

Technical report 15-023

# **Efficient real-time train scheduling for urban rail transit systems using iterative convex programming\***

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

Delft Center for Systems and Control  
Delft University of Technology  
Mekelweg 2, 2628 CD Delft  
The Netherlands  
phone: +31-15-278.51.19 (secretary)  
fax: +31-15-278.66.79  
URL: <http://www.dcsc.tudelft.nl>

---

\*This report can also be downloaded via [http://pub.deschutter.info/abs/15\\_023.html](http://pub.deschutter.info/abs/15_023.html)

# Efficient Real-Time Train Scheduling for Urban Rail Transit Systems Using Iterative Convex Programming

Yihui Wang, Bin Ning, Tao Tang, Ton J.J. van den Boom, and Bart De Schutter

**Abstract**—The real-time train scheduling problem for urban rail transit systems is considered with the aim of minimizing the total travel time of passengers and the energy consumption of the operation of trains. Based on the passenger demand in the urban rail transit system, the optimal departure times, running times, and dwell times are obtained by solving the scheduling problem. A new iterative convex programming (ICP) approach is proposed to solve the train scheduling problem. The performance of the ICP approach is compared with other alternative approaches, i.e., nonlinear programming approaches, a mixed integer nonlinear programming (MINLP) approach, and a mixed integer linear programming (MILP) approach. In addition, this paper formulates the real-time train scheduling problem with stop-skipping and shows how to solve it using an MINLP approach and an MILP approach. The ICP approach is shown, via a case study, to provide a better trade-off between performance and computational complexity for the real-time train scheduling problem. Furthermore, for the train scheduling problem with stop-skipping, the MINLP approach turns out to have a good trade-off between the control performance and the computational efficiency.

**Index Terms**—real-time train scheduling, urban rail transit, stop-skipping

## I. INTRODUCTION

WITH the increasing passenger demand for urban rail transit systems, such as subway systems, the frequency of train operations is becoming very high, especially in large cities like Beijing, Shanghai, Tokyo, New York, and Paris, where trains arrive at a station every 2 to 5 minutes. The planning process for the urban rail transit systems becomes more important for reducing the operation costs of railway operators and for guaranteeing passenger satisfaction, as characterized by waiting times, on-board times, and number of transfers. The planning process is traditionally a sequential process consisting of five phases [1]: demand analysis, line planning, train scheduling, rolling stock planning, and crew scheduling. This paper considers the train scheduling problem for urban rail transit systems.

In the urban rail transit systems considered in this paper, the lines are assumed to be separated from each other and each direction of the line has a separate rail track. Therefore,

Y. Wang, B. Ning, and T. Tang are with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Laboratory of Urban Mass Transit, Beijing Key Laboratory of Urban Mass Transit Automation and Control, Beijing Jiaotong University, Beijing, China, email: {yihui.wang, ttang, bning}@bjtu.edu.cn.

B. De Schutter and T.J.J. van den Boom are with the Delft Center for Systems and Control, Delft University of Technology, The Netherlands, email: {b.deschutter, a.j.j.vandenboom}@tudelft.nl.

trains do not overtake each other. In addition, for urban rail transit systems with high frequencies, it is not a major issue to the passengers anymore whether or not the train schedule is cyclic since new trains arrive at a station every 2 to 5 minutes. In practice, rail transport operators therefore do not announce the train schedule to passengers but only provide some information, such as that a train will arrive within 2 minutes. Hence, rail transport operators can schedule trains in real time based on the current situation, such as the number of waiting passengers at stations, the passenger arrival rates, and the number and position of running trains.

In the literature, there are several interpretations for real-time scheduling. For example, in [2]–[11], real-time scheduling is based on the existing timetable data and is used to handle route conflicts due to train delays or incidents. However, in [12]–[15], real-time scheduling is based on a constant headway between trains and it regulates the headways between trains through holding, deadheading, zone scheduling, short turning, and/or stop-skipping. Furthermore, automatic train regulation proposed in [16], [17] adjusts the running times and dwell times of trains to operate trains according to the train schedule and maintain the headway adherence. In the current paper, real-time scheduling means that there is no existing timetable or constant headways, but the schedule of trains is optimized in a rolling horizon way taking passenger demands and operation costs into consideration.

In [18], [19] the train schedule was obtained in an energy-efficient way but without considering the passenger demands. In that situation, the train schedule may be optimal from the point view of the rail operator, but it may not be optimal for the passengers. In [12], [20], [21] the passenger demand was taken into account in the train scheduling model. However, the capacity of trains is assumed to be unlimited, which is not the case in practice. In addition, only the passenger travel time was considered as objective function and the cost of the rail operator was not included in [12], [20], [21]. Niu and Zhou [22] proposed a binary integer programming model to optimize the urban rail timetable under time-dependent passenger demand, where the capacity of trains and the over-saturated stations are included in the model formulation. The uneven-headway timetable obtained in [22] was illustrated to reduce the average waiting time by almost 40% when compared with the regular timetable through a case study on Line 8 in Guangzhou, China. However, the energy consumption is not considered in [22]. A stochastic approximation approach is proposed to adjust the frequencies of different urban transit lines according to the

observed variable passenger demand [23]. However, the energy consumption of railway operation and dwell times at stations are not included in the model of [23]. In our scheduling model, the passenger characteristics are considered and the maximum capacity of trains is included. Furthermore, both the passenger travel time and the operation cost are included in the objective function.

We have proposed a real-time scheduling approach for trains based on the passenger demand in [24], where the capacity of the trains, the capacity of the stations, and the safety constraints caused by urban rail transit systems are included. The objective of the real-time scheduling problem is to minimize the total travel time of passengers. Furthermore, the problem is solved by the sequential quadratic programming (SQP) approach in [24]. However, the train scheduling problem is essentially a multi-objective optimization problem because it should consider both the benefits of the rail transport operators and the passengers [25], [26]. The rail transport operators prefer to minimize the operation cost (e.g., energy consumption). This conflicts with the benefit of the passengers (e.g., travel time) since a lower operation cost usually results in a longer travel time. In [27], we have solved the multi-objective train scheduling problem (i.e., minimizing the energy consumption and the passenger travel time) using the pattern search method, the mixed integer nonlinear programming (MINLP) approach, and the mixed integer linear programming (MILP) approach.

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

- A more realistic model for the operation of trains: Three operation phases for the operation of trains are considered, i.e., the acceleration phase, the speed holding phase, and the deceleration phase. As a result, the calculation of the energy consumption based on these three phases is more precise than that using the mean speed between stations as was done in [27].
- A new iterative convex programming (ICP) approach: We propose a new ICP approach to solve the real-time train scheduling problem more efficiently. The performance of the ICP approach is compared with the nonlinear programming approaches (i.e., pattern search method, SQP algorithm), the MINLP approach, and the MILP approach through a case study. Among these methods, the new ICP algorithm proposed in this paper obtains comparable results with respect to other approaches but with a lower computational complexity for the case study, which indicates that this approach can be applied for real-time use.
- A real-time train scheduling model with stop-skipping: This paper also considers stop-skipping, where trains may skip some small stations to reduce the passenger travel time and energy consumption (see Appendix C). Binary variables are introduced to indicate whether a train stops or skips a station. We show that the MINLP approach and the MILP approach can be directly applied to the problem with stop-skipping.

The rest of this paper is structured as follows. Section II formulates the dynamics of the operation of trains, the passenger

demand characteristics, and the passenger/vehicle interaction. Section III describes the multi-objective cost function and the constraints of the real-time train scheduling problem. Section IV proposes several solution approaches for the train scheduling problem, in particular, the new iterative convex programming approach. Section V compares the performance of the solution approaches in Section IV with a case study. Finally, Section VI concludes the paper with a short discussion of some topics for future work.

## II. MODEL FORMULATION

This paper considers one direction of an urban transit line consisting of  $J$  stations as shown in Figure 1. Station 1 is the origin station and station  $J$  is the final station of each trip. The track section between station  $j$  and station  $j+1$  is denoted as segment  $j$ .

We make the following assumptions when formulating the real-time scheduling model:

- A1. Station  $j$  for  $j \in \{2, 3, \dots, J-1\}$  can only accommodate one train at a time and no passing can occur at any point in the line.
- A2. Passengers arrive randomly at a constant rate  $\lambda_j$  at station  $j$ .
- A3. The number of passengers alighting from trains at station  $j$  for  $j \in \{1, 2, \dots, J\}$  is a fixed proportion  $\rho_j$  of its arrival load.
- A4. The number of passengers waiting at a station and the number of passengers on-board immediately after a train's departure are approximated by real numbers.

Assumption A1 generally holds for most urban transit systems, which are usually operated in first-in first-out order from station 1 to  $J$ . With Assumption A2, the passenger arrival rates are considered as constants for a scheduling period. This assumption is consistent with observed random passenger arrivals for short headway (less than 10 minutes) services [28]. The passenger arrival rates can be estimated based on a combination of the historical data, the data from the current automatic fare collection systems, and the data obtained by personal digital devices (PDAs) in future. Since the passenger demands of urban rail transit systems usually have certain patterns for different days, e.g., weekdays and weekend, in Assumption A3 we adopt the alighting proportions proposed in [29] to calculate the number of alighting passengers. The alighting proportions are usually assumed as constants for a scheduling period and could be estimated based on historical data and data collected in real-time from automatic fare collection systems and PDAs. The time-varying alighting proportions can be taken into account by the rolling horizon approach, where the alighting proportions can differ for each scheduling period. For Assumption A4, if the number of passengers is high, then the error made by this assumption is small. Furthermore, this assumption simplifies the optimization problem later on.

### A. The operation of a train

In the literature on train scheduling, the operation of trains is usually described by the departure times, arrival times,

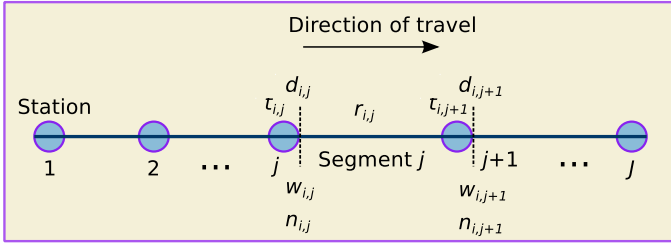


Fig. 1. Illustration of the subway line and variables

running times, and dwell times. In order to obtain a balanced trade-off between the accuracy and the computation speed, a macroscopic model is used. The detailed train dynamics, the position of block signals, the detection of trains, etc. can then be taken into account by the lower level control layer. The departure time  $d_{i,j}$  of train  $i$  at station  $j$  is

$$d_{i,j} = a_{i,j} + \tau_{i,j}, \quad (1)$$

where  $a_{i,j}$  and  $\tau_{i,j}$  are the arrival time and the dwell time of train  $i$  at station  $j$ . In the literature, the dwell time is usually considered as a constant. However, in practice, it is influenced by the number of passengers boarding and alighting from a train. Therefore, we consider a variable dwell time, as will be explained in Section II-C. The arrival time  $a_{i,j+1}$  of train  $i$  at station  $j+1$  equals the sum of the departure time  $d_{i,j}$  at station  $j$  and the running time  $r_{i,j}$  on segment  $j$  (i.e., the track between station  $j$  and station  $j+1$  as shown in Figure 1) for train  $i$ :

$$a_{i,j+1} = d_{i,j} + r_{i,j}. \quad (2)$$

In this paper, we assume that the operation of trains only consists of three phases: the acceleration phase, the speed holding phase, and the deceleration phase. In addition, we assume the acceleration  $a_{i,j}^{\text{acc}}$  and the deceleration  $a_{i,j}^{\text{dec}}$  are known constants. Define the speed in the speed holding phase as  $v_{i,j}$ , the running distance of these three phases can be calculated as

$$s_{i,j}^{\text{acc}} = \frac{v_{i,j}^2}{2a_{i,j}^{\text{acc}}}, \quad s_{i,j}^{\text{dec}} = -\frac{v_{i,j}^2}{2a_{i,j}^{\text{dec}}}, \quad s_{i,j}^{\text{hold}} = s_j - s_{i,j}^{\text{acc}} - s_{i,j}^{\text{dec}}.$$

where  $s_j$  is the length of segment  $j$ . The running time  $r_{i,j}$  of train  $i$  on segment  $j$  is equal to the sum of the acceleration time, the holding time, and the deceleration time, i.e.,  $r_{i,j} = t_{i,j}^{\text{acc}} + t_{i,j}^{\text{hold}} + t_{i,j}^{\text{dec}}$ , where  $t_{i,j}^{\text{acc}} = v_{i,j}/a_{i,j}^{\text{acc}}$ ,  $t_{i,j}^{\text{hold}} = s_{i,j}^{\text{hold}}/v_{i,j}$ , and  $t_{i,j}^{\text{dec}} = -v_{i,j}/a_{i,j}^{\text{dec}}$ . So the running time can be recast as

$$\begin{aligned} r_{i,j} &= t_{i,j}^{\text{acc}} + t_{i,j}^{\text{hold}} + t_{i,j}^{\text{dec}} \\ &= \frac{v_{i,j}}{a_{i,j}^{\text{acc}}} + \frac{s_j - s_{i,j}^{\text{acc}} - s_{i,j}^{\text{dec}}}{v_{i,j}} + \frac{-v_{i,j}}{a_{i,j}^{\text{dec}}} \\ &= \frac{v_{i,j}}{a_{i,j}^{\text{acc}}} + \frac{s_j}{v_{i,j}} - \frac{v_{i,j}}{2a_{i,j}^{\text{acc}}} + \frac{v_{i,j}}{2a_{i,j}^{\text{dec}}} - \frac{v_{i,j}}{a_{i,j}^{\text{dec}}} \\ &= \frac{s_j}{v_{i,j}} + \frac{v_{i,j}}{2a_{i,j}^{\text{acc}}} - \frac{v_{i,j}}{2a_{i,j}^{\text{dec}}}. \end{aligned} \quad (3)$$

**Remark.** The coasting phase of the operation of trains can be included as follows (at the cost of an increased number of variables and an increased computational complexity). Both

the traction force and braking force are equal to zero in the coasting phase, where the train speed slows down because of the resistance. With the model given above, the entering speed (i.e., the holding speed) of the coasting phase is  $v_{i,j}$ . We denote the speed at the end of the coasting phase as  $v'_{i,j}$ . The resistance varies with the train speed. Here, we approximate the resistance using the mean speed  $v_{m,i,j} = (v_{i,j} + v'_{i,j})/2$  of the coasting phase:

$$R_{i,j} = m(k_{1i} + k_{2i}v_{m,i,j}) + k_{3i}v_{m,i,j}^2,$$

where  $m$  is the mass of the train itself and of the passengers on board of the train and  $k_{1i}$ ,  $k_{2i}$ , and  $k_{3i}$  are the resistance coefficients of train  $i$ . Hence, the running distance and running time of the coasting phase can be calculated. The coasting phase can then be included in the model formulation for the optimization of train schedule.

Note that the speed  $v_{i,j}$  of the holding phase should satisfy

$$v_{i,j} \in [v_{i,j,\min}, v_{i,j,\max}], \quad (4)$$

where  $v_{i,j,\min}$  and  $v_{i,j,\max}$  are the minimal and maximal running speed for the speed holding phase of train  $i$  at segment  $j$ , respectively. The maximum running speed is limited by the train characteristics and the condition of the line. The minimum running speed is introduced to ensure passenger satisfaction since if trains run too slow, the passengers may complain.

The minimum headway is the minimum time interval between two successive trains so that they can enter and depart from a station safely [30]. Followed from assumption A1, a train cannot enter a station until a minimum headway  $h_0$  after the preceding train's departure, which can be formulated as

$$a_{i,j} - d_{i-1,j} \geq h_0. \quad (5)$$

In addition, we select the order in which the trains run such that vehicle  $i-1$  always precedes train  $i$  for  $i \in \{1, 2, 3, \dots, I\}$  with  $I$  the total number of trains.

**Remark.** After the schedule of trains has been obtained, a more accurate speed profile can be calculated in a lower control layer based on the detailed dynamics of the operation trains and line segments between station. For more information about the speed profile optimization see [31]–[34].

## B. Passenger demand characteristics

The number of passengers still remaining at the station after the departure of train  $i-1$  immediately after its departure at station  $j$  is defined as  $w_{i-1,j}$ . The number of passengers who want to get on train  $i$  at station  $j$  can then be formulated as

$$w_{i,j}^{\text{waiting}} = w_{i-1,j} + \lambda_j(d_{i,j} - d_{i-1,j}), \quad (6)$$

where  $\lambda_j(d_{i,j} - d_{i-1,j})$  is the number of the passengers that arrived during the departure of train  $i-1$  and the departure of train  $i$ .

By defining the number of passengers on train  $i$  immediately after its departure at station  $j-1$  as  $n_{i,j-1}$ , the remaining

capacity of train  $i$  at station  $j$  immediately after the alighting process of the passengers is

$$n_{i,j}^{\text{remaining}} = C_{i,\text{max}} - n_{i,j-1}(1 - \rho_j), \quad (7)$$

where  $C_{i,\text{max}}$  is the effective maximal capacity of train  $i$ , and  $n_{i,j-1}(1 - \rho_j)$  is the number of passengers remaining on train  $i$  immediately after all the passengers that wanted to leave the train have gotten off. Note that the effective maximal capacity can be estimated based on the data from the daily operations, where the distribution of on-board passengers and the effect of the distribution of waiting passengers on the platforms, etc. can be taken into account.

The number of passengers boarding train  $i$  at station  $j$  is equal to the minimum of the remaining capacity and the number of waiting passengers, i.e.

$$n_{i,j}^{\text{boarding}} = \min(n_{i,j}^{\text{remaining}}, w_{i,j}^{\text{waiting}}). \quad (8)$$

The number of passengers at station  $j$  immediately after the departure of train  $i$ , i.e. the passengers who cannot get on train  $i$ , is then given by

$$w_{i,j} = w_{i,j}^{\text{waiting}} - n_{i,j}^{\text{boarding}}. \quad (9)$$

The number of passengers on train  $i$  when it departs from station  $j$  is equal to the sum of the passengers arriving but not getting off at station  $j$  and the passengers boarding on train  $i$  at station  $j$ , which can be formulated as

$$n_{i,j} = n_{i,j-1}(1 - \rho_j) + n_{i,j}^{\text{boarding}}. \quad (10)$$

### C. Passenger/vehicle interaction

As mentioned before, the dwell time is influenced by the number of alighting and boarding passengers. According to the research for Beijing subway stations in [35], the goodness of fit of the linear and nonlinear model for the dwell time are 86% and 94%, respectively. Since the nonlinear model needs detailed information of passengers, such as the passenger distributions, here we adopt the linear model, as it has an acceptable performance and as it will simplify the model for passenger characteristics and the optimization problem later on. The minimum dwell time can be described as [35]:

$$\tau_{i,j,\text{min}} = \alpha_{1,d} + \alpha_{2,d}n_{i,j-1}\rho_j + \alpha_{3,d}n_{i,j}^{\text{boarding}}, \quad (11)$$

where  $\alpha_{1,d}$ ,  $\alpha_{2,d}$ , and  $\alpha_{3,d}$  are coefficients that can be estimated based on historical data. The dwell time  $\tau_{i,j}$  should be larger than the minimal dwell time  $\tau_{i,j,\text{min}}$  such that the passengers can get on and get off the train. However, it should be less than a maximum dwell time  $\tau_{i,j,\text{max}}$  to ensure the passenger satisfaction.

**Remark.** In this paper, the time for opening and closing of doors are not included in the calculation of the minimum dwell time. In addition, a minimum operational dwell time, e.g., 30 s, is not considered here. When calculating the train schedules in practice, both the minimum operational dwell time and the door opening and closing time should be taken into account.

## III. THE REAL-TIME TRAIN SCHEDULING PROBLEM

Based on the model formulation in Section II, we now formulate the real-time train scheduling problem. The energy consumption caused by the operation of trains and the total travel time of all passengers are minimized using the weighted sum strategy for the real-time train scheduling problem. In [27], [36], the energy consumption is proportional to the product of the resistance and the mean speed, where the resistance includes the rolling resistance, air resistance, and grade resistance. However, in the iteration of the operation of trains, the computation of the energy consumption is more precise [18], [31]–[34], [37]. Hence, we consider an operation model of a train with the acceleration phase, the speed holding phase, and the deceleration phase in this paper.

The energy consumption for train  $i$  on segment  $j$  is equal to the sum of the consumption of the acceleration phase, the speed holding phase, and the deceleration phase. The energy consumption for the acceleration phase of train  $i$  on segment  $j$  is

$$E_{i,j}^{\text{acc}} = \int_0^{t_{i,j}^{\text{acc}}} \left( (m_{e,i} + n_{i,j}m_p) \left( a_{i,j}^{\text{acc}} + k_{1i} + k_{2i}v(t) + g \sin(\theta_j) \right) + k_{3i}v^2(t) \right) v(t) dt, \quad (12)$$

where  $m_{e,i}$  is the mass of train  $i$  itself,  $m_p$  is the mass of one passenger,  $(m_{e,i} + n_{i,j}m_p)$  is the mass of train  $i$  and the passengers on-board of train  $i$  at station  $j$ ,  $k_{1i}$ ,  $k_{2i}$ , and  $k_{3i}$  are the resistance coefficients of train  $i$ ,  $\theta_j$  is the grade profile of segment  $j$ , and the speed in the acceleration phase is changing with time  $t$  and can be calculated by  $v(t) = a_{i,j}^{\text{acc}}t$ . In addition, the speed at the end of the acceleration phase is equal to the speed of the holding phase, i.e.,  $v_{i,j} = a_{i,j}^{\text{acc}}t_{i,j}^{\text{acc}}$ . The energy consumption for the speed holding phase of train  $i$  on segment  $j$  is

$$E_{i,j}^{\text{hold}} = \int_{t_{i,j}^{\text{acc}}}^{t_{i,j}^{\text{acc}} + t_{i,j}^{\text{hold}}} \left( (m_{e,i} + n_{i,j}m_p) (k_{1i} + k_{2i}v_{i,j} + g \sin(\theta_j)) + k_{3i}v_{i,j}^2 \right) v_{i,j} dt. \quad (13)$$

In the deceleration phase, the energy consumption for the air braking system is small compared with the traction energy and is usually ignored in the literature of the operation of trains [18], [31]–[34], [36], [37]. For the sake of completeness, we include it in the energy consumption for deceleration phase. In addition, electric motors work as electric generators to generate energy for the regenerative braking system. Hence, the energy consumption may become negative in the braking process of trains. The energy consumption of the deceleration phase of train  $i$  on segment  $j$  is calculated by

$$E_{i,j}^{\text{dec}} = \varepsilon_{i,j} + \eta_{i,j} \int_{t_{i,j}^{\text{acc}} + t_{i,j}^{\text{hold}}}^{t_{i,j}^{\text{dec}}} \left( (m_{e,i} + n_{i,j}m_p) \left( a_{i,j}^{\text{dec}} + k_{1i} + k_{2i}v(t) + g \sin(\theta_j) \right) + k_{3i}v_{i,j}^2 \right) v_{i,j} dt, \quad (14)$$

where  $\varepsilon_{i,j}$  is the energy consumption of the air braking system for train  $i$  on segment  $j$  and  $\eta_{i,j}$  is the energy recovery rate



in the deceleration phase of train  $i$  on segment  $j$ . The total energy consumption for all  $I$  trains running with  $J$  stations can then be formulated as

$$E_{\text{total}} = \sum_{i=1}^I \sum_{j=1}^J (E_{i,j}^{\text{acc}} + E_{i,j}^{\text{hold}} + E_{i,j}^{\text{dec}}). \quad (15)$$

The total travel time is the sum of the passenger waiting time and the passenger in-vehicle time. The passenger waiting time  $t_{\text{wait},i,j}$  at station  $j$  for train  $i$  includes the waiting time of both passengers left by the previous train  $i-1$  and the newly arrived passengers, and it can be calculated by

$$t_{\text{wait},i,j} = w_{i-1,j}(d_{i,j} - d_{i-1,j}) + \frac{1}{2}\lambda_j(d_{i,j} - d_{i-1,j})^2, \quad (16)$$

where the first term represents the waiting time of the passengers left by train  $i-1$  at station  $j$ , and the second term represents the waiting time of randomly arriving passengers between the departures of train  $i-1$  and train  $i$ . The passenger in-vehicle time for train  $i$  running from station  $j$  to  $j+1$  includes the running time for all passengers on train  $i$  after its departure from station  $j$  and the waiting time of the passengers who do not get off the train at station  $j+1$ , which can be formulated as

$$t_{\text{in-vehicle},i,j} = n_{i,j}r_{i,j} + n_{i,j}(1 - \rho_{j+1})\tau_{i,j+1}. \quad (17)$$

The total passenger travel time for all  $I$  trains can then be formulated as

$$t_{\text{total}} = \sum_{i=1}^I \sum_{j=1}^{J-1} (t_{\text{wait},i,j} + t_{\text{in-vehicle},i,j}). \quad (18)$$

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

$$f_{\text{opt}} = \frac{E_{\text{total}}}{E_{\text{total,nom}}} + \lambda \frac{t_{\text{total}}}{t_{\text{total,nom}}}, \quad (19)$$

where  $\lambda$  is a non-negative weight, and the normalization factors  $E_{\text{total,nom}}$  and  $t_{\text{total,nom}}$  are the nominal values of the total energy consumption and the total travel time of passengers, respectively. These nominal values can e.g. be determined by running trains using a feasible initial schedule.

The constraints of the real-time scheduling problem consist of the running time constraints, dwell time constraints, headway constraints, and capacity of trains, shown as (1)-(5), (9)-(11) in Section II.

Since passenger demands vary with the 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 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 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  $t_0 + t_p$ . Once the new optimal schedule is computed, it is executed for  $t_p$  time units, and next 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.

## IV. SOLUTION APPROACHES

The resulting real-time train scheduling problem with objective function (19) and constraints (1)-(5) and (9)-(11) is a non-smooth non-convex problem, where the non-smoothness is caused by the min function in (8), and the non-convexity is due to the nonlinear non-convex objective function and the non-convex set defined by constraints. We propose a new iterative convex programming (ICP) approach to solve the real-time train scheduling problem in Section IV-A. In addition, we solve the train scheduling problem using several alternative approaches in Section IV-B. One is a gradient-free non-smooth programming approach, e.g., pattern search. Another one is a gradient-based nonlinear programming, e.g., sequential quadratic programming. Furthermore, a general purpose nonlinear integer programming approach, e.g., the branch-and-bound algorithm, is also used. By approximating the nonlinear objective function using piecewise affine functions, the train scheduling problem can be recast into an MILP problem, which can be solved efficiently by existing solvers, e.g., CPLEX.

### A. A new approach: ICP

The non-differentiability of the train scheduling problem is introduced by the min function, which is used to describe the number of waiting passengers  $w_{i,j}$  and the passengers onboard  $n_{i,j}$  immediately after train's departure. Since the non-differentiability and non-convexity of the train scheduling problem are introduced by the calculation of the number of waiting passengers and onboard passengers, we propose an iterative convex programming (ICP) approach inspired by the iterative (alternating) approaches in [38]–[41]. The ICP approach is based on a reduced-complexity model, where the number of waiting passengers and onboard passengers are not variables anymore but estimated values. The estimated values of  $\hat{w}_{i,j}(p)$  and  $\hat{n}_{i,j}(p)$  are calculated via simulation using the model proposed in Section II with fixed departure times  $d_{i,j}(p)$ , fixed running times  $r_{i,j}(p)$ , and fixed dwell times  $\tau_{i,j}(p)$ . This eliminates the min function and the nonlinear terms  $w_{i,j}d_{i,j}$ ,  $w_{i,j}d_{i-1,j}$ ,  $n_{i,j}r_{i,j}$  and  $n_{i,j}\tau_{i,j+1}$  in the objective function. Hence, the resulting optimization problem is a smooth and convex problem, which can often be solved efficiently for the global optimum using interior point algorithms [42]. By solving the convex problems iteratively, an approximation of the global optimum of the original non-smooth non-convex problem can be obtained. The procedure of the ICP method is given in Algorithm 1.

For the ICP approach, the variables of the real-time scheduling problem are the departure times  $d_{i,j}$ , the running times  $r_{i,j}$ , and the dwell times  $\tau_{i,j}$ . The number of passengers  $w_{i,j}$  waiting at stations and the number of passengers  $n_{i,j}$  on-board the trains are estimated by  $\hat{w}_{i,j}$  and  $\hat{n}_{i,j}$ , respectively. The other variables, such as the passenger waiting time  $t_{\text{wait},i,j}$  and the passenger on-board times  $t_{\text{in-vehicle},i,j}$ , are functions of the decision variables and hence are not explicitly represented in the solution process. The solution obtained by the ICP approach is not necessarily the global minimum of the formulated scheduling problem since the ICP approach

solves a sequence of convex approximations of the formulated nonlinear non-convex problem. For the ICP approach, we should in general use multiple starting points. However, for this case study, we found that one random feasible initial point yields comparable results with respect to other alternative approaches. The resulting convex problem in the ICP approach can be solved using the ellipsoid algorithm and interior point algorithm, which are implemented in the Matlab software CVX for disciplined convex programming<sup>1</sup>.

---

**Algorithm 1** The procedure of the ICP method

---

- 1: **Input**: a feasible initial solution of departure times, running times, and dwell times, i.e.,  $d_{i,j}(0)$ ,  $r_{i,j}(0)$ , and  $\tau_{i,j}(0)$  for  $i = 1, \dots, I$  and  $j = 1, \dots, J$ ,  $p_{\max}$ , convergence tolerance  $\zeta$ , maximum number of iterations  $p_{\max}$ ;
  - 2: iteration index  $p \leftarrow 0$ ;
  - 3: calculate initial estimates  $\hat{w}_{i,j}(p)$  and  $\hat{n}_{i,j}(p)$  using (9) and (10) based on  $d_{i,j}(p)$ ,  $r_{i,j}(p)$ , and  $\tau_{i,j}(p)$ ;
  - 4: calculate the initial objective  $f_{\text{opt}}(p)$  using (19) ;
  - 5: **Repeat**
  - 6:      $p = p + 1$ ;
  - 7:     substitute the estimates  $\hat{w}_{i,j}(p-1)$  and  $\hat{n}_{i,j}(p-1)$  into the original problem to get a convex problem;
  - 8:     obtain optimal departure time  $d_{i,j}^*(p)$ , running time  $r_{i,j}^*(p)$ , and dwell time  $\tau_{i,j}^*(p)$  by solving the convex problem;
  - 9:     compute  $\hat{w}_{i,j}(p)$  and  $\hat{n}_{i,j}(p)$  using (9) and (10) based on  $d_{i,j}^*(p)$ ,  $r_{i,j}^*(p)$ , and  $\tau_{i,j}^*(p)$ ;
  - 10:     calculate the objective  $f_{\text{opt}}(p)$  using (19);
  - 11: **Until**  $p = p_{\max}$  or  $|f_{\text{opt}}(p) - f_{\text{opt}}(p-1)| \leq \zeta$
  - 12: **Return**  $d_{i,j}^*(p)$ ,  $r_{i,j}^*(p)$ ,  $\tau_{i,j}^*(p)$ ,  $f_{\text{opt}}(p)$
- 

### B. Alternative approaches

Nonlinear programming approaches, an MINLP approach, and an MILP approach are adopted to solve the real-time train scheduling problem.

1) *Gradient-free nonlinear programming*: Nonlinear programming approaches can be grouped in gradient-free approaches and gradient-based approaches. The gradient-free approaches do not explicitly require gradient and Hessian information but only require that the values of the objective function can be ranked. Moreover, gradient-free methods are suitable for non-smooth problems. Since the real-time train scheduling problem is non-smooth due to the min function, the first choice is to use the gradient-free method. Here, in particular we propose to use the pattern search method.

The pattern search method handles optimization problems with nonlinear, linear, and bound constraints, and does not require objective functions to be differentiable or continuous. The pattern search method was first proposed by Hooke and Jeeves [43], and it has been proved successful in practice even for objective functions with many local minima, in combination with a multi-start method, to improve the probability of obtaining a solution close to a globally optimal solution.

When solving the real-time scheduling problem using the pattern search method, the variables are the departure times  $d_{i,j}$ , the running times  $r_{i,j}$ , and the dwell times  $\tau_{i,j}$ . The other variables, such as the number of passengers  $w_{i,j}$  waiting at stations, the number of passengers  $n_{i,j}$  on-board the trains, the passenger waiting times  $t_{\text{wait},i,j}$ , and the passenger on-board times  $t_{\text{in-vehicle},i,j}$ , can be eliminated. The pattern search method has been implemented in the global optimization toolbox of Matlab.

2) *Gradient-based nonlinear programming*: The gradient-based nonlinear programming methods rely on gradient and Hessian information. If this information is not available, it can be approximated numerically. We consider the gradient-based sequential quadratic programming (SQP) algorithm here since it is widely used to solve nonlinear programming problems. A requirement for the SQP algorithm is that the objective function and the constraints of the nonlinear programming problem should be continuously differentiable [44]. In the SQP method, the nonlinear programming problem is recast as a sequence of quadratic programming problems, which can be solved easily and efficiently. The train scheduling problem in Section III is non-differentiable because of the min function. Even though the SQP<sup>2</sup> algorithm is a gradient-based method, we also apply it to our problem setting since it yields good solutions to most of the nonlinear programming problems.

When solving the real-time scheduling problem using the SQP algorithm, the variables are the same as those in the pattern search method. The SQP algorithm has been implemented in many packages, such as SNOPT and the optimization toolbox of MATLAB.

3) *The MINLP approach*: In the MINLP approach, we introduce auxiliary binary variables and auxiliary real variables to deal with the non-smooth min function in (8). By introducing a binary variable  $\delta_{i,j} \in \{0, 1\}$  and defining

$$\tilde{f}_{i,j} = w_{i-1,j} + \lambda_j(d_{i,j} - d_{i-1,j}) - [C_{i,\max} - n_{i,j-1}(1 - \rho_j)], \quad (20)$$

the following equivalence holds [45]:

$$[\tilde{f}_{i,j} \leq 0] \Leftrightarrow [\delta = 1] \quad (21)$$

is true iff

$$\begin{cases} \tilde{f}_{i,j} \leq \tilde{M}_{i,j}(1 - \delta_{i,j}) \\ \tilde{f}_{i,j} \geq \varepsilon + (\tilde{m} - \varepsilon)\delta_{i,j} \end{cases}, \quad (22)$$

where  $\varepsilon$  is a small positive number that is introduced to transform a strict equality into a non-strict inequality, and  $\tilde{M}_{i,j}$  and  $\tilde{m}_{i,j}$  are the maximum value and the minimum value of  $\tilde{f}_{i,j}$ , respectively. Equation (8) can now be rewritten as

$$\begin{aligned} n_{i,j}^{\text{boarding}} = & \delta_{i,j}[w_{i-1,j} + \lambda_j(d_{i,j} - d_{i-1,j})] + (1 - \delta_{i,j})[C_{i,\max} \\ & - n_{i,j-1}(1 - \rho_j)]. \end{aligned} \quad (23)$$

Note that in equation (23) there are four nonlinear terms (i.e.,  $\delta_{i,j}w_{i-1,j}$ ,  $\delta_{i,j}d_{i,j}$ ,  $\delta_{i,j}d_{i-1,j}$ ,  $\delta_{i,j}n_{i,j-1}$ ), which are the products

<sup>2</sup>When the SQP algorithm is applied to the non-differentiable problem, it might get stuck in a local solution. In the non-differentiable problem proposed in this paper, the minimum value of the objective function is usually not obtained at the non-differentiable points, so the SQP algorithm jumps over these points successfully. Therefore, the SQP approach with multiple initial points works well in this case.

<sup>1</sup>More information about Matlab software CVX, see <http://cvxr.com/cvx>.

of the binary variable  $\delta_{i,j}$  and real variables. Four auxiliary real variables can then be introduced to transform these four nonlinear terms into linear terms with linear constraints. The detailed information about this transformation is given in Appendix A.

The variables of the resulting MINLP problem include the variables occurring in the pattern search method and the binary variables  $\delta_{i,j}$ . The MINLP problem can be solved using a branch-and-bound algorithm, such as the MINLP BB solver<sup>3</sup> and SCIP [46]. In addition, we now propose a bi-level optimization method to solve the MINLP problem. This method consists of two levels of optimization. The high-level optimization optimizes the binary variables, where a brute force approach can be used to find all the combinations for the binary variables when the size of the problem is small. Alternatively, integer programming approaches, such as genetic algorithms implemented in the global optimization toolbox of Matlab [47], can be applied in the high-level optimization. For each combination of binary variables, the nonlinear optimization problem in the lower level is now a smooth problem, which can be solved using gradient-based approaches, such as the interior point algorithm implemented in the optimization toolbox of MATLAB [48].

4) *The MILP approach:* In our earlier work [33], we have shown that the mixed integer linear programming (MILP) approach can be very efficient for train trajectory planning problems. Therefore, we also apply the MILP approach to the real-time train scheduling problem. In this approach, we approximate the nonlinear terms in the objective function by piecewise affine approximations and then transform the non-smooth non-convex problem into an MILP problem. The MILP approach deals with the min function of (8) in the same way as the MINLP approach. In the MINLP problem in Section IV-B3, the constraints are linear, but the objective function is nonlinear and non-convex. Therefore, in order to solve the real-time rescheduling problem as an MILP problem, we need to approximate the nonlinear terms as piecewise affine functions, such as  $w_{i-1,j}d_{i,j}$ ,  $n_{i,j}r_{i,j}$ , and  $n_{i,j}\tau_{i,j}$ . For more information about the piecewise affine approximation, see Appendix B.

The variables of the resulting MILP problem include the variables in the MINLP problem and the binary variables and the auxiliary variables introduced by the approximations of nonlinear terms in the objective function. The MILP problem can be solved by the branch-and-bound algorithm implemented in several existing commercial and free solvers, such as CPLEX, Xpress-MP, and GLPK [49], [50].

## V. CASE STUDY

### A. Set-up

In order to demonstrate and compare the performance of the approaches proposed in Section IV for the real-time train scheduling problem, the train characteristics and the line data of the Yizhuang subway line in Beijing are used as a test case study. Note that this case study that considers one subway line is compact and easily interpreted. The Yizhuang line has

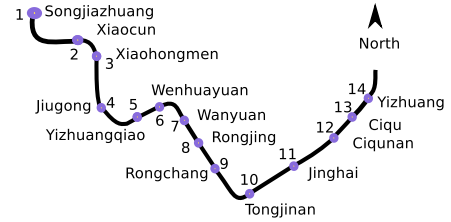


Fig. 2. The layout of the Yizhuang subway line

TABLE I  
INFORMATION OF THE YIZHUANG SUBWAY LINE

Station number	Distance to next station [m]	Passenger arrival rate [passenger/s]	Passenger alighting proportion	Minimum running time [s]
1	1332	3	0	87.721
2	1286	0.5	0.05	85.651
3	2086	3	0.3	121.654
4	2265	4	0.38	129.710
5	2331	0.4	0.04	132.680
6	1354	4	0.32	88.711
7	1280	4	0.38	85.380
8	1544	3	0.7	97.260
9	992	3	0.6	72.420
10	1975	3	0.7	116.659
11	2369	3	0.7	134.391
12	1349	2	0.5	88.486
13	2610	2	0.5	145.237
14	-	0	1	-

14 stations, and the speed limit for the line is 80 km/h (i.e., 22.2 m/s). The detailed information about these 14 stations is listed in Table I. The minimum running time in Table I is calculated by taking a fixed acceleration of 0.8 m/s<sup>2</sup> and a fixed deceleration of -0.8 m/s<sup>2</sup>. Furthermore, the speed  $v_{i,j}$  of the holding phase in (3) is equal to the maximum speed 22.2 m/s for the minimum running time. We assume the maximum running time is  $r_{i,j,\max} = \zeta r_{i,j,\min}$ , where  $\zeta$  is larger than 1. We have chosen  $\zeta$  as 1.2 in order to ensure that the passengers do not complain that the train is too slow. Based on the maximum running time, the minimal holding speed can be calculated.

The other parameters of trains and passengers are shown in Table II. The mass of the train itself and the standard mass of one passenger is shown in Table II. The values of the dwell time coefficients are chosen as shown in Table II according to [51]. The minimum dwell time can then be calculated by (11). The maximal dwell time is chosen as 150 s. The effective capacity of each train is 1468 passengers. The communication-based train control system (i.e., a moving block signaling scheme) is implemented in the Beijing Yizhuang subway line and the minimum headway  $h_0$  between two successive trains is 90 s. Moreover, in practice the regenerated energy of the Beijing Yizhuang line is hardly fed back to the power supply system due to some technical problems, e.g., over voltages. Therefore, the regenerative energy is not taken into account in this case study.

In order to illustrate the performance of the solution approaches proposed for different-sized problems, we considered 9 scenarios with different problem sizes as shown in Table III, where the values of  $I$  and  $J$  are the number of trains and stations involved. For the cases with  $J$  less than 14,

<sup>3</sup>For more details about MINLP BB solver, see <http://tomopt.com>.



TABLE II  
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
Maximum dwell time [s]	$\tau_{\max}$	150
Coefficients of the minimal dwell time	$\alpha_{1,d}$	4.002
	$\alpha_{2,d}$	0.047
	$\alpha_{3,d}$	0.051
Coefficients of resistance	$k_{1i}$	0.012
	$k_{2i}$	$5.049 \cdot 10^{-4}$
	$k_{3i}$	8.521

TABLE III  
THE NOMINAL VALUES OF THE ENERGY CONSUMPTION AND THE PASSENGER TRAVEL TIME

Scenario index	$I \& J$	Nominal passenger travel time [s]	Nominal energy consumption [J]
1	$I=2, J=3$	$6.402 \cdot 10^5$	$1.216 \cdot 10^8$
2	$I=3, J=4$	$1.954 \cdot 10^6$	$3.285 \cdot 10^8$
3	$I=4, J=5$	$6.457 \cdot 10^6$	$4.780 \cdot 10^8$
4	$I=5, J=6$	$7.211 \cdot 10^6$	$1.402 \cdot 10^9$
5	$I=6, J=7$	$1.582 \cdot 10^7$	$1.992 \cdot 10^9$
6	$I=7, J=8$	$2.537 \cdot 10^7$	$1.943 \cdot 10^9$
7	$I=7, J=10$	$2.943 \cdot 10^7$	$2.859 \cdot 10^9$
8	$I=7, J=12$	$3.523 \cdot 10^7$	$2.557 \cdot 10^9$
9	$I=7, J=14$	$3.298 \cdot 10^7$	$4.926 \cdot 10^9$

the passenger arrival rate and the passenger alighting rate at station  $J$  is changed to 0 and 1, respectively, because we assume that station  $J$  is the last station of the trip. The weight  $\lambda$  in the multi-objective function (19) is chosen as 1, which means that the energy consumption and the passenger travel time is equally important since they are normalized in the objective function. However, rail operators can take any positive and reasonable value for the weight  $\lambda$ . In addition, the nominal values of the passenger travel time and the energy consumption shown in Table III are obtained using random feasible train schedules for different cases. Note that the random feasible train schedule is actually with fixed departure headway at station 1 but with random running times (a value in  $[r_{\min}, 1.2 \cdot r_{\min}]$ ) later on.

### B. Simulation and results

We have applied the approaches proposed in Section IV to solve the real-time train scheduling problem. For the pattern search method, we used the patternsearch function in the global optimization toolbox of Matlab. The SNOPT solver implemented in the Matlab Tomlab toolbox is adopted for the SQP algorithm to solve the nonlinear non-convex train scheduling problem. In the ICP approach, the resulting smooth and convex problem is also solved by the SNOPT. In addition, the low-level optimization problem of the bi-level approach is also solved using SNOPT. The ga function in the global optimization toolbox of Matlab is applied for the high-level optimization of the bi-level approach with genetic algorithm. Furthermore, we use CPLEX, implemented through the cplex interface function of the Matlab Tomlab toolbox as MILP solver.

The schedule of trains for scenario 5 is shown in Figure 3, which is obtained by solving the train scheduling problem

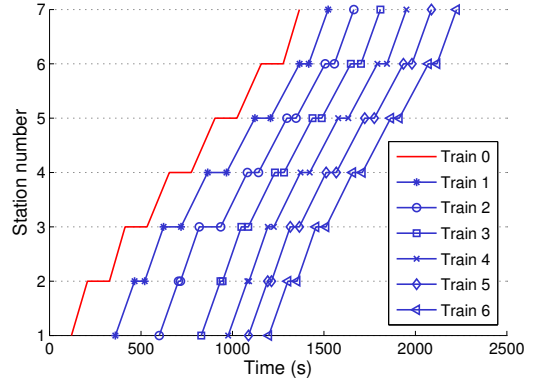


Fig. 3. The train schedule of scenario 5 with 6 trains and 7 stations obtained by the sequential quadratic programming approach

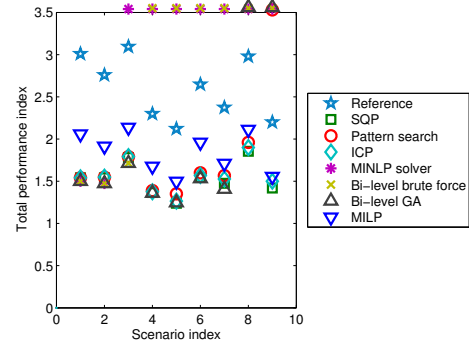


Fig. 4. Performance comparison of the solution approaches for the real-time train scheduling problem. In order to visualize the scenarios of which the computation cannot finish within 5 hours, the performance indices  $f_{\text{opt}}$  calculated using (19) of these scenarios are set larger than 3.5.

for 6 trains (i.e., train  $i \in \{1, 2, \dots, 6\}$ ) and 7 stations (i.e., station  $j \in \{1, 2, \dots, 7\}$ ) using the SQP approach. In addition, the corresponding running times, dwell times, and departure times are given in Table IV. The model formulation in this paper allows the presence of waiting passengers at platforms at the beginning of the scheduling period and allows trains to be running somewhere on the transit line. In this case study, we consider the train schedule at the start of a day. Train 0 is the first train entering the urban rail transit line for that day. There are no passengers left by train 0 because not too many passengers wait for the first train in the morning. As we can observe from Figure 3, the departure headways between train 1, train 2, and train 3 at station 1 are larger than those between the later trains. This is because of the schedule of train 0, which stops at each station with a dwell time of 120 s. Therefore, in order to satisfy the headway constraints at all stations, the departure headway at the station 1 must be much larger than the minimum headway 90 s.

In order to compare the performance of the schedules obtained by different approaches proposed in this paper, a reference schedule with a fixed departure headway defined, where the schedule of trains is the same as train 0 but with a constant departure headway 210 s, which is essentially the sum of the minimum headway and the maximum dwell time.

TABLE IV  
DEPARTURE TIMES, ARRIVAL TIMES, AND DWELL TIMES OF THE TRAIN SCHEDULE WITH 6 TRAINS AND 7 STATIONS OBTAINED BY THE SEQUENTIAL QUADRATIC PROGRAMMING APPROACH

Station		1	2	3	4	5	6	7
Train 0	Running time	87.7	85.7	121.7	129.7	132.7	88.7	-
	Dwell time	120	120	120	120	120	120	-
	Departure time	120	327.7	533.4	775.0	1024.7	1277.4	-
Train 1	Running time	105.3	102.8	146.0	155.7	159.2	106.5	-
	Dwell time	150.0	55.3	95.7	102.2	85.4	49.6	-
	Departure time	360.0	520.6	719.0	967.2	1208.2	1417.0	-
Train 2	Running time	105.3	102.8	146.0	155.7	159.2	106.5	-
	Dwell time	150.0	10.6	117.3	60.6	49.6	50.1	-
	Departure time	600.0	715.9	935.9	1142.5	1347.8	1557.2	-
Train 3	Running time	105.3	102.8	146.0	155.7	159.2	106.5	-
	Dwell time	139.5	11.4	37.6	49.8	50.0	55.2	-
	Departure time	829.5	946.2	1086.6	1282.3	1487.9	1702.3	-
Train 4	Running time	105.3	102.8	146.0	155.7	159.2	106.5	-
	Dwell time	57.2	8.7	32.9	50.0	55.2	52.4	-
	Departure time	976.7	1090.7	1226.3	1442.3	1633.1	1844.7	-
Train 5	Running time	105.3	102.8	146.0	155.7	159.2	106.5	-
	Dwell time	20.8	20.8	50.0	55.2	53.4	46.9	-
	Departure time	1087.5	1213.6	1336.3	1567.4	1775.5	1981.6	-
Train 6	Running time	105.3	102.8	146.0	155.7	159.2	106.5	-
	Dwell time	20.8	50.0	55.2	52.4	46.9	46.9	-
	Departure time	1198.3	1353.5	1511.5	1709.8	1912.4	2118.5	-

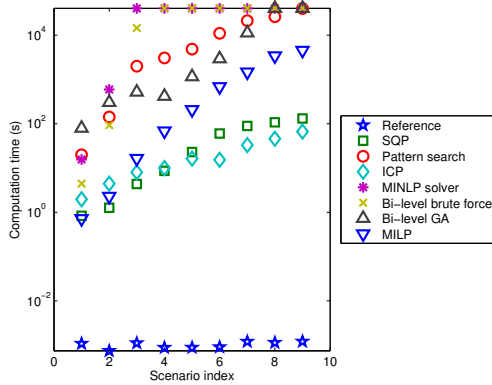


Fig. 5. The computation time of the solution approaches for the real-time train scheduling problem. In order to visualize the scenarios of which the computation cannot finish within 5 hours, the computation time of these scenarios are set larger than  $4 \times 10^4$  s.

Since the train scheduling problem is nonlinear and non-convex, multi-start methods should be applied. Hence, we have selected 10 feasible random initial points for the pattern search method, the SQP algorithm, and the ICP approach. When we solve the MINLP problem using bi-level optimization approach, the `fmincon` function in the lower optimization is executed for 10 feasible random initial points. For the MILP approach, only one feasible random initial point is needed to obtain the global minimum of the MILP problem.

The control performance  $f_{\text{opt}}$  and the computation time of these methods for the 9 scenarios are shown in Figure 4 and Figure 5. The value of the objective function is influenced by the nominal values and the weight  $\lambda$  in (19). A smaller value of the objective function means a better performance since we solve a minimization problem. Note that in the MILP approach the nonlinear objective function is approximated by piecewise affine functions. However, the objective value of the

original nonlinear objective function is calculated here using the optimal schedule obtained by the MILP approach. We set the upper bound of the computation time as five hours. If the computation cannot finish within 5 hours, no results are reported, so we cannot determine the control performance index. In order to visualize the scenarios of which the computation cannot finish within 5 hours, we set the total performance index of these scenarios larger than 3.5 as shown in Figure 4 and set the computation time larger than  $4 \times 10^4$  s as shown in Figure 5. It is noted that only the performance indices of the SQP algorithm and the ICP algorithm are reported for scenario 9 as shown in Figure 4, and the calculation of other algorithms for scenario 9 cannot finish within 5 hours. It is observed that the reference schedule has the worst control performance but also the lowest computation time. In addition, the performance of the MILP approach is worse compared with the other solution approaches that have similar control performance. It is observed from Figure 5 that the computation time of the ICP approach is higher than that of the SQP method for the smaller-sized train scheduling problems, i.e., scenario 1 to 4. However, for scenario 5 to 9, the computation time of the ICP approach is the smallest among all the solution approaches except the reference approach. For instance, the computation time of the ICP approach is just half of that of the SQP approach for scenario 9, where the computation time of the ICP approach and the SQP approach are around 66.701 s and 131.589 s, respectively. Furthermore, the values of the objective function of the random starting points, the solutions obtained by the ICP approach, and the solutions obtained by the SQP approach are illustrated in Table V. It can be observed that the function values of the random starting points are much higher compared with those of the solutions obtained by the ICP approach. In addition, we can observe that the function values of the ICP approach are quite close to those of the solutions of the SQP approach.

In order to evaluate the sensitivity to the weight in (19),

TABLE V

THE OBJECTIVE FUNCTION VALUES OF THE INITIAL POINTS, THE OBTAINED SOLUTIONS BY THE ICP APPROACH, AND THE OBTAINED SOLUTIONS BY THE SEQUENTIAL QUADRATIC PROGRAMMING APPROACH

Scenario	1	2	3	4	5	6	7	8	9
Initial Point	3.043	2.797	3.113	2.305	2.122	2.657	2.391	3.005	2.220
Solution of ICP	1.541	1.549	1.792	1.375	1.265	1.569	1.527	1.900	1.500
Solution of SQP	1.496	1.540	1.781	1.366	1.240	1.544	1.476	1.707	1.320

TABLE VI

THE ENERGY CONSUMPTION AND PASSENGER TRAVEL TIME FOR DIFFERENT WEIGHTS IN OBJECTIVE FUNCTION (19)

Weight $\lambda$ [-]	Energy consumption [J]	Passenger travel time [s]
0.1	$4.005 \cdot 10^9$	$2.294 \cdot 10^7$
1	$4.019 \cdot 10^9$	$2.265 \cdot 10^7$
10	$4.318 \cdot 10^9$	$2.212 \cdot 10^7$

Table VI gives the energy consumption and the passenger travel time obtained by the ICP approach for the weight taking values 0.1, 1, and 10. We can observe that with an increasing weight, the energy consumption increases and the passenger travel time decreases.

The performance of the schedule obtained by the ICP approach is close to the best performance of the schedules we obtained using alternative approaches. For smaller-sized train scheduling problems, the computation time of the SQP approach is the smallest except the reference approach, while for larger-sized train scheduling problems, the ICP approach produces a better trade-off between performance and computational complexity.

**Remark.** *When calculating the optimal train schedule using a train operation model with three phases, the energy consumption is  $4.077 \cdot 10^9$  J without the regenerative braking scheme. However, the energy consumption of the obtained schedule using the mean speed method in [27], [36], the energy consumption in the objective is  $9.5172 \cdot 10^8$  J, which is only 23.34% of that based on the three phases operation model. In addition, when optimizing the train schedule including the regenerative energy  $E_{i,j}^{\text{dec}}$  in (14), the energy consumption is  $2.0193 \cdot 10^9$  J (i.e., 49.53% of the energy consumption without regenerative braking) with the energy recovery rate  $\eta_{i,j} = 0.7$  and  $\varepsilon_{i,j} = 0$  for all  $i$  and  $j$ . Due to the short distances between stations in urban rail transit, the running times and distances of the acceleration phase and deceleration phase cannot be ignored anymore. Therefore, it is important to use an operation model with three phases for train scheduling.*

## VI. CONCLUSIONS

In the current paper, the real-time train scheduling problem for urban rail transit systems has been considered. We have proposed a new iterative convex programming (ICP) approach to solve this train scheduling problem. In addition, we have compared the ICP approach with a gradient-free nonlinear programming approach (in particular pattern search method), gradient-based nonlinear programming approach (in particular sequential quadratic programming (SQP)), a mixed integer nonlinear programming (MINLP) approach, and a mixed

integer linear programming (MILP) approach. Furthermore, the resulting MINLP problem is solved by the following three methods: the existing MINLP solver, a brute force bi-level optimization method, and a bi-level optimization with a genetic algorithm for the high-level integer optimization. A reference schedule with a fixed headway is also included in the case study, which shows that the reference schedule has the lowest computation time and the worst control performance. The optimal solutions obtained by the ICP approach, the pattern search method, the SQP approach, and the MINLP approach are close to each other for the train scheduling problem. The ICP approach can provide a better trade-off between performance and computational complexity.

Since the urban rail transit lines are operated with high frequency and are physically separated from each other, rail operators can schedule trains for each line in an optimal way based on the current passenger demands. Therefore we will apply the solution approaches proposed in this paper to solve the train scheduling problem in a rolling horizon way and will compare the performance with that of the train schedule used in practice. In our future work, we will consider train scheduling for an urban rail transit network where the origins and destinations of the passengers and the passenger transfer behavior from one line to the other will be characterized. Hierarchical optimization techniques, distributed optimization techniques, and fast optimization methods can be investigated for train scheduling in an urban rail transit network [52], [53]. Furthermore, we will investigate the effect of more detailed models (modeling the operation of trains in terminals, short turns, the distribution of on-board passengers and waiting passengers at platforms, the passengers appearing at platforms after the last train has passed, the passenger flows as described by origin-destination matrices) on the trade-off between performance and computational complexity. Moreover, an extensive comparison and assessment of the approaches proposed in this paper for a wide range of set-ups and scenarios will be a topic for future work. We will also characterize under what circumstances the ICP approach gives good initial results, characterize the convergence rate of the ICP approach, and also characterize the degree of optimality of the obtained solutions. Furthermore, in this paper we use the default settings for various algorithms as these have been shown to perform sufficiently well for a wide range of problems. The case study gives a general indication about the relative ranking and potential of various approaches and our results. In our future work, we will investigate this in more detail in order to obtain extra performance gains by choosing different settings and tuning all the parameters of the different solution approaches for a wide range of set-ups and scenarios.

## APPENDIX A

## TRANSFORMATION OF THE PRODUCT OF BINARY-VALUED VARIABLES AND REAL-VALUED VARIABLES

The product of binary-valued variable  $\delta$  and real-valued affine function  $\tilde{f}(\tilde{x})$  of  $\tilde{x}$  can be replaced by an auxiliary real-valued variable  $z = \delta\tilde{f}(\tilde{x})$ , which satisfies  $[\delta = 0] \Rightarrow [z = 0]$  and  $[\delta = 1] \Rightarrow [z = \tilde{f}(\tilde{x})]$  [45], [54]. Then  $z = \delta\tilde{f}(\tilde{x})$  is equivalent to

$$\begin{cases} z \leq \tilde{M}\delta, \\ z \geq \tilde{m}\delta, \\ z \leq \tilde{f}(\tilde{x}) - \tilde{m}(1 - \delta), \\ z \geq \tilde{f}(\tilde{x}) - \tilde{M}(1 - \delta), \end{cases} \quad (24)$$

where  $\tilde{M} = \max_{\tilde{x} \in \mathcal{X}} \tilde{f}(\tilde{x})$  and  $\tilde{m} = \min_{\tilde{x} \in \mathcal{X}} \tilde{f}(\tilde{x})$  with  $\mathcal{X}$  is the set of feasible  $\tilde{x}$ .

## APPENDIX B

## THE PRODUCT OF TWO REAL-VALUED VARIABLES

The product of two real-valued variables  $xy$  can be rewritten as

$$xy = \frac{1}{4}(x+y)^2 - \frac{1}{4}(x-y)^2. \quad (25)$$

Define  $\phi = x+y$  and  $\xi = x-y$ . Then we have  $xy = \frac{1}{4}\phi^2 + \frac{1}{4}\xi^2$ , where the quadratic terms  $\phi^2$  and  $\xi^2$  can be approximated by a piecewise affine function of the following form:

$$f_{\text{PWA}}(z) = \begin{cases} \alpha_1 z + \beta_1 & \text{for } z \leq Z_1, \\ \alpha_2 z + \beta_2 & \text{for } z > Z_1. \end{cases} \quad (26)$$

For each nonlinear term in the objective function, the values of  $\alpha$ ,  $\beta$ , and  $Z$  are optimized based on least-squares optimization. Furthermore, the approximation error can be reduced by focusing on the interesting part of the domain, which is done by introducing a weight function in the least-squares optimization. By introducing the binary variables and auxiliary variables, the product  $xy$  can be recast as a linear expression with linear constraints (see [33], [54] for more information).

## APPENDIX C

## EXTENSION: TRAIN SCHEDULING MODEL WITH STOP-SKIPPING

In this appendix, train scheduling with stop-skipping is considered, where trains may skip some stations to reduce the passenger travel time and energy consumption. In practice, the SEPTA line in Philadelphia and the rail system in Santiago, Chile, apply the stop-skipping schedule based on empirical analysis. The stop-skipping strategy that already exists in operational scheduling is mostly static, e.g. the A/B skip-stop strategy, where stations are divided into three types: A, B, and AB; A train services stop at A stations and AB stations, while B train services stop at B stations and AB stations. Major stations are usually labeled with the type AB; so all trains stop there. Now we show how the model of Section II can be extended to include *dynamic* stop-skipping, where the stop-skipping stations are not fixed, but are optimized based on the passenger demand. With the help of the information provided via smart mobile devices and the real-time displays and announcements at stations, we assume that passengers

can obtain enough information and board the right train. In the long run, both the passengers and the rail operator can get benefits from stop-skipping. The rail operator can benefit from the shorter cycle times, increased operating speed, and less energy consumption. For most of the passengers, the travel time is shortened and the on-board environment will be better, i.e., lower train occupation. However, the passengers at the skipped stations experience longer waiting time and thus a longer total travel time. Therefore, the skipping of trains at stations should be carefully coordinated to benefit passengers. For example, additional constraints can be considered by the scheduling of trains, such as two successive trains cannot skip the same station. In this case, the waiting time of passengers will be limited to an acceptable value.

Section C-A extends the train scheduling model in Section II to the model with stop-skipping. Section C-B describes the multi-objective cost function and the constraints of the real-time train scheduling problem with stop-skipping. Section C-C proposes that the problem with stop-skipping can be solved using the MINLP approach and the MILP approach in Section IV. The performance of the solution approaches is compared through a case study in Section C-D. A short discussion on train scheduling with stop-skipping is given in Section C-E.

## A. Train scheduling model with stop-skipping

We introduce a binary variable  $y_{i,j}$  to indicate whether train  $i$  stops at station  $j$  or not:

$$y_{i,j} = \begin{cases} 1 & \text{if train } i \text{ stops at station } j, \\ 0 & \text{if train } i \text{ skips station } j. \end{cases} \quad (27)$$

Hence, instead of (1) we get

$$d_{i,j} = a_{i,j} + y_{i,j}\tau_{i,j}. \quad (28)$$

Since train  $i$  may skip station  $j$  or station  $j+1$ , the running distance of the speed holding phase is then rewritten as

$$s_{\text{hold}} = s_j - y_{i,j} \frac{v_{i,j}^2}{2a_{i,j}^{\text{acc}}} - y_{i,j+1} \frac{v_{i,j}^2}{2a_{i,j}^{\text{dec}}}, \quad (29)$$

which means that if train  $i$  skips station  $j$ , then train  $i$  will run with the holding speed  $v_{i,j}$  in the running distance of the acceleration phase. Similarly, if train  $i$  skips station  $j+1$ , train  $i$  will run with the holding speed  $v_{i,j}$  in the distance of the deceleration phase. Note that we have

$$(1 - y_{i,j+1})(v_{i,j+1} - v_{i,j}) = 0, \quad (30)$$

since when train  $i$  skips station  $j+1$ , i.e.,  $y_{i,j+1} = 0$ , the operation of the train between station  $j$  and station  $j+2$  only contains three phases.

**Remark.** For the sake of simplicity of the expressions, now we assume that a train can skip all the stations. However, in practice, if the passenger alighting percentage and the passenger arrival rate are high, the train will not skip that station. This is conceptually equivalent to  $y_{i,j} = 1$  if  $\rho_j$  is high.

The running time of train  $i$  for segment  $j$  can be written as

$$r_{i,j} = \frac{s_j}{v_{i,j}} + y_{i,j} \frac{v_{i,j}}{2a_{i,j}^{\text{acc}}} - y_{i,j+1} \frac{v_{i,j}}{2a_{i,j}^{\text{dec}}}. \quad (31)$$



The remaining capacity of train  $i$  at station  $j$  immediately after the passengers alight is reformulated as

$$n_{i,j}^{\text{remaining}} = C_{i,\text{max}} - n_{i,j-1}(1 - y_{i,j}\rho_j)$$

instead of (7). Instead of (8), the number of passengers boarding train  $i$  at station  $j$  can be calculated using

$$n_{i,j}^{\text{boarding}} = \min(n_{i,j}^{\text{remaining}}, y_{i,j}w_{i,j}^{\text{waiting}}), \quad (32)$$

where  $y_{i,j}w_{i,j}^{\text{waiting}}$  is the number of passengers who want to get on train  $i$  at station  $j$ . Furthermore, the number of passengers at station  $j$  immediately after the departure of train  $i$  can be computed by (9). Instead of (10), the number of passengers on train  $i$  when it departs from station  $j$  is reformulated as

$$n_{i,j} = n_{i,j-1}(1 - y_{i,j}\rho_j) + n_{i,j}^{\text{boarding}}. \quad (33)$$

Compared with the standard model proposed in Section II, a binary variable is introduced for each train at each station to indicate whether a train stops or skips at a station for train scheduling with stop-skipping. The calculation of number of passengers, running times, etc. is related with the binary variable.

#### B. The real-time train scheduling problem with stop-skipping

For the real-time train scheduling problem with stop-skipping, the total energy consumption can be calculated as

$$E_{\text{total}} = \sum_{i=1}^I \sum_{j=1}^J (y_{i,j}E_{i,j}^{\text{acc}} + E_{i,j}^{\text{hold}} + y_{i,j+1}E_{i,j}^{\text{dec}}). \quad (34)$$

In addition, the passenger in-vehicle time for train  $i$  running from station  $j$  to  $j+1$  is reformulated as

$$t_{\text{in-vehicle},i,j} = n_{i,j}r_{i,j} + y_{i,j+1}n_{i,j}(1 - \rho_{j+1})\tau_{i,j+1} \quad (35)$$

instead of (17). The total travel time for the stop-skipping problem can then be calculated by (18). The weighted sum strategy to solve the multi-objective optimization shown in (19) can also be used for the train scheduling problem with stop-skipping. The constraints of the train scheduling problem with stop-skipping are shown in (2), (4)-(6), (9), (11), (28), and (30)-(33).

#### C. Solution approaches for the scheduling problem with stop-skipping

In the train scheduling model with stop-skipping, binary variables are introduced to indicate whether a train stops at a station or not. The MINLP approach and the MILP approach in Section IV-B can be directly applied to solve the train scheduling problem with stop-skipping since they can easily deal with binary variables. For the MILP approach, the nonlinear constraint (30) need to be rewritten as PWA constraints, where the nonlinear terms  $y_{i,j+1}v_{i,j+1}$  and  $y_{i,j+1}v_{i,j}$  can be transformed into PWA constraints based on the transformation given in Appendix A. The corresponding solvers and solution methods for the resulting MINLP and MILP problem can be used to solve the problem with stop-skipping.

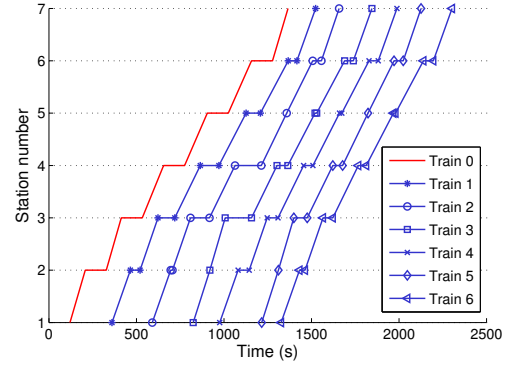


Fig. 6. The train schedule of scenario 5 with 6 trains and 7 stations for train scheduling with stop-skipping obtained by the bi-level optimization method with a genetic algorithm for the high-level optimization

#### D. Case study with stop-skipping

Now we consider the real-time train scheduling problem with stop-skipping, where a train may skip several small stations to reduce the passenger travel time and energy consumption. According to the information given in Table I, station 2 and station 5 are small stations since the passenger arrival rate and the passenger alighting proportion are smaller compared with other stations. In this case study, we allow trains to skip station 2 or/and station 5. We apply the MINLP approach and the MILP approach to solve the train scheduling problem with stop-skipping. The resulting MINLP problem is solved using a branch-and-bound algorithm, the brute force bi-level method, and the bi-level method with a genetic algorithm for the high-level binary optimization. The corresponding solvers and settings of these methods are chosen the same as those mentioned in Section V.

The schedule of trains with stop-skipping for scenario 5 with 6 trains and 7 stations is shown in Figure 6 and the corresponding running times, dwell times, and departure times are given in Table VII, which is obtained by the bi-level method with a genetic algorithm for the high-level optimization. In addition, the schedule of train 0 shown as the red line in Figure 6 is the same as that in Figure 3. It can be observed that train 3 and train 5 skip station 2 and train 2 and train 5 skip station 5 since trains are allowed to skip the small stations 2 and 5. Furthermore, the headways between trains at stations are influenced by the skipping of trains at station 2 and station 5 in order to satisfy the headway constraints at all stations.

The performance index and the computation time of the MINLP approach and the MILP approach for the 9 scenarios with stop-skipping are shown in Figure 7 and Figure 8. The reference schedule which has the lowest computation time and the worst performance is the same as that in Section V. Due to the errors introduced by the piecewise affine approximations of nonlinear terms in the objective function, the performance index of the MILP approach is much higher than other solution methods. The performance of the direct MINLP approach, the brute force bi-level optimization, and the bi-level optimization with a genetic algorithm for the real-time train scheduling problem with stop-skipping is similar to each other. However,

TABLE VII  
DEPARTURE TIMES, ARRIVAL TIMES, AND DWELL TIMES OF THE TRAIN SCHEDULE WITH STOP-SKIPPING FOR 6 TRAINS AND 7 STATIONS OBTAINED BY THE BI-LEVEL APPROACH

Station		1	2	3	4	5	6	7
Train 0	Running time	87.7	85.7	121.7	129.7	132.7	88.7	-
	Dwell time	120	120	120	120	120	120	-
	Departure time	120	327.7	533.4	775.0	1024.7	1277.4	-
Train 1	Running time	105.3	102.8	146.0	155.7	159.2	106.5	-
	Dwell time	150.0	55.3	95.7	106.0	81.5	49.6	-
	Departure time	360.0	520.6	719.0	971.0	1208.2	1417.0	-
Train 2	Running time	105.3	102.8	146.0	145.1	148.7	101.3	-
	Dwell time	140.7	10.3	108.2	150.0	0	49.6	-
	Departure time	590.7	706.3	917.2	1213.2	1358.4	1556.6	-
Train 3	Running time	95.6	87.6	146.0	155.7	159.2	106.5	-
	Dwell time	143.4	0	150.0	61.7	9.7	49.6	-
	Departure time	824.1	919.7	1157.2	1364.9	1530.3	1739.1	-
Train 4	Running time	105.3	102.8	146.0	151.9	155.5	106.5	-
	Dwell time	61.7	63.4	61.7	53.3	13.4	52.6	-
	Departure time	975.8	1144.5	1308.9	1508.2	1673.6	1881.7	-
Train 5	Running time	95.6	87.6	146.0	145.1	148.7	101.3	-
	Dwell time	150.0	0	76.9	56.1	0	52.9	-
	Departure time	1215.8	1311.3	1475.8	1677.9	1823.0	2024.5	-
Train 6	Running time	105.3	102.8	146.0	155.7	159.2	106.5	-
	Dwell time	20.8	31.2	56.1	44.9	12.1	56.8	-
	Departure time	1326.6	1463.0	1621.9	1812.8	1980.5	2196.5	-

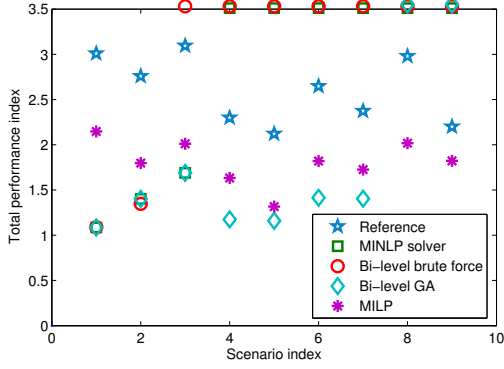


Fig. 7. Performance comparison of the solution approaches for the real-time train scheduling problem with stop-skipping. In order to visualize the scenarios of which the computation cannot finish within 5 hours, the performance indices  $f_{opt}$  calculated using (19) of these scenarios are set larger than 3.5.

the brute force bi-level approach can only report the results for scenario 1 and 2, and the MINLP solver can only report the results for scenarios 1 to 3. Since for the other scenarios, the computation of the brute force bi-level approach and the direct MINLP approach cannot finish within 5 hours. In addition, the bi-level optimization approach with a genetic algorithm for the high-level optimization cannot finish the calculation within 5 hours for scenarios 8 and 9. Therefore, the performance indices of the MINLP approach and the MILP approach are set higher than 3.5 as shown in Figure 7. It is observed that the MILP approach needs less computation effort but at the cost of much less optimal performance indices. The bi-level optimization methods with a genetic algorithm requires a longer computation time, but it has a better performance.

The comparison of the performance index for train scheduling without stop-skipping and train scheduling with stop-skipping, obtained by the bi-level optimization approach with a genetic algorithm for the high-level integer optimization, is

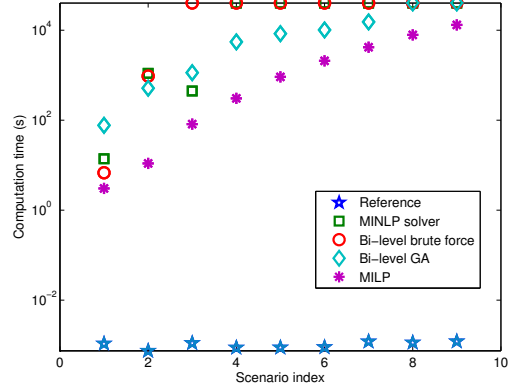


Fig. 8. The computation time of the solution approaches for the real-time train scheduling problem with stop-skipping. In order to visualize the scenarios of which the computation cannot finish within 5 hours, the computation time of these scenarios are set larger than  $4 \times 10^4$  s.

given in Table VIII. Skipping some small stations can reduce the travel time of most passengers due to the zero dwell time at small stations and lower running times. Hence, the performance of train scheduling with stop-skipping is better than that of train scheduling without stop-skipping. In addition, the energy consumption is also reduced since some trains do not need to decelerate and accelerate again at those small stations.

### E. Discussion

The real-time train scheduling model with stop-skipping has been formulated by introducing binary variables to indicate whether a trains stops at a station or not. The MINLP approach and the MILP approach are applied to solve this scheduling problem since they can handle integer variables. The case study shows that the control performance of the MILP ap-

TABLE VIII

COMPARISON OF THE TOTAL PERFORMANCE INDEX FOR TRAIN SCHEDULING WITH AND WITHOUT STOP-SKIPPING USING THE BI-LEVEL OPTIMIZATION WITH A GENETIC ALGORITHM FOR THE HIGH-LEVEL INTEGER OPTIMIZATION

Scenario index	1	2	3	4	5	6	7	8	9
Without stop-skipping	1.500	1.473	1.713	1.360	1.246	1.532	1.407	-	-
With stop-skipping	1.086	1.402	1.689	1.175	1.159	1.413	1.404	-	-

proach is worse than that of the MINLP approach, which includes 3 submethods, i.e., a MINLP approach, a brute force bi-level optimization, and a bi-level optimization approach with a genetic algorithm. Among these 3 submethods, the bi-level approach with a genetic algorithm offers the best trade-off between performance and computational efficiency. The computation of this approach is still tractable for small-sized problem (up to, i.e., 20 stations and 10 trains) using parallel processing. However, this approach is too slow for large-scale real-time applications. So in the future, new approaches need to be investigated to solve the train scheduling problem with stop-skipping efficiently.

#### ACKNOWLEDGMENT

Research supported by the Talents Foundation of Beijing Jiaotong University (Contract No. 2015RC003), the State Key Laboratory of Rail Traffic Control and Safety of Beijing Jiaotong University (Contract No. RCS2015ZQ002), the Beijing Municipal Commission of Education (Contract No. I15H100010), 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.

#### REFERENCES

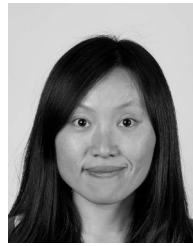
- [1] M. R. Bussieck, T. Winter, and U. T. Zimmermann, "Discrete optimization in public rail transport," *Mathematical Programming*, vol. 79, no. 1-3, pp. 415-444, Oct. 1997.
- [2] A. D'Ariano, M. Pranzo, and I. A. Hansen, "Conflict resolution and train speed coordination for solving real-time timetable perturbations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 208-222, Jun. 2007.
- [3] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "Bi-objective conflict detection and resolution in railway traffic management," *Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 79-94, Feb. 2012.
- [4] J. T. Krasemann, "Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances," *Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 62-78, Oct. 2012.
- [5] T. Dollevoet, F. Corman, A. D'Ariano, and D. Huisman, "An iterative optimization framework for delay management and train scheduling," Erasmus School of Economics (ESE), Rotterdam, Netherlands, Tech. Rep. EI 2012-10, 2012.
- [6] V. Cacchiani and P. Toth, "Nominal and robust train timetabling problems," *European Journal of Operational Research*, vol. 219, no. 3, pp. 727-737, Jun. 2009.
- [7] M. Khan and X. Zhou, "Stochastic optimization model and solution algorithm for robust double-track train-timetabling problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 81-89, Mar. 2010.
- [8] M. Shafia, M. Aghaee, S. Sadjadi, , and A. Jamili, "Robust train timetabling problem: Mathematical model and branch and bound algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 307-317, Mar. 2012.
- [9] W. Jiang, X. Chen, and Z. Zhong, "The impact of quality of services in chinese train control system on train delays analysis," in *Proceedings of the 2010 IEEE Vehicular Technology Conference Fall*, Ottawa, Canada, Sep. 2010, pp. 1-5.
- [10] L. Meng and X. Zhou, "Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach," *Transportation Research Part B: Methodological*, vol. 45, no. 7, pp. 1080-1102, Aug. 2011.
- [11] P. Vansteenwegen and D. V. Oudheusden, "Developing railway timetables which guarantee a better service," *European Journal of Operational Research*, vol. 173, no. 1, pp. 337-350, Aug. 2006.
- [12] X. J. Elberlein, N. H. M. Wilson, and D. Bernstein, "The holding problem with real-time information available," *Transportation Science*, vol. 35, no. 1, pp. 1-18, Feb. 2001.
- [13] N. Ghoneim and S. Wirasinghe, "Optimum zone structure during peak periods for existing urban rail lines," *Transportation Research Part B*, vol. 20, no. 1, pp. 7-18, Feb. 1986.
- [14] A. Ceder, "Optimal design of transit short-turn trips," *Transportation Research Record*, vol. 1221, pp. 8-22, Jan. 1989.
- [15] P. D. Site and F. Filippi, "Service optimization for bus corridors with short-turn strategies and variable vehicle size," *Transportation Research Part A*, vol. 32, no. 1, pp. 19-38, Jan. 1998.
- [16] W.-S. Lin and J.-W. Sheu, "Metro traffic regulation by adaptive optimal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 743-750, Dec. 2011.
- [17] J.-W. Sheu and W.-S. Lin, "Energy-saving automatic train regulation using dual heuristic programming," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1503-1514, May 2012.
- [18] S. Su, X. Li, T. Tang, and Z. Gao, "A subway train timetable optimization approach based on energy-efficient operation strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 883-893, Jun. 2013.
- [19] X. Yang, X. Li, Z. Gao, H. Wang, and T. Tang, "A cooperative scheduling model for timetable optimization in subway systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 438-447, Mar. 2013.
- [20] L. Fu, Q. Liu, and P. Calamai, "Real-time optimization model for dynamic scheduling of transit operations," *Transportation Research Record*, vol. 1857, pp. 48-55, Jan. 2003.
- [21] R. C. Wong, T. W. Yuen, K. W. Fung, and J. M. Leung, "Optimizing timetable synchronization for rail mass transit," *Transportation Science*, vol. 42, no. 1, pp. 57-69, Feb. 2008.
- [22] H. Niu and X. Zhou, "Optimizing urban rail timetable under time-dependent demand and oversaturated conditions," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 212-230, Nov. 2013.
- [23] F. J. Vázquez-Abad and L. Zubieta, "Ghost simulation model for the optimization of an urban subway system," *Discrete Event Dynamic Systems*, vol. 15, no. 3, pp. 207-235, Sep. 2005.
- [24] Y. Wang, B. De Schutter, T. van den Boom, B. Ning, and T. Tang, "Real-time scheduling in urban rail transit operations control," in *Proceedings of the 1st IEEE International Conference on Intelligent Rail Transportation (2013 IEEE ICIRT)*, Beijing, China, Aug.-Sep. 2013, paper 108.
- [25] C. M. Kwan and C. S. Chang, "Application of evolutionary algorithm on a transportation scheduling problem - the mass rapid transit," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 2, Edinburgh, Scotland, Sep. 2005, pp. 987-994.
- [26] C. Gassel and T. Albrecht, "The impact of request stops on railway operations," in *Proceedings of the 3rd International Seminar on Railway Operations Modeling and Analysis*, Zurich, Switzerland, Apr. 2009, pp. 1-15.
- [27] Y. Wang, B. De Schutter, T. van den Boom, B. Ning, and T. Tang, "Real-time scheduling for trains in urban rail transit systems using nonlinear optimization," in *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, The Netherlands, Oct. 2013, pp. 1334-1339.



- [28] M. M. Okrent, *Effects of Transit Service Characteristics on Passenger Waiting Times*. Northwestern University, Evanston, USA: Master thesis, 1974.
- [29] J. Simon and P. G. Furth, "Generating a bus route O-D matrix from on-off data," *Journal of Transportation Engineering*, vol. 111, no. 6, pp. 583–593, Nov. 1985.
- [30] J. Pachl, *Railway Operation and Control Second Edition*. Centralia, USA: Gorham Printing, 2009.
- [31] E. Khmelnitsky, "On an optimal control problem of train operation," *IEEE Transactions on Automatic Control*, vol. 45, no. 7, pp. 1257–1266, Jul. 2000.
- [32] R. Liu and I. M. Golovicher, "Energy-efficient operation of rail vehicles," *Transportation Research Part A: Policy and Practice*, vol. 37, no. 10, pp. 917–931, Oct. 2003.
- [33] Y. Wang, B. De Schutter, T. van den Boom, and B. Ning, "Optimal trajectory planning for trains – A pseudospectral method and a mixed integer linear programming approach," *Transportation Research Part C*, vol. 29, pp. 97–114, Apr. 2013.
- [34] S. Lu, S. Hillmansen, T. Ho, and C. Roberts, "Single-train trajectory optimization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 743–750, Jun. 2013.
- [35] T. Lin and N. H. M. Wilson, "Dwell time relationships for light rail systems," *Transportation Research Record*, vol. 1361, pp. 287–295, Feb. 1992.
- [36] K. Ghoseiri, F. Szidarovszky, and M. J. Asgharpour, "A multi-objective train scheduling model and solution," *Transportation Research Part B*, vol. 38, no. 10, pp. 927–952, Dec. 2004.
- [37] B.-R. Ke, M.-C. Chen, and C.-L. Lin, "Block-layout design using maximum ant system for saving energy on mass rapid transit systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 226–235, Apr. 2009.
- [38] J. Rosen, "Iterative solution of nonlinear optimal control problems," *SIAM Journal on Control*, vol. 4, no. 1, pp. 223–244, Feb. 1966.
- [39] J. Bezdek and R. Hathaway, "Some notes on alternating optimization," in *Advances in Soft Computing – AFSS 2002*, vol. 2275, Lecture Notes in Computer Science, Feb. 2002, pp. 288–300.
- [40] L. Koehler, W. Kraus, and E. Camponogara, "Iterative quadratic optimization for the bus holding control problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1568–1575, Sep. 2011.
- [41] C. Byrne, *Iterative Optimization in Inverse Problems*. Boca Raton, FL, USA: CRC Press, 2014.
- [42] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An interior point algorithm for large-scale nonlinear programming," *Society for Industrial and Applied Mathematics Journal on Optimization*, vol. 9, no. 4, pp. 877–900, Aug. 1999.
- [43] R. Hooke and T. A. Jeeves, "Direct search" solution of numerical and statistical problems," *Journal of the Association for Computing Machinery*, vol. 8, no. 2, pp. 212–229, Apr. 1961.
- [44] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4, no. 1, pp. 1–51, Jan. 1995.
- [45] H. P. Williams, *Model Building in Mathematical Programming (4th ed.)*. Chichester, England: John Wiley & Sons, 1999.
- [46] T. Berthold, G. Gamrath, A. M. Gleixner, S. Heinz, T. Koch, and Y. Shinano, "Solving mixed integer linear and nonlinear problems using the SCIP Optimization Suite," Zuse Institute Berlin, Takustr. 7, 14195 Berlin, ZIB-Report 12-17, 2012.
- [47] The Mathworks Inc., *Genetic Algorithm and Direct Search Toolbox for Use with MATLAB: Users Guide*. Natick, MA, USA: The Mathworks Inc., 2004.
- [48] —, *Optimization Toolbox for use with Matlab: Users Guide*. Natick, MA, USA: The Mathworks Inc., 1999.
- [49] J. Linderoth and T. Ralphs, "Noncommercial software for mixed-integer linear programming," *Optimization Online*, Jan. 2005.
- [50] A. Atamturk and M. W. P. Savelsbergh, "Integer-programming software systems," *Annals of Operations Research*, vol. 140, no. 1, pp. 67–124, Nov. 2005.
- [51] B. Zhang and Y. Lu, "Research on dwelling time modeling of urban rail transit," *Traffic & Transportation*, vol. 27, no. 12, pp. 48–52, Dec. 2011.
- [52] B. Chen and H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 485–497, Jun. 2010.
- [53] A. Koulakenzian, H. Soliman, T. Tang, and A. Leon-Garcia, "Robust traffic assignment in transportation networks using network criticality,"

in *Proceedings of the 2012 IEEE Vehicular Technology Conference Fall*, Quebec, Canada, Sep. 2012, pp. 1–5.

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



**Yihui Wang** received the B.Sc. degree in control engineering from Beijing Jiaotong University, Beijing, China, in 2007. She is currently working toward the Ph.D. degree in hybrid control of railway systems, with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. Her research interests include train optimal control, train scheduling, and hybrid systems control.



**Bin Ning** (M'94) received the Ph.D. degree Beijing Jiaotong University, Beijing, China, in 2005. He is currently a Professor with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, as well as the President of Beijing Jiaotong University. His research interests include intelligent transportation systems, train control, rail transport systems, fault-tolerant system design, fault diagnosis, system reliability, and safety studies. Prof. Ning is an Associate Editor for the IEEE Transactions on Intelligent Transportation Systems.



**Tao Tang** received the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China, in 1991. He is currently a Professor and the Director of the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University. He has participated in several major projects on high-speed trains funded by the Chinese government. He is one of the leading scientists in the field with extensive experience in high-speed train control, safety, and energy saving.



**Ton van den Boom** received the MSc and PhD degrees in Electrical Engineering from Eindhoven University of Technology, Eindhoven, The Netherlands, in 1988 and 1993, respectively. Currently he is an Associate Professor at Delft University of Technology, The Netherlands. His research interests are in the areas of linear and nonlinear model predictive control, control and identification of discrete event systems and hybrid systems with applications in railway, robotics, and printers.



**Bart De Schutter** (M'08–SM'10) received the Ph.D. degree in applied sciences (summa cum laude with congratulations of the examination jury) from K. U. Leuven, Belgium, in 1996. He is currently a Full Professor with the Delft Center for Systems and Control, Delft University of Technology, The Netherlands. His research interests include the control of intelligent transportation systems, hybrid system control, and multiagent systems. Bart De Schutter is an Associate Editor for *Automatica* and *IEEE Transactions on Intelligent Transportation Systems*.