed approaches are expected to be required in order to model and solve the train scheduling problem efficiently (De Schutter et al., 2012; Shi and Zhou, 2015; Camponogara et al., 2002). Moreover, we will also investigate simplifications of the proposed model to obtain a good trade-off between modeling accuracy and computation speed.

## Acknowledgments

Research supported by the research foundation of Beijing Jiaotong University (Contract No. 2015RC003), the Beijing Municipal Commission of Education (Contract No. I15H100010), the State Key Laboratory of Rail Traffic Control and Safety (Contract No. RCS2013ZZ001), the Dutch Technology Foundation STW project “Model-predictive railway traffic management” (11025), and the European Union Seventh Framework Programme [FP7/ 2007-2013] under grant agreement no. 257462 HYCON2 Network of Excellence. The authors would also like to thank four anonymous reviewers for their constructive suggestions, which improved the content substantially.

## References

- Albrecht, T., 2009. Automated timetable design for demand-oriented service on suburban railways. *Public Transport* 1, 5–20.
- Assis, W., Milani, B., 2004. Generation of optimal schedules for metro lines using model predictive control. *Automatica* 40, 1397–1404.
- Barrena, E., Cana, D., Coelho, L., Laporte, G., 2014. Exact formulations and algorithm for the train scheduling problem with dynamic demand. *Computers & Operations Research* 44, 66–74.
- Bocharnikov, Y., Tobias, A., Roberts, C., Hillmansen, S., Goodman, C., 2007. Optimal driving strategy for traction energy saving on DC suburban railways. *IEE Proceedings - Electric Power Applications* 1, 675–682.
- Boggs, P., Tolle, J., 1995. Sequential quadratic programming. *Acta Numerica* 4, 1–51.
- Bussieck, M., Winter, T., Zimmermann, U., 1997. Discrete optimization in public rail transport. *Mathematical Programming* 79, 415–444.
- Camponogara, E., Jia, D., Krogh, B., Talukdar, S., 2002. Distributed model predictive control. *IEEE Control System Magazine* 22, 44–52.
- Cordeau, J., Toth, P., Vigo, D., 1998. A survey of optimization models for train routing and scheduling. *Transportation Science* 32, 380–420.
- Cury, J., Gomide, F., Mendes, M.J., 1980. A methodology for generation of optimal schedules for an underground railway systems. *IEEE Transactions on Automatic Control* 25, 217–222.
- D’Ariano, A., Pranzoand, M., Hansen, I., 2007. Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems* 8, 208–222.
- Ding, Y., Bai, Y., Liu, F., Mao, B., 2009. Simulation algorithm for energy-efficient train control under moving block system, in: *Proceedings of the 2009 World Congress on Computer Science and Information Engineering*, Los Angeles, CA, USA. pp. 498–502.
- Ghoseiri, K., Szidarovszky, F., Asgharpour, M., 2004. A multi-objective train scheduling model and solution. *Transportation Research Part B: Methodological* 38, 927–952.
- Higgins, A., Kozan, E., Ferreira, L., 1996. Optimal scheduling of trains on a single line track. *Transportation Research Part B: Methodological* 30, 147–161.
- Hooke, R., Jeeves, T., 1961. “Direct search” solution of numerical and statistical problems. *Journal of the Association for Computing Machinery* 8, 212–229.
- Kraay, D., Harker, P., Chen, B., 1991. Optimal pacing of trains in freight railroads: Model formulation and solution. *Operations Research* 39, 82–99.
- Kwan, C., Chang, C., 2005. Application of evolutionary algorithm on a transportation scheduling problem - The mass rapid transit, in: *Proceedings of the IEEE Congress on Evolutionary Computation*, Edinburgh, UK. pp. 987–994.
- Leemis, L., 1995. *Reliability: Probabilistic Models and Statistical Methods*. Prentice Hall, Englewood Cliffs, NJ, USA.
- Li, X., Lo, H., 2014. An energy-efficient scheduling and speed control approach for metro rail operations. *Transportation Research Part B: Methodological* 64, 73–89.
- Liebchen, C., 2006. *Periodic Timetable Optimization in Public Transport*. Ph.D. thesis, Technical University of Berlin, Berlin, Germany.
- Liebchen, C., 2008. The first optimized railway timetable in practice. *Transportation Science* 42, 420–435.
- Niu, H., Zhou, X., 2013. Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. *Transportation Research Part C: Emerging Technologies* 36, 212–230.
- Niu, H., Zhou, X., Gao, R., 2015. 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.

- Petersen, E., Taylor, A., Martland, C., 1986. An introduction to computer-assisted train dispatch. *Journal of Advanced Transportation* 20, 63–72.
- De Schutter, B., Ploeg, J., Baskar, L., Naus, G., Nijmeijer, H., 2012. Hierarchical, intelligent and automatic controls, in: Eskandarian, A. (Ed.), *Handbook of Intelligent Vehicles*. Springer, London, UK. chapter 5, pp. 81–116.
- Shi, T., Zhou, X., 2015. A mixed integer programming model for optimizing multi-level operations process in railroad yards. *Transportation Research Part B: Methodological* 80, 19–39.
- Su, S., Li, X., Tang, T., Gao, Z., 2013. A subway train timetable optimization approach based on energy-efficient operation strategy. *IEEE Transactions on Intelligent Transportation Systems* 14, 883–893.
- Szpigiel, B., 1972. Optimal train scheduling on a single line railway, in: *Proceedings of the International Conference on Operational Research*, Amsterdam, The Netherlands. pp. 344–351.
- Vázquez-Abad, F., Zubieta, L., 2005. Ghost simulation model for the optimization of an urban subway system. *Discrete Event Dynamic Systems* 15, 207–235.
- Wang, Y., Ning, B., Tang, T., van den Boom, T., De Schutter, B., 2015. Efficient real-time train scheduling for urban rail transit systems using iterative convex programming. *IEEE Transactions on Intelligent Transportation Systems* Accepted for publication.
- Wang, Y., De Schutter, B., van den Boom, T., Ning, B., 2013. Optimal trajectory planning for trains – A pseudospectral method and a mixed integer linear programming approach. *Transportation Research Part C: Emerging Technologies* 29, 97–114.
- Wong, R., Yuen, T., Fung, K., Leung, J., 2008. Optimizing timetable synchronization for rail mass transit. *Transportation Science* 42, 57–69.
- Wong, S., Tong, C., 1998. Estimation of time-dependent origin-destination matrices for transit networks. *Transportation Research Part B: Methodological* 32, 35–48.
- Yang, X., Li, X., Gao, Z., Wang, H., Tang, T., 2013. A cooperative scheduling model for timetable optimization in subway systems. *IEEE Transactions on Intelligent Transportation Systems* 14, 438–447.

### **List of Keywords**

train scheduling  
passenger demands  
event-driven  
urban rail transit network

### List of Table captions

Table 1: Parameters and variables

Table 2: Information of the two cyclic lines

Table 3: Parameters of the trains and the passengers

Table 4: Passenger arrival rates (passengers/s) in the small urban rail network (columns correspond to destinations)

Table 5: Number of passengers on board of trains at time  $t_0$  for the two cyclic lines

Table 6: Number of passengers waiting at platforms of the two cyclic lines

Table 7: Performance comparison of the SQP method and the genetic algorithm (Fixed headway 1 corresponds to the train schedule in which dwell times are the same, i.e., 60 s, for all stations and fixed headway 2 corresponds to the train schedule in which dwell times are longer, i.e., 90 s, at transfer stations.  $\sigma$  denotes the standard deviation of 10 runs.)

Table 8: Performance comparison of the fixed-headway approach and the SQP method for a longer scheduling period (i.e., 3600 s)

Table 1: Parameters and variables

|  |  |
|--|--|
| $\ell$   | index of lines;  |
| $i$  | index of train services;   |
| $j$  | index of stations;   |
| $p$  | index of platforms;  |
| $k$  | index of events;   |
| $L$  | total number of lines in the network;  |
| $J$  | total number of stations in the network;   |
| $I_\ell$   | total number of trains in line $\ell$ ;  |
| $S_{\text{in}}$                                  | set of lines;  |
| $S_{\text{sta}}$                                 | set of stations;   |
| $S_{\text{tra}}$                                 | set of train services;   |
| $S_p^{\text{tra}}$                               | set of train services that stop at platform $p$ ;  |
| $P_j$  | set of platforms at transfer station $j$ ;   |
| $s_{\text{pla}}(p)$                              | successor platform of platform $p$ ;   |
| $p_{\text{pla}}(p)$                              | predecessor platform of platform $p$ ;   |
| $s_{\text{tra}}(i)$                              | successor train of train $i$ ;   |
| $p_{\text{tra}}(i)$                              | predecessor train of train $i$ ;   |
| $t_0$  | starting time of the scheduling period;  |
| $t_{\text{end}}$                                 | end time of the scheduling period;   |
| $e_k$  | $k$ th event;  |
| $t_k$  | time instant at which event $e_k$ occurs;  |
| $Y_{\text{type},k}$                              | event type of $e_k$ ;  |
| $i_k$  | train index of $e_k$ ;   |
| $p_k$  | platform index of $e_k$ ;  |
| $\lambda_{j,m}^{\text{station}}$                 | passenger arrival rate at station $j$ of passengers with station $m$ as their destination;   |
| $\lambda_{p,m}$                                  | passenger arrival rate at platform $p$ of passengers with station $m$ as their destination;  |
| $\beta_{p,m}^{\text{station}}$                   | splitting rate of passengers that arrive at station $j$ , go to platform $p \in P_j$ , and have destination $m$ ;  |
| $\beta_{i,p,m}^{\text{train}}$                   | splitting rate of passengers that are on board of train $i$ at station $j$ and have destination $m$ , and go to platform $p \in P_j$ ;   |
| $a_{p,p'}^{\text{walk}}, b_{p,p'}^{\text{walk}}$ | coefficients for the average walking time from platform $p$ to platform $p'$ ; $p = 0$ and $p' = 0$ correspond to the entrance and exit of the station to which platform $p$ and $p'$ belong;  |
| $t_p^{\text{walk-in}}$                           | total walking time for passengers from entrances to platform $p$ ;   |
| $t_p^{\text{walk-out}}$                          | total walking time for passengers from platform $p$ to exits of the station;   |
| $t_{i,p}^{\text{transf}}$                        | total transfer time for transfer passengers getting off train $i$ at platform $p$ ;  |
| $t_p^{\text{wait}}$                              | waiting time of passengers at platform $p$ during the scheduling period;   |
| $t_{i,p}^{\text{in-vehicle}}$                    | passenger in-vehicle time of train $i$ including the running time from predecessor platform $p_{\text{pla}}(p)$ to $p$ and the dwell time at platform $p$ ;                                    |
| $n_{i,p,p'}^{\text{transf}}$                     | number of passengers that get off train $i$ and transfer from platform $p$ to $p'$ ;   |
| $n_{i,p}^{\text{transf}}$                        | number of transfer passengers that get off train $i$ at platform $p$ ;   |
| $n_{i,p}^{\text{alight}}$                        | number of passengers alighting from train $i$ at platform $p$ ;  |
| $n_{i,p,m}^{\text{alight}}$                      | number of passengers that have destination $m$ and get off train $i$ at platform $p$ ;   |
| $n_{i,p}^{\text{board}}$                         | number of passengers boarding train $i$ at platform $p$ ;  |
| $n_{i,p}^{\text{remain}}$                        | remaining space on train $i$ after the alighting process of passengers at platform $p$ ;   |
| $n_{i,p}^{\text{before}}$                        | number of passengers on board of train $i$ at platform $p$ before the start of the boarding process;   |
| $n_{i,p}^{\text{after}}$                         | number of passengers on board of train $i$ at platform $p$ after the boarding process has completed;   |
| $w_{p_k,m}^{\text{wait,before}}(t_k)$            | number of passengers that are waiting at platform $p_k$ and have destination $m$ immediately before event $e_k$ occurs, where $e_k$ is the event that occurs at time $t_k$ at platform $p_k$ ; |
| $w_{p_k,m}^{\text{wait,after}}(t_k)$             | number of passengers that are waiting at platform $p$ and have destination $m$ immediately after event $e_k$ occurs, where $e_k$ is the event that occurs at time $t_k$ at platform $p_k$ ;    |
| $d_{i,p}$  | departure time of train $i$ at platform $p$ ;  |
| $a_{i,p}$  | arrival time of train $i$ at platform $p$ ;  |
| $r_{i,p}$  | running time of train $i$ from predecessor platform $p_{\text{pla}}(p)$ to $p$ ;   |
| $\zeta_1, \zeta_2$                               | non-negative weights in the objective function.  |

Table 2: Information of the two cyclic lines

|                              |      |      |       |       |       |      |
|------------------------------|------|------|-------|-------|-------|------|
| Station number (Line 1)      | 1    | 2    | 3     | 4     | 5     | 6    |
| Distance to next station [m] | 700  | 1500 | 1700  | 2200  | 1900  | 800  |
| Minimal running time [s]     | 59.3 | 95.3 | 104.3 | 126.8 | 113.3 | 63.8 |

|                              |      |      |      |      |      |      |      |      |
|------------------------------|------|------|------|------|------|------|------|------|
| Station number (Line 2)      | 7    | 8    | 3    | 9    | 10   | 11   | 5    | 12   |
| Distance to next station [m] | 860  | 1400 | 1500 | 1300 | 1600 | 1200 | 1100 | 730  |
| Minimal running time [s]     | 66.5 | 90.8 | 95.3 | 86.3 | 99.8 | 81.8 | 77.3 | 66.5 |

Table 3: Parameters of the trains and the passengers

| Property                                  |                               | Symbol               | Value                 |
|---|-------------------------------|----------------------|-----------------------|
| Train mass                                | [kg]                          | $m_{e,i}$            | $199 \cdot 10^3$      |
| Mass of one passenger                     | [kg]                          | $m_p$                | 60                    |
| Capacity of trains                        | [passengers]                  | $C_{i,max}$          | 1500                  |
| Minimum dwell time                        | [s]                           | $\tilde{\tau}_{min}$ | 30                    |
| Maximum dwell time                        | [s]                           | $\tau_{max}$         | 150                   |
| Coefficients of the<br>minimal dwell time | [s]                           | $\alpha_{1,d}$       | 4.002                 |
|   | [s/passengers]                | $\alpha_{2,d}$       | 0.047                 |
|   | [s/passengers]                | $\alpha_{3,d}$       | 0.051                 |
|   | [s/passengers <sup>-4</sup> ] | $\alpha_{4,d}$       | $1.0 \cdot 10^{-6}$   |
| Coefficients of resistance                | [m/s <sup>2</sup> ]           | $k_{1i}$             | 0.012                 |
|   | [s <sup>-1</sup> ]            | $k_{2i}$             | $5.049 \cdot 10^{-4}$ |
|   | [m <sup>-1</sup> ]            | $k_{3i}$             | $2.053 \cdot 10^{-5}$ |

Table 4: Passenger arrival rates (passengers/s) in the small urban rail network (columns correspond to destinations)

| Station | Time period [s] | 1 | 2 | 3    | 4    | 5    | 6    | 7 | 8 | 9    | 10   | 11   | 12   |
|---------|-----------------|---|---|------|------|------|------|---|---|------|------|------|------|
| 1       | 2500 - 5000     | 0 | 0 | 0    | 0    | 0    | 0    | 0 | 0 | 0    | 0    | 0    | 0    |
| 2       | 2500 - 3000     | 0 | 0 | 0.48 | 0.64 | 0.32 | 0.32 | 0 | 0 | 0.64 | 0.48 | 0.32 | 0.32 |
|         | 3000 - 3600     | 0 | 0 | 0.32 | 0.48 | 0.32 | 0.16 | 0 | 0 | 0.48 | 0.40 | 0.35 | 0.16 |
|         | 3600 - 5000     | 0 | 0 | 0.32 | 0.32 | 0.32 | 0.32 | 0 | 0 | 0.48 | 0.38 | 0.56 | 0.16 |
| 3       | 2500 - 3100     | 0 | 0 | 0    | 0.32 | 0.32 | 0.16 | 0 | 0 | 0.32 | 0.34 | 0.29 | 0.32 |
|         | 3100 - 3700     | 0 | 0 | 0    | 0.48 | 0.32 | 0.32 | 0 | 0 | 0.32 | 0.22 | 0.42 | 0.32 |
|         | 3700 - 5000     | 0 | 0 | 0    | 0.16 | 0.64 | 0.16 | 0 | 0 | 0.48 | 0.38 | 0.26 | 0.16 |
| 4       | 2500 - 3250     | 0 | 0 | 0    | 0    | 0.45 | 0.30 | 0 | 0 | 0    | 0    | 0    | 0.30 |
|         | 3250 - 5000     | 0 | 0 | 0    | 0    | 0.53 | 0.38 | 0 | 0 | 0    | 0    | 0    | 0.38 |
| 5       | 2500 - 2850     | 0 | 0 | 0    | 0    | 0    | 0.60 | 0 | 0 | 0    | 0    | 0    | 0.30 |
|         | 2850 - 3390     | 0 | 0 | 0    | 0    | 0    | 0.75 | 0 | 0 | 0    | 0    | 0    | 0.30 |
|         | 3390 - 3830     | 0 | 0 | 0    | 0    | 0    | 0.60 | 0 | 0 | 0    | 0    | 0    | 0.30 |
|         | 3830 - 5000     | 0 | 0 | 0    | 0    | 0    | 0.75 | 0 | 0 | 0    | 0    | 0    | 0.30 |
| 6       | 2500 - 5000     | 0 | 0 | 0    | 0    | 0    | 0    | 0 | 0 | 0    | 0    | 0    | 0    |
| 7       | 2500 - 5000     | 0 | 0 | 0    | 0    | 0    | 0    | 0 | 0 | 0    | 0    | 0    | 0    |
| 8       | 2500 - 3100     | 0 | 0 | 0.12 | 0.24 | 0.36 | 0.12 | 0 | 0 | 0.24 | 0.29 | 0.22 | 0.24 |
|         | 3100 - 3700     | 0 | 0 | 0.12 | 0.24 | 0.60 | 0.12 | 0 | 0 | 0.24 | 0.34 | 0.19 | 0.24 |
|         | 3700 - 5000     | 0 | 0 | 0.12 | 0.24 | 0.36 | 0.12 | 0 | 0 | 0.24 | 0.29 | 0.29 | 0.24 |
| 9       | 2500 - 3100     | 0 | 0 | 0    | 0    | 0.12 | 0.24 | 0 | 0 | 0    | 0.14 | 0.19 | 0.24 |
|         | 3100 - 5000     | 0 | 0 | 0    | 0    | 0.12 | 0.36 | 0 | 0 | 0    | 0.29 | 0.31 | 0.36 |
| 10      | 2500 - 2900     | 0 | 0 | 0    | 0    | 0.15 | 0.15 | 0 | 0 | 0    | 0    | 0.21 | 0.15 |
|         | 2900 - 5000     | 0 | 0 | 0    | 0    | 0.15 | 0.15 | 0 | 0 | 0    | 0    | 0.24 | 0.21 |
| 11      | 2500 - 2900     | 0 | 0 | 0    | 0    | 0.24 | 0.20 | 0 | 0 | 0    | 0    | 0    | 0.24 |
|         | 2900 - 5000     | 0 | 0 | 0    | 0    | 0.36 | 0.32 | 0 | 0 | 0    | 0    | 0    | 0.28 |
| 12      | 2500 - 5000     | 0 | 0 | 0    | 0    | 0    | 0    | 0 | 0 | 0    | 0    | 0    | 0    |

Table 5: Number of passengers on board of trains at time  $t_0$  for the two cyclic lines

| Destination      | 1 | 2 | 3  | 4  | 5   | 6   | 7 | 8 | 9   | 10  | 11  | 12  | Total |
|------------------|---|---|----|----|-----|-----|---|---|-----|-----|-----|-----|-------|
| Train 1 (Line 1) | 0 | 0 | 0  | 0  | 130 | 150 | 0 | 0 | 0   | 0   | 0   | 80  | 360   |
| Train 2 (Line 1) | 0 | 0 | 80 | 70 | 90  | 50  | 0 | 0 | 60  | 140 | 130 | 80  | 700   |
| Train 6 (Line 2) | 0 | 0 | 0  | 0  | 230 | 280 | 0 | 0 | 0   | 0   | 0   | 170 | 680   |
| Train 7 (Line 2) | 0 | 0 | 0  | 0  | 160 | 180 | 0 | 0 | 0   | 120 | 134 | 162 | 756   |
| Train 8 (Line 2) | 0 | 0 | 79 | 98 | 100 | 130 | 0 | 0 | 120 | 80  | 120 | 80  | 787   |

Table 6: Number of passengers waiting at platforms of the two cyclic lines

| Destination | 1 | 2 | 3   | 4   | 5   | 6   | 7 | 8 | 9  | 10  | 11  | 12  | Total |
|-------------|---|---|-----|-----|-----|-----|---|---|----|-----|-----|-----|-------|
| Station 1   | 0 | 0 | 0   | 0   | 0   | 0   | 0 | 0 | 0  | 0   | 0   | 0   | 0     |
| Station 2   | 0 | 0 | 120 | 240 | 140 | 10  | 0 | 0 | 80 | 200 | 250 | 140 | 1180  |
| Station 3   | 0 | 0 | 0   | 150 | 200 | 130 | 0 | 0 | 0  | 0   | 0   | 90  | 570   |
| Station 4   | 0 | 0 | 0   | 0   | 200 | 230 | 0 | 0 | 0  | 0   | 0   | 120 | 550   |
| Station 5   | 0 | 0 | 0   | 0   | 0   | 210 | 0 | 0 | 0  | 0   | 0   | 0   | 210   |
| Station 6   | 0 | 0 | 0   | 0   | 0   | 0   | 0 | 0 | 0  | 0   | 0   | 0   | 0     |

| Destination | 1 | 2 | 3   | 4   | 5   | 6   | 7 | 8 | 9   | 10  | 11  | 12  | Total |
|-------------|---|---|-----|-----|-----|-----|---|---|-----|-----|-----|-----|-------|
| Station 7   | 0 | 0 | 0   | 0   | 0   | 0   | 0 | 0 | 0   | 0   | 0   | 0   | 0     |
| Station 8   | 0 | 0 | 150 | 120 | 80  | 100 | 0 | 0 | 180 | 100 | 140 | 140 | 1010  |
| Station 3   | 0 | 0 | 0   | 0   | 100 | 130 | 0 | 0 | 190 | 110 | 130 | 100 | 760   |
| Station 9   | 0 | 0 | 0   | 0   | 110 | 150 | 0 | 0 | 0   | 100 | 160 | 120 | 640   |
| Station 10  | 0 | 0 | 0   | 0   | 130 | 170 | 0 | 0 | 0   | 0   | 130 | 150 | 580   |
| Station 11  | 0 | 0 | 0   | 0   | 100 | 190 | 0 | 0 | 0   | 0   | 0   | 170 | 460   |
| Station 5   | 0 | 0 | 0   | 0   | 0   | 0   | 0 | 0 | 0   | 0   | 0   | 210 | 210   |
| Station 12  | 0 | 0 | 0   | 0   | 0   | 0   | 0 | 0 | 0   | 0   | 0   | 0   | 0     |

Table 7: Performance comparison of the SQP method and the genetic algorithm (Fixed headway 1 corresponds to the train schedule in which dwell times are the same, i.e., 60 s, for all stations and fixed headway 2 corresponds to the train schedule in which dwell times are longer, i.e., 90 s, at transfer stations.  $\sigma$  denotes the standard deviation of 10 runs.)

| Solution approaches  | Fixed headway 1                   | Fixed headway 2                   | SQP  | Genetic algorithm  |
|--|-----------------------------------|-----------------------------------|--|--|
| Objective value [-]<br>[ $\sigma$ , max, min]  | 4.754<br>-                        | 4.836<br>-                        | 4.413<br>[0.017, 4.395, 4.445]   | 4.554<br>[0.139, 4.447, 4.853]   |
| Computation time [s]<br>[ $\sigma$ , max, min]   | 14.381<br>[0.848, 16.609, 13.611] | 15.985<br>[0.811, 17.567, 14.896] | $1.128 \cdot 10^4$<br>[ $8.898 \cdot 10^2$ , $1.256 \cdot 10^4$ , $9.645 \cdot 10^3$ ] | $3.490 \cdot 10^4$<br>[ $1.423 \cdot 10^4$ , $5.765 \cdot 10^4$ , $9.476 \cdot 10^3$ ] |
| Energy consumption [J]<br>[ $\sigma$ , max, min]   | $3.370 \cdot 10^9$<br>-           | $3.370 \cdot 10^9$<br>-           | $2.734 \cdot 10^9$<br>[ $1.676 \cdot 10^7$ , $2.709 \cdot 10^9$ , $2.764 \cdot 10^9$ ] | $2.739 \cdot 10^9$<br>[ $3.423 \cdot 10^7$ , $2.698 \cdot 10^9$ , $2.823 \cdot 10^9$ ] |
| Number of passengers that finished their trip [passengers]<br>[ $\sigma$ , max, min]       | $2.471 \cdot 10^4$<br>-           | $2.496 \cdot 10^4$<br>-           | $2.558 \cdot 10^4$<br>[ $3.654 \cdot 10^2$ , $2.483 \cdot 10^4$ , $2.608 \cdot 10^4$ ] | $2.434 \cdot 10^4$<br>[ $1.235 \cdot 10^3$ , $2.227 \cdot 10^4$ , $2.532 \cdot 10^4$ ] |
| Number of passengers that did not finish their trip [passengers]<br>[ $\sigma$ , max, min] | $9.395 \cdot 10^3$<br>-           | $9.369 \cdot 10^3$<br>-           | $8.808 \cdot 10^3$<br>[74.72, $8.707 \cdot 10^3$ , $8.906 \cdot 10^3$ ]                | $9.531 \cdot 10^3$<br>[ $8.523 \cdot 10^2$ , $8.849 \cdot 10^3$ , $1.132 \cdot 10^4$ ] |
| Travel time for passengers that finished their trip [s]<br>[ $\sigma$ , max, min]          | $1.862 \cdot 10^7$<br>-           | $1.883 \cdot 10^7$<br>-           | $1.871 \cdot 10^7$<br>[ $2.172 \cdot 10^5$ , $1.852 \cdot 10^7$ , $1.924 \cdot 10^7$ ] | $1.885 \cdot 10^7$<br>[ $7.676 \cdot 10^5$ , $1.688 \cdot 10^7$ , $1.965 \cdot 10^7$ ] |
| Waiting time for passengers that did not travel [s]<br>[ $\sigma$ , max, min]              | $3.005 \cdot 10^6$<br>-           | $2.986 \cdot 10^6$<br>-           | $2.586 \cdot 10^6$<br>[ $4.922 \cdot 10^4$ , $2.464 \cdot 10^6$ , $2.635 \cdot 10^6$ ] | $2.887 \cdot 10^6$<br>[ $5.223 \cdot 10^5$ , $2.641 \cdot 10^6$ , $4.329 \cdot 10^6$ ] |

Table 8: Performance comparison of the fixed-headway approach and the SQP method for a longer scheduling period (i.e., 3600 s)

| Solution approaches  | Fixed headway      | SQP                |
|--|--------------------|--------------------|
| Objective value [-]  | 7.840              | 7.376              |
| Energy consumption [J]   | $4.710 \cdot 10^9$ | $4.018 \cdot 10^9$ |
| Number of passengers that finished their trip [passengers]       | $3.225 \cdot 10^4$ | $3.406 \cdot 10^4$ |
| Number of passengers that did not finish their trip [passengers] | $1.292 \cdot 10^4$ | $1.162 \cdot 10^4$ |
| Travel time for passengers that finished their trip [s]          | $2.931 \cdot 10^7$ | $3.273 \cdot 10^7$ |
| Waiting time for passengers that did not travel [s]              | $5.390 \cdot 10^6$ | $4.877 \cdot 10^6$ |

### List of Figure captions

- Figure 1: Model structure of the event-driven system
- Figure 2: Definition of known events and unknown events
- Figure 3:  $\lambda$ -profile query module
- Figure 4: Typical passenger arrival rate profile at a station for a working day
- Figure 5: Example of a base profile with corner points for passenger arrival rates
- Figure 6: Variables for the splitting of passenger flows and passenger transfers
- Figure 7: Typical walking time profile for the transfer passengers
- Figure 8: Variables for the passenger characteristics before and after the boarding process
- Figure 9: Layout of a small urban rail transit network
- Figure 10: Train schedules for line 1 and line 2 obtained by the fixed-headway approach, the SQP method, and the genetic algorithm
- Figure 11: Number of onboard passengers for line 1 and line 2 obtained by the fixed-headway approach, the SQP method, and the genetic algorithm
- Figure 12: Number of onboard passengers with different destinations at transfer station 3 and 5 obtained by the SQP method for line 1 and line 2
- Figure 13: Train schedules for line 1 and line 2 obtained by the fixed-headway approach and the SQP method over a longer scheduling period (i.e., 3600 s)

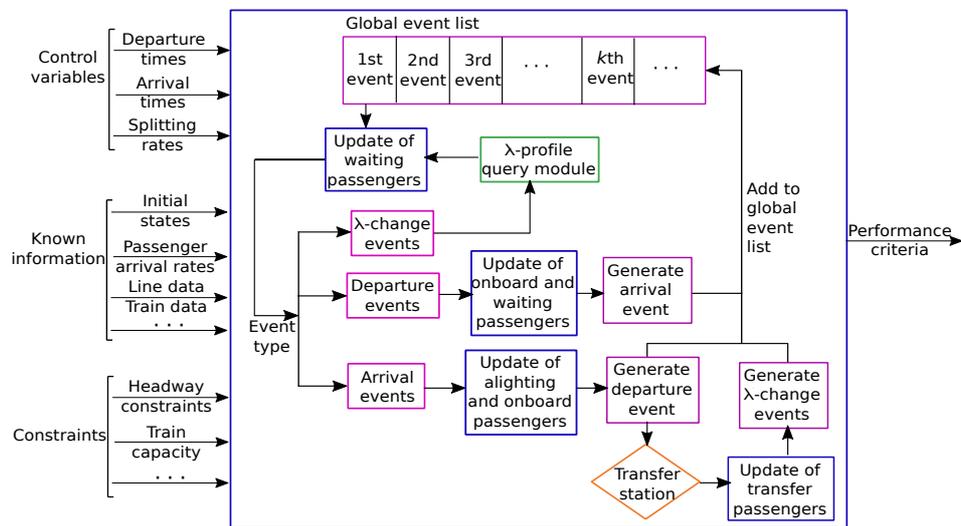


Figure 1: Model structure of the event-driven system

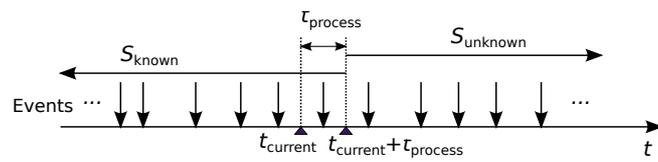


Figure 2: Definition of known events and unknown events

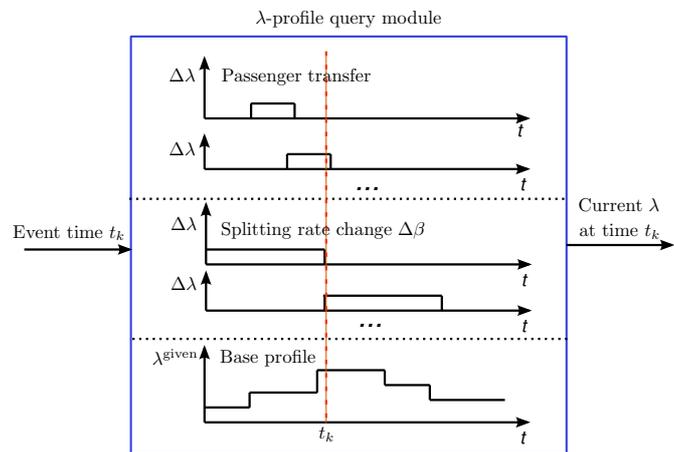


Figure 3:  $\lambda$ -profile query module

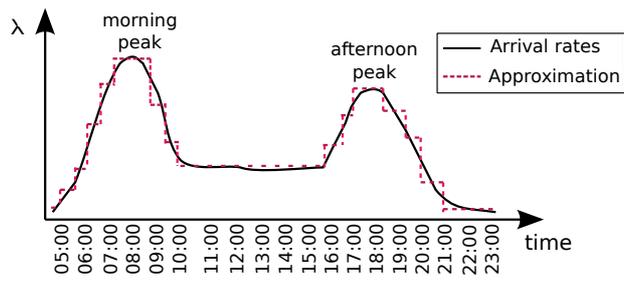


Figure 4: Typical passenger arrival rate profile at a station for a working day

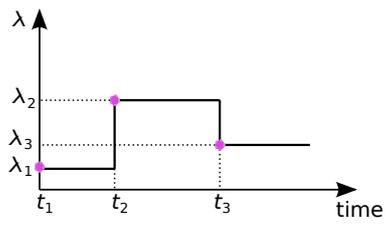


Figure 5: Example of a base profile with corner points for passenger arrival rates

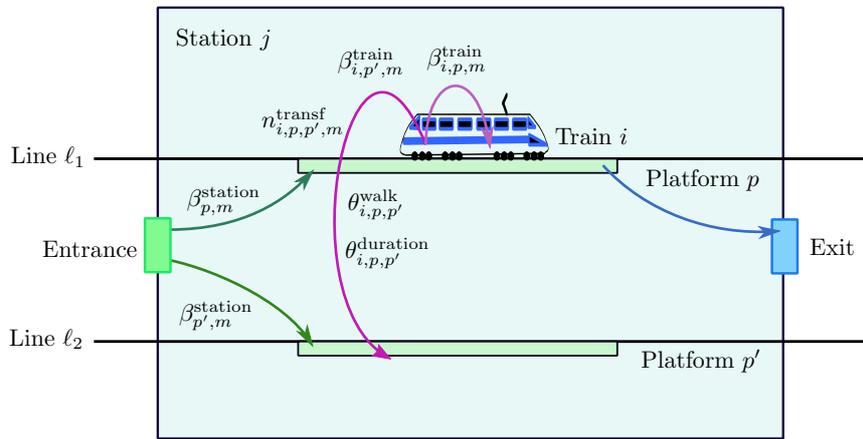


Figure 6: Variables for the splitting of passenger flows and passenger transfers

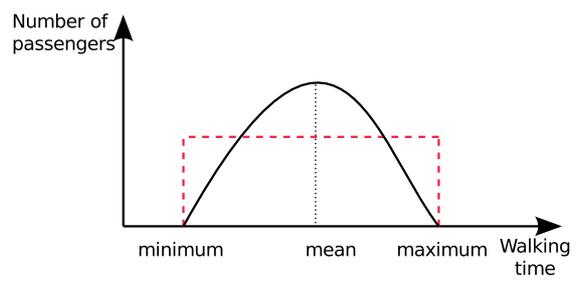
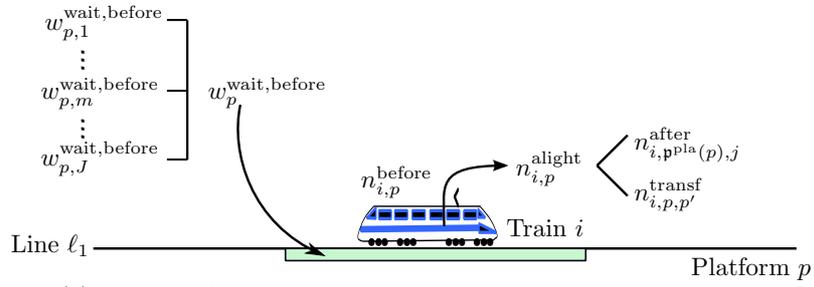
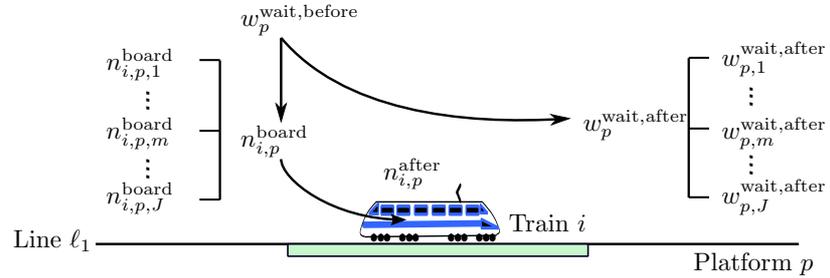


Figure 7: Typical walking time profile for the transfer passengers



(a) Variables immediately before the boarding process of passengers



(b) Variables immediately after the boarding process of passengers

Figure 8: Variables for the passenger characteristics before and after the boarding process

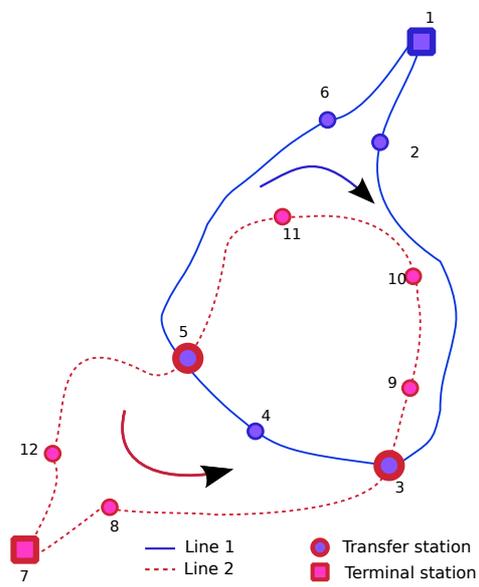
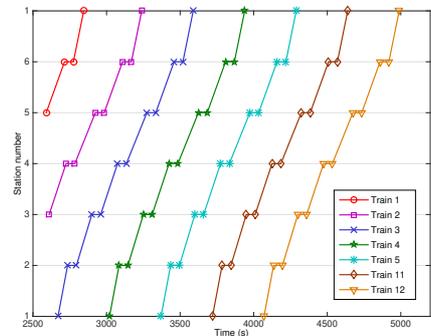
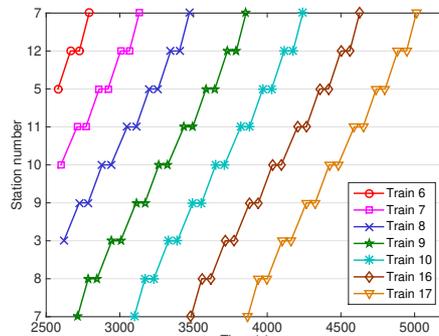


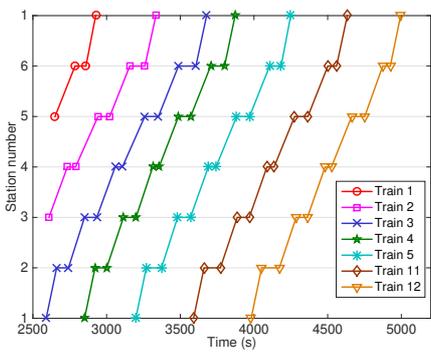
Figure 9: Layout of a small urban rail transit network



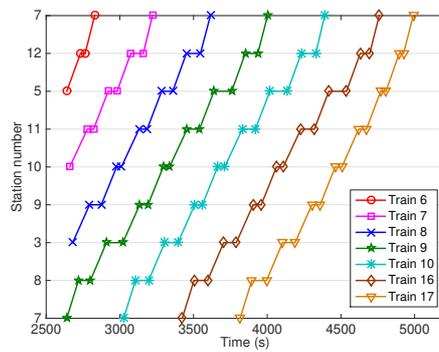
(a) Line 1, fixed headway



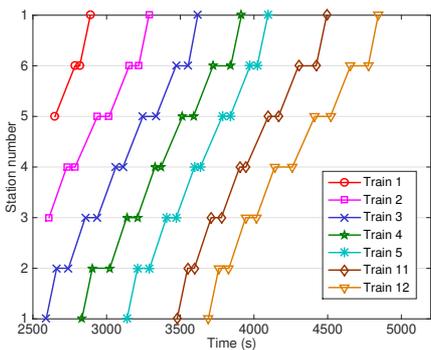
(b) Line 2, fixed headway



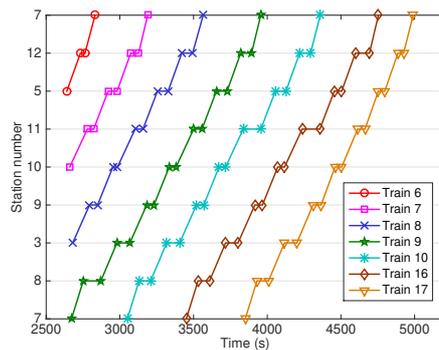
(c) Line 1, SQP



(d) Line 2, SQP



(e) Line 1, genetic algorithm



(f) Line 2, genetic algorithm

Figure 10: Train schedules for line 1 and line 2 obtained by the fixed-headway approach, the SQP method, and the genetic algorithm

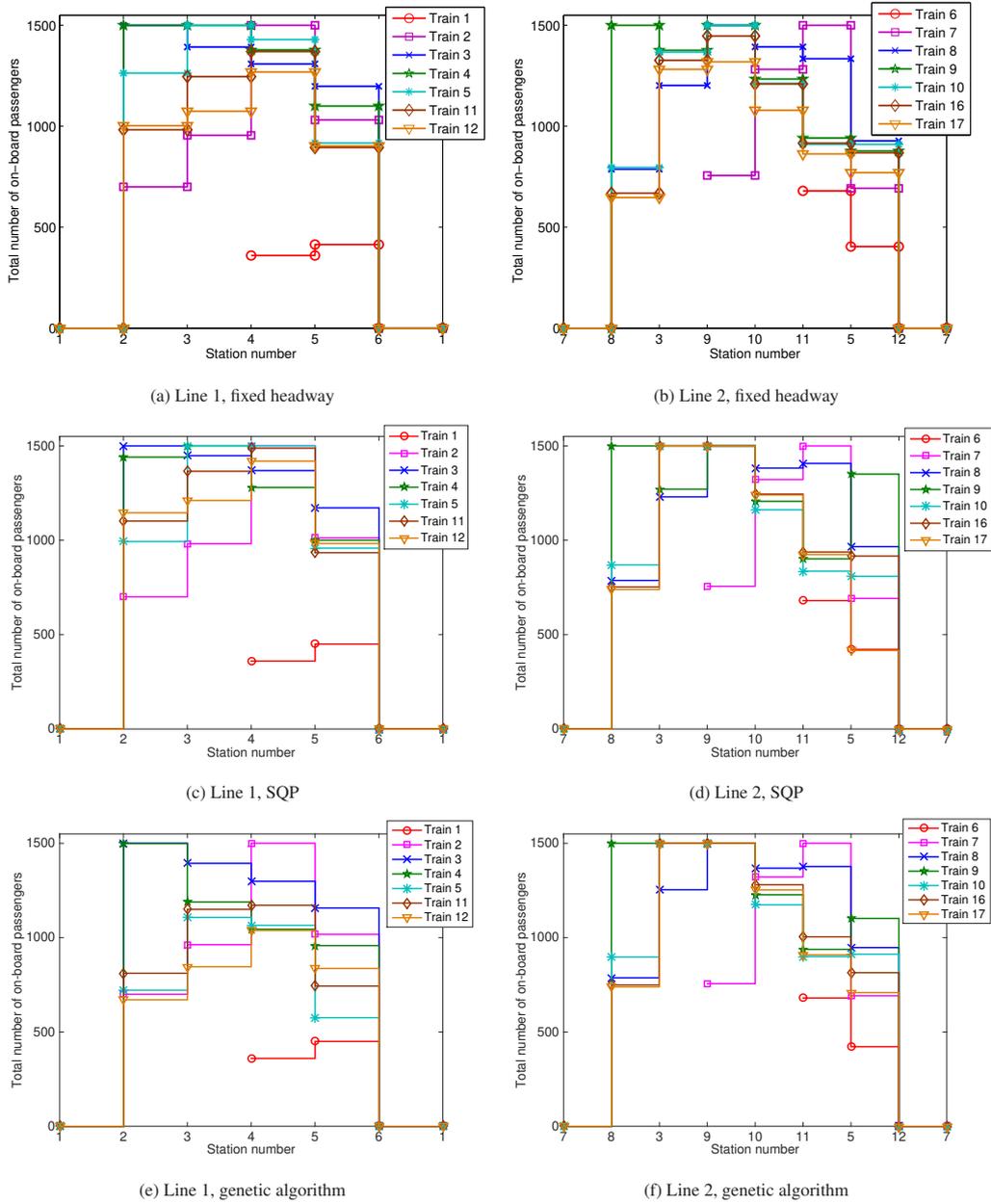
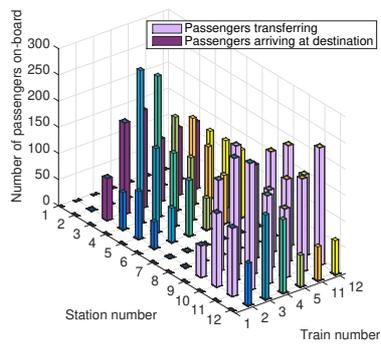
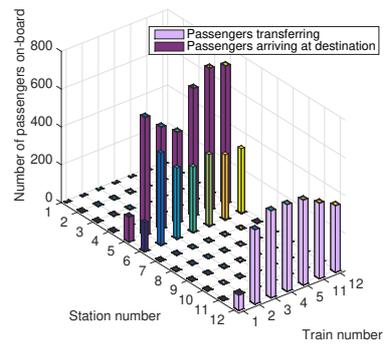


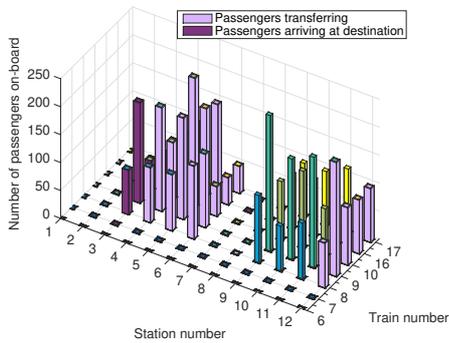
Figure 11: Number of onboard passengers for line 1 and line 2 obtained by the fixed-headway approach, the SQP method, and the genetic algorithm



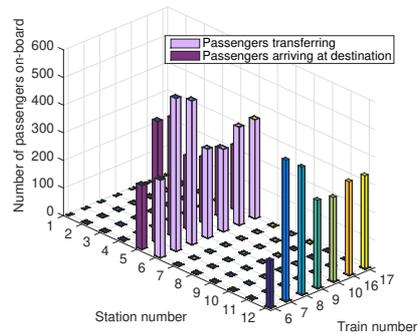
(a) Line 1, station 3



(b) Line 1, station 5

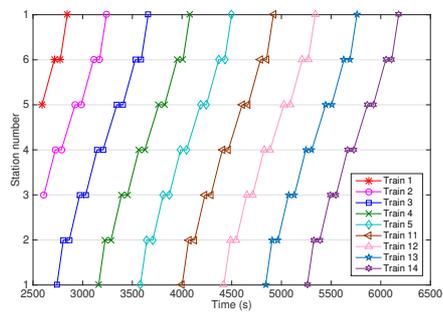


(c) Line 2, station 3

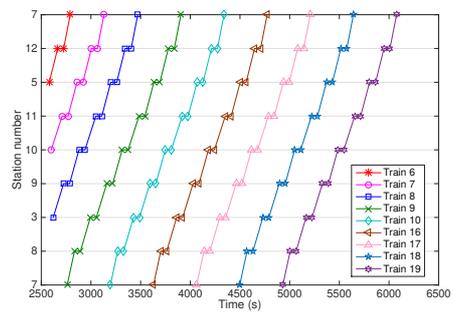


(d) Line 2, station 5

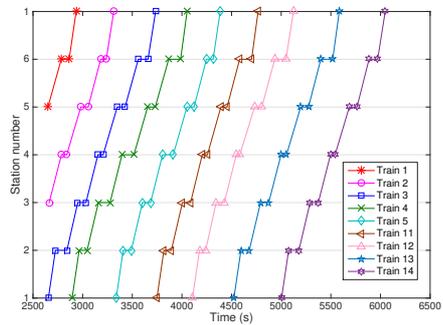
Figure 12: Number of onboard passengers with different destinations at transfer station 3 and 5 obtained by the SQP method for line 1 and line 2



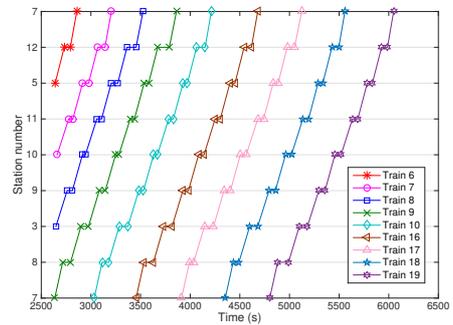
(a) Line 1, fixed headway



(b) Line 2, fixed headway



(c) Line 1, SQP



(d) Line 2, SQP

Figure 13: Train schedules for line 1 and line 2 obtained by the fixed-headway approach and the SQP method over a longer scheduling period (i.e., 3600 s)