Integrated Train Timetabling and Rolling Stock Scheduling Model Based on Time-Dependent Demand for Urban Rail Transit

Yixiang Yue (Corresponding author)
School of Traffic and Transportation, Beijing Jiaotong University,
No.3 Shang Yuan Cun, Hai Dian District, Beijing, China. 100044
Phone: 8613511054454; Fax: 86-10-51683124
Email: yixiangyue@yahoo.com

Juntao Han
School of Traffic and Transportation, Beijing Jiaotong University,
No.3 Shang Yuan Cun, Hai Dian District, Beijing, China. 100044
Phone: 8613810437903;
Email: 15120813@bjtu.edu.cn

Shifeng Wang
China Railway Electrification Survey Design & Research Institute Co. Ltd.,
No.33 Jiangdu Road, He Dong District, Tianjin, China. 300250
Phone: 8615210568355
Email: wangshifeng@tjedi.com.cn

Xiang Liu
Department of Civil and Environmental Engineering
Rutgers, The State University of New Jersey
CoRE 606, 96 Frelinghuysen Road, Piscataway, NJ 08854-8018
Email: xiang.liu@rutgers.edu

Word count: 4,796words text + (6 figures + 4 tables) x 250 words (each) = 6,796 words
Submission Date: August 1st, 2016
Paper submitted for peer review at the 96th Transportation Research Board Annual Meeting.
**ABSTRACT**

The congestion problem of urban transportation is becoming increasingly critical for many metropolises. The Urban Rail Transit (URT) system has attracted substantial attention due to its safety, high speed, high capacity and sustainability. With a focus on providing a holistic modelling framework for train scheduling problems, this paper proposes a novel optimization methodology that integrates both train timetabling and rolling stock scheduling based on time-dependent passenger flow demands. We particularly consider the trade-off between waiting times of passengers and operation costs of the URT system. Using train paths and rolling stock indicators as decision variables, this problem is formulated as a bi-level programming model. The simulated-annealing (SA) based heuristic algorithm is employed to solve the proposed model and generate approximate optimal solutions. Numerical examples are developed to demonstrate the performance of the proposed approaches. The calculation results and comparisons indicate that, even for the large-scale Beijing rail transit operations, the SA-based algorithm can efficiently produce the approximate optimal scheduling strategies within acceptable computational limits, demonstrating the practical value of our proposed approaches.

*Keywords*: Urban rail transit, Passenger flow, Train timetable, Rolling stock schedule, Bi-level programming, Simulated annealing algorithm
INTRODUCTION

With the emergence of a more social economy, many rail transit networks have recently been put into operation or are under construction in many metropolises to satisfy large passenger travel demands. Therefore, the efficient operation of rail transit lines has become an important issue for all URT systems.

The planning process for public transportation usually consists of several consecutive phases. The process begins with network design, which is typically followed by line planning, timetabling, and vehicle and crew scheduling (1). Obtaining a high quality railway operation plan will take several iterations. Although there are many available models and algorithms to address each step, the entire multi-step process is still very challenging and computationally cumbersome. Therefore, formulating and solving a large-scale integrated railway planning problem has long been a pursuit for academics and practitioners.

In compiling the train diagram for urban rail system, it is important to consider the interval time of trains, the circulation and the running route of trains (2). Min et al. (2011) consider the train-conflict resolution problem for which the optimal conflict-free timetable and use a column-generation method to compute it (3). Cormann et al. (2012) consider a bi-objective problem of minimizing train delays and missed connections to provide a set of feasible non-dominated schedules (4). Kecman et al. (2013) propose underlying algorithms automatically identify route conflicts with conflicting trains, determine accurate arrival and departure times/delays at stations (5). These papers focus on arranging trains depending on conflicts and use some useful algorithms to solve it.

Line planning is the first important strategic element in the railway operation planning process. Goossens et al. (2004, 2006) consider a model formulation of the line-planning problem where total operating costs are to be minimized and the model is solved with a branch-and-cut approach (6) (7). Kaspi and Raviv (2013) formulate an integrated line planning and timetabling model with the objective of minimizing both user inconvenience and operational costs (8). These researches lay the foundation for our study of timetable and rolling stock.

Developing optimization models for constructing periodic transit timetables and synchronized schedules is another research direction in the field. Carey and Crawford (2007) design a series of heuristics for finding and resolving trains conflicts so to satisfy various operational constraints and objectives (9). Zhou and Zhong (2005, 2006) formulate train scheduling models which consider segment and station headway capacities as limited resources, and developed algorithms to minimize both passenger waiting times and total train travel times (10) (11). Goverde (2007) describe a railway timetable stability measure by using a max-plus system theory and analyzed train delay propagation processes (12). Focusing on reducing passenger waiting time at stops and transfers, Liebchen (2008) adapts a periodic event-scheduling approach and a well-established graph model to optimize the Berlin subway timetable (13). Wong et al. (2008) concentrate on the synchronization between the different lines of an URT network to minimize passengers’ transfer times (14).

There are also a number of studies related to rolling stock scheduling. Schlake et al. (2011) conduct an analysis of the effect of lean production methods on main-line railway operations (15). Nielsen et al. (2012) deal with real-time disruption management of railway rolling stock (16). Thorlacius et al. (2015) propose an integrated rolling stock planning model that simultaneously takes into account all practical requirements for rolling stock planning (17).

A number of recent studies have put more attention on developing integrated optimization models for timetable and passenger flow or rolling stock schedule. Niu and Zhou (2013) focus on
optimizing a passenger train timetable in a heavily congested urban rail corridor (18). Another paper of them (2015) focuses on the demand-oriented passenger train-timetable optimization for a rail corridor under time-dependent demand and skip-stop patterns (19). Yang et al. (2015) propose a new collaborative optimization method for both train stop planning and train scheduling problems on the tactic level (20). Yue et al. (2016) propose an innovative methodology using a column-generation-based heuristic algorithm to simultaneously account for both passenger service demands and train scheduling (21).

Although a few researchers have attempted to explore integrated railway planning, simultaneously accounting for line planning, timetabling and rolling stock scheduling is rarely seen in the literature. Line plan, timetable and rolling stock usage of integrated optimization is the high-priority problem that must be solved on most busy URT lines. In this paper, we propose a new methodology to simultaneously account for total passengers’ waiting times, train timetabling cost and rolling stock usage for URT lines. The framework of our proposed methodology is illustrated in Figure 1. In our model, the inputs are URT line data and section-specific passenger flow. The decision variables correspond to the train path and rolling stock trajectory. In both the model and algorithm, the first step is to optimize train frequency while guaranteeing passenger flow constraints. Following this, the model simultaneously schedules train timetables and the rolling stock usage while guaranteeing constraints related to passenger flow, train flow and rolling stock size. The model output is a near-optimal train timetable rolling stock scheduling.

**FIGURE 1 Framework of integrated URT planning methodology**
We intend to address the following specific objectives:

1. Analyze time-dependent passenger flow, train timetabling and rolling stock scheduling in URT systems and discusses their interactions among them. The paper develops an optimization methodology that enables the integration of train timetabling, line planning, and rolling stock scheduling.

2. Include both loop and linear URT lines. In order to formulate an integrated optimization model, this paper proposes a general train flow model that can be applied to all types of URT lines.

3. Formulate a bi-level programming model for integrated train timetabling and rolling stock scheduling. The upper-level model optimizes train frequency and train timetables; minimize passengers’ waiting times and operation costs. The lower-level model schedules rolling stock to minimize the number of infeasible train paths and proposes a SA-based heuristic algorithm to solve the model.

4. Illustrates the use of an integrated optimization model and the SA-based algorithm to improve the timetable of typical real-world URT lines in Beijing, China.

The remainder of this paper proceeds as follows. First, we present detailed problem descriptions and assumptions for URT lines. Second, we develop an integrated train frequency, train timetabling and rolling stock scheduling model. Third, a SA-based algorithm to solve the model is proposed. Subsequently, we generate computational results from real-world instances of the Beijing URT, and demonstrate the effectiveness of our proposed model. Finally are principal conclusions and suggest possible future research directions.

**PROBLEM DESCRIPTION**

**Key elements in URT**

A train timetable defines train departure and arrival times at each station, which is an essential plan for the operation of a railway system. Each train departs from a depot when service begins and returns to a depot when service terminates. Stations that directly connect to depots are of greater importance and are named “**D-stations**” in the model. For example, in Figure 2, Xizhimen and Jishuitan are **D-stations** for Line 2, Tiantongyuanbei and Songjiazhuang are **D-stations** for Line 5, Bagou, Chedaogou and Songjiazhuang are **D-stations** for Line 10.

**FIGURE 2 Illustration of rail transit network in our cases**
There are three key elements in URT lines: passenger flow, train flow and rolling stock. Passenger flow is easy to understand. And we illustrate relationships between train flow and rolling stock in Figure 3. Each solid line represents a train and the lines of the same color denote rolling stock trajectories. Assume there are three rolling stocks in depot 1. The number of trains between station 1 and station 2 are determined by train intervals (passenger flow). Due to constraints on the number of available rolling stocks, some trains (denoted by solid lines) can be placed into service, whereas other trains (denoted by dashed lines cannot. Dashed lines need to be deleted in the final output train timetable.

![Figure 3 Relationship between train flow and rolling stocks](image)

**MATHEMATICAL FORMULATION AND ALGORITHM**

**Notations**

The general subscripts, input parameters and decision variables that are employed in our mathematical formulations are listed as follows.

<table>
<thead>
<tr>
<th>Subscripts</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>Set of stations: $1, \ldots, c$</td>
</tr>
<tr>
<td>$D$</td>
<td>Set of D-stations: $1, \ldots, d$, $D \subseteq C$</td>
</tr>
<tr>
<td>$E$</td>
<td>Set of time intervals: $1, \ldots, e$</td>
</tr>
<tr>
<td>$F$</td>
<td>Set of trains</td>
</tr>
<tr>
<td>$G$</td>
<td>Set of rolling stocks</td>
</tr>
<tr>
<td>$i, i', j, j'$</td>
<td>Station indices, $i, i', j, j' \in C$</td>
</tr>
<tr>
<td>$k, k'$</td>
<td>D-station indices, $k, k' \in D$</td>
</tr>
<tr>
<td>$t, s, r$</td>
<td>Time interval indices, $t, s, r \in E$</td>
</tr>
<tr>
<td>$f$</td>
<td>Trains index, $f \in F$</td>
</tr>
<tr>
<td>$g$</td>
<td>Rolling stocks index, $g \in G$</td>
</tr>
</tbody>
</table>
1 Parameters:

\[ o_{i,j}^t \] Number of passengers who leave station \( i \) for station \( j \) at time \( t \)

\[ q_{i,j}^t \] Passengers volume between station \( i \) and station \( j \) at time \( t \)

\[ L^{\text{max}} \] The maximum number of passengers for a train

\[ L^{\text{opt}} \] Expected number of passengers for a train

\[ d(f) \] The origin station of train path, \( d(f) \in D \)

\[ h \] Minimum headway (time interval) between two consecutive train paths at a station

\[ G_k^{\text{max}} \] The maximum rolling stocks of depot which connected to D-station \( k \)

\[ d(g) \] Depot which connected to origin station that rolling stock \( g \) starts to carry out a task

\[ e_{i,j} \] Running time between station \( i \) and station \( j \)

\[ e_{\text{ran}}^{\text{max}} \] The maximum running time for a rolling stock in a day

\[ e_{\text{dwell}}^{\text{max}} \] The maximum waiting time in a D-station for a rolling stock

\[ e_{\text{stop}}^{\text{min}} \] The minimum stopping time in the depot for a rolling stock

\[ w_i^t \] Waiting passengers of station \( i \) at time \( t \)

2 Decision variables:

\[ u_i^t(f) \] 1, if train \( f \) departs from station \( i \) at time \( t \)

0, otherwise

\[ v_i^t(f) \] 1, if train \( f \) arrives at station \( i \) at time \( t \)

0, otherwise

\[ x_{i,k}^{s,s}(g) \] 1, if rolling stock \( g \) departs from D-station \( k \) at time \( t \) and arrives at D-station \( k' \) at time \( s \)

0, otherwise

1, if rolling stock \( g \) waits at D-station \( k \) from time \( t \) to time \( s \)

0, otherwise

and \( s = t + 1 \)

\[ y_{i,k}^{s,s}(g) \] 1, if rolling stock \( g \) stops at depot connected to D-station \( k \) from time \( t \) to time \( s \)

0, otherwise

4

5 General train flow model for urban transit lines

Each rail passenger trip consists of an origin station, a destination station and travel time. For URT systems, the number of passengers who travel between two stations is important. Thus, we need to calculate this number using the number of passengers for each origin-destination matrix.

\[
q_{i,j}^{t} = \sum_{f \in \mathcal{F}_{i,j}} \sum_{s \in \mathcal{S}_{i,j}} o_{i,j}^{t,s} \quad \forall t \in T, i \in C
\]
Using equation 1, we can obtain the passenger volume between two adjacent stations. The passenger volume may differ among various sections. We use a reference value to replace the real value, as shown in Equation 2. Based on the section-specific passenger flow between two adjacent stations, we can obtain the maximum passenger volume of some successive sections.

\[ q_{ij} = \max\{q_{ij,1}, q_{ij,2}, \ldots, q_{ij,n}\} \quad \forall t \in E, i, j \in C \]  

(2)

In a URT network, the line topology structure can be divided into two main categories: linear lines and loop lines. Different types of lines have different rolling stock trajectories. Figure 4(a) shows the topologies of six common types of rail transit lines. Each line may have a single depot, such as type 1, type 3 and type 5, or multiple depots, such as type 2, type 4 and type 6. In types 1 and 2, depots are connected to origin or terminal stations; in types 3 and 4, depots are connected to intermediate stations. For mathematical modelling purposes, we modify linear lines to loop lines by regarding tracks in different directions as virtually different stations. Thus, we can transfer linear lines into a looped line topology with one or more depots. A rolling stock departs from a depot, takes some loops along the stations and returns to the origin depot, as shown in Figure 4(b). In this example, type 1 has one depot, type 2 has two depots and type 3 has three depots. Multiple-depot lines trajectories must be coordinated to satisfy minimum train headway constraints at D-stations.

(a) Origin topologies and modified topologies of URT lines

(b) Trajectories of rolling stocks

FIGURE 4 Illustration for topological structure of URT lines and rolling stock arrangement
Bi-Level Mathematical Model

We use a bi-level programming model to describe the problem for URT. The upper level model is used to optimize train timetables, including minimizing waiting times for passengers and reducing operating costs for URT system. The objective of the lower level model is to schedule rolling stock and minimize infeasible trains. When the solutions of the upper level model and the lower level model are feasible, the problem will be solved.

The upper level model

In the upper level model, we replace the waiting times of each passenger with the number of queuing passengers in each time interval. \( \alpha, \beta \) are weights for the cost of passengers and the operation cost. If decision makers want to save passengers waiting time and improve service levels, the value of \( \alpha / \beta \) should be larger; if decision makers want to reduce the train operating costs and improve operating income, the value of \( \alpha / \beta \) should be smaller.

\begin{equation}
obj_{\text{up}} = \min (\alpha \times obj_w + \beta \times obj_u)
\end{equation}

\begin{equation}
obj_w = \sum_{t \in D} \sum_{k \in E} w_t^k
\end{equation}

\begin{equation}
obj_u = \sum_{f \in F} \sum_{k \in D} \sum_{r \in E} u(f)
\end{equation}

1) “Passengers flow” constraints

The number of passengers who wait at time \( t \) is the number of waiting passengers at the prior time \( t-1 \) plus the number of arrival passengers at time \( t \) minus the number of departure passengers at time \( t \). During saturation, the number of departing passengers is equal to \( L^m \), and the number of departing passengers in non-saturation is less than \( L^m \). Considering cases of saturation and non-saturation, we use “greater than” in equation (6), equation (7) and equation (8).

\begin{equation}
w_t^k \geq 0 \quad \forall t \in E, k \in D
\end{equation}

\begin{equation}
w_t^k \geq q_{k,k'}^t - \sum_{f \in F} u^t(f) \times L^m \quad \forall k,k' \in D
\end{equation}

\begin{equation}
w_t^k \geq w_{t-1}^k + q_{k,k'}^t - \sum_{f \in F} u^t(f) \times L^m \quad \forall t \in E \setminus \{1\}, k,k' \in D
\end{equation}

2) “Train flow” constraints

Two trains that travel in the same direction cannot depart from a station at the same time, and a reasonable time interval between the two trains is needed (typically, the minimum headway is based on the shortest braking distance between the trains given the train types and the signaling systems). The interval between two consecutive train departures from the same station \( i \) must be greater than or equal to the minimum departure headway \( h \).

\begin{equation}
\sum_{f \in F \cap (r,s+h)} u(f) \leq 1 \quad \forall r \in E, k \in D
\end{equation}

\begin{equation}
\sum_{f \in F \cap (r,s+h)} v(f) \leq 1 \quad \forall r \in E, k \in D
\end{equation}

Train flow \( f \) departs from \( D\text{-station} \ k \) at time \( t \) and it arrives at \( D\text{-station} \ k' \) at time \( t + e_{k,k'} \). If \( k' \neq d(f) \), it departs from \( D\text{-station} \ k' \) at time \( t + e_{k,k'} \).

\begin{equation}
u^t(f) = v^{t+e_{k,k'}}(f) \quad \forall t \in E, k, k' \in D, f \in F
\end{equation}

\begin{equation}v^t(f) = u^t(f) \quad \forall t \in E, k \in D \setminus \{d(f), f \in F
\end{equation}

For all train paths, the number of departure times and arrival times are equal.

\begin{equation}\sum_{k \in D \cap E} \sum_{r \in E} v^t(f) = \sum_{k \in D \cap E} \sum_{r \in E} u^t(f) \quad \forall f \in F
\end{equation}
The lower level model

The lower level model is used to schedule rolling stock based on train flows. The objective function is to minimize the number of infeasible train paths.

\[
\text{obj}_{-\text{low}} = \text{Min } \text{obj}_{-x}
\]

\[
\text{obj}_{-x} = \sum_{k\in D} \sum_{f\in E} u'_i(f) - \sum_{k\in D} \sum_{f\in E} x'_{k,i}(g) \quad \forall u'_i(f) = 1, f \in F, g \in G
\]

1) “Flow conservation” constraint

For any space-time node \((k,t)\), the number of outflows is no more than one.

\[
\sum_{i\in E} \sum_{k\in D} x'_{k,i}(g) + y'_{i,j}(g) + z'_{i,j}(g) \leq 1 \quad \forall t \in E, k \in D, g \in G
\]

2) “Train flow” constraints

For any space-time node \((k,t)\), the number of train trajectories is less than the number of train paths.

\[
\sum_{i\in E} \sum_{k\in D} x'_{k,i}(g) \leq u'_i(f) \quad \forall t \in E, k \in D, d(f) = d(g)
\]

3) “Rolling stock” constraints

For any space-time node \((k,t)\), trains cannot be overtaken in a station, and only one train at a time may remain in a station.

\[
\sum_{i\in E} \sum_{k\in D} x'_{k,i}(g) + \sum_{g\in E} y'_{i,j}(g) \leq 1 \quad \forall r \in E, k \in D
\]

To solve the model, we tried commercial solvers such as GAMS and CPLEX, and we also designed our own algorithm. Because of the scale of the problem, we select a simulated-annealing-based algorithm to solve our proposed models. Its evaluation function is

\[
\text{obj}_{-A} = \alpha \times \text{obj}_{-w} + \beta \times \text{obj}_{-u} + \gamma \times \text{obj}_{-x}, \quad \text{where } \gamma \text{ is the weight for the number of infeasible train paths.}
\]

In the early stage of iteration, \(\gamma \times \text{obj}_{-x}\) has little influence on the objective function and reduces the limitation of feasible solution. In the late stage of iteration, \(\gamma \times \text{obj}_{-x}\) has great influence on the objective function, this can ensure the rolling stock usage plan. Figure 5 shows the framework for the simulated annealing algorithm.
We provide detailed procedures for the simulated annealing algorithm to solve the model:

**Step 1.** Initialize parameters and variables.
Input date of line and passenger volume, initialize parameters of the algorithm, and initialize variables: $u(f), v(f)$ and $w_i^f$.

**Step 2.** Obtain an initial feasible solution.
We use a blank train timetable as the initial feasible solution: $A$.

**Step 3.** Update the current feasible solution, $A'$.
We add and reduce trains based on a small probability to obtain a new feasible solution.

**Step 4.** Compute an evaluation function for the new feasible solution, $A'$.
Compute $obj_{w'}$, $obj_{u'}$.
Compute $obj_{x'}$. We use a neighborhood search method to solve $obj_{x'}$.
Calculate the evaluation function: $obj_{A'} = \alpha \times obj_{w'} + \beta \times obj_{u'} + \gamma \times obj_{x'}$.
Calculate the difference: $\Delta obj = obj_{A} - obj_{A'}$

**Step 5.** Update the current best solution:
Determine whether the new solution $A'$ is accepted based on the Metropolis-Hastings algorithm. If accepted, $A \leftarrow A'$

**Step 6.** Stop or continue.
Two rules can end the algorithm: 1) Iteration number reaches the maximum limit, 2) Stable iteration number reaches the maximum limit and the number of infeasible train paths equals “0”. If one rule is satisfied, we proceed to **Step 8**; otherwise, we proceed to

**Step 7.**

**Step 7.** Update temperature
We update the temperature using a method of “two stage-exponential decline,” which can ensure efficiency and accuracy.

**Step 8.** Schedule rolling stock.
We use a neighborhood search method to schedule rolling stock.

**Step 9.** Present the output.

**CASE STUDY: LINE 2 OF BEIJING RAIL TRANSIT NETWORK**
We use a real-world instance (Subway Line 2) from Beijing’s rail transit network to demonstrate the application of our optimization model and algorithm. The scheduling algorithms are
implemented in Microsoft Visual Studio 2010 on Windows 7 OS. All experiments are conducted on a PC with an Intel Core Duo 2.93 GHz CPU and 4 GB RAM.

Table 1 shows the parameters of the algorithm. $d$ is the number of depots and $b$ represents the number of time intervals of 1 min. (For example, when $b=1$, a time interval is 60 seconds; when $b=4$, a time interval is 15 seconds.) The time span for the train operation considered in this paper is 20 hours (1,200 mins) from 5:00 am to 1:00 am the next day. The time interval of passenger flow is from 5:00 am to 11:00 pm for a total time of 18 hours (1,080 mins).

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of time intervals</td>
<td>$1,200 \times b$</td>
<td>Number of time intervals</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>1</td>
<td>Weight of passenger waiting time</td>
</tr>
<tr>
<td>$\beta$</td>
<td>$2 \times b \times L^{\text{exp}}$</td>
<td>Weight of train frequency</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>$500,000 \times d$</td>
<td>Weight of infeasible trains</td>
</tr>
<tr>
<td>$M_0$</td>
<td>$500,000 \times b \times d$</td>
<td>Initial temperature</td>
</tr>
<tr>
<td>$\rho^{\text{rise}}$</td>
<td>0.3</td>
<td>Heating coefficient</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.95</td>
<td>Cooling coefficient</td>
</tr>
<tr>
<td>$N_0$</td>
<td>$30,000 \times b \times (d+3)$</td>
<td>Maximum number of iterations</td>
</tr>
<tr>
<td>$N^{\text{stop}}$</td>
<td>$N_0 / 200$</td>
<td>Iteration number of terminating algorithm for no longer improvement</td>
</tr>
<tr>
<td>$N_{1\rightarrow 2}$</td>
<td>$N_0 / 10,000$</td>
<td>Iteration number in initial stage</td>
</tr>
<tr>
<td>$N_{2\rightarrow 3}$</td>
<td>$N_0 / 500$</td>
<td>Iteration number in medium stage</td>
</tr>
<tr>
<td>$N^{\text{update}}$</td>
<td>$N_0 / 200$</td>
<td>Iteration number for temperature to update</td>
</tr>
<tr>
<td>$N^{\text{rise}}$</td>
<td>$N_0 / 4$</td>
<td>Iteration number for temperature to increase</td>
</tr>
<tr>
<td>$p_1^{\text{change}}$</td>
<td>$0.02/b/d$</td>
<td>Update probability in initial stage</td>
</tr>
<tr>
<td>$p_2^{\text{change}}$</td>
<td>$0.005/b/d$</td>
<td>Update probability in medium stage</td>
</tr>
<tr>
<td>$p_3^{\text{change}}$</td>
<td>$0.001/b/d$</td>
<td>Update probability in later stage</td>
</tr>
<tr>
<td>$\epsilon_{\text{run}}^{\text{max}}$</td>
<td>$720 \times b$</td>
<td>Maximum running time</td>
</tr>
<tr>
<td>$\epsilon_{\text{dwell}}^{\text{max}}$</td>
<td>$3 \times b$</td>
<td>Maximum dwelling time in D-stations</td>
</tr>
<tr>
<td>$\epsilon_{\text{stop}}^{\text{min}}$</td>
<td>$90 \times b$</td>
<td>Minimum stopping time in depots</td>
</tr>
</tbody>
</table>

Table 2(a) lists the data of Beijing rail transit lines 2. Line 2 of the Beijing railway transit network is a loop line. There is only one depot (refer to green line in Figure 2) that connects with Xizhimen station and Jishuitian station. We consider Xizhimen station as a D-station in this case. The minimum departure headway between two consecutive train paths is 2 minutes. In this case,
We consider the clockwise direction of this line in our (model/parameter-use one). Table 2(b) shows passenger demand in workdays.

**TABLE 2 Input data of Beijing rail transit Line 2**

(a) Basic data of line 2

<table>
<thead>
<tr>
<th>Line</th>
<th>Number of depots</th>
<th>Number of rolling stocks</th>
<th>Number of stations</th>
<th>Running time (minutes/lap)</th>
<th>Line length (km/lap)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>25</td>
<td>18</td>
<td>46</td>
<td>23.1</td>
</tr>
</tbody>
</table>

(b) Passenger demand of line 2

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>5:00~7:00</th>
<th>7:00~9:00</th>
<th>9:00~11:00</th>
<th>11:00~17:00</th>
<th>17:00~19:00</th>
<th>19:00~22:00</th>
<th>22:00~23:00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workday (persons/min)</td>
<td>50</td>
<td>750</td>
<td>500</td>
<td>400</td>
<td>600</td>
<td>400</td>
<td>50</td>
</tr>
<tr>
<td>Weekend (persons/min)</td>
<td>20</td>
<td>300</td>
<td>450</td>
<td>400</td>
<td>500</td>
<td>400</td>
<td>50</td>
</tr>
</tbody>
</table>

**Results of Line 2 and analysis.**

Figure 6(a) shows the results of passenger flow during workdays. We counted the average number and the maximum number of waiting passengers in 30 minutes. Operators can make reasonable decisions based on passenger data. The number of waiting passengers reaches the maximum value of 1550 between 7:30am and 8:00am. The train timetable of Beijing rail transit Line 2 for workdays is shown in Figure 6(b). Due to larger passenger flows, the train paths on the train diagram are more intensive, and train operating frequency reaches its maximum limit during peak hours (7:00-9:00, and 17:00-19:00). Fewer trains operate during off-peak hours. The solution not only avoids wasting resources and underutilization of transport capacity, but also decreases the cost of train operation. The density of the train paths adequately reflects the change in traffic flow.
Table 3 shows the rolling stock schedule of the depot that is connected to Xizhimen station for workdays. The first column represents the rolling stocks number. The second column shows total running time for each rolling stock. The third column represents the timetable of every rolling stock. For example, [8 54] means that for the first rolling stock, it begins its first service at the 8th minute, and ends at the 54th minute (service time interval is 46 minutes). Similarly, its last service starts at the 962nd minute and ends at the 1008th minute. As mentioned before, the time span of rolling stock operation is 1200min. The results reveal that the proposed model and algorithm applies to rail transit ring line of multi-depots and can get an approximate optimal solution.
Efficiency analysis of the algorithm

We also apply a similar optimization approach to Line 5 and Line 10 in Beijing. To evaluate the efficiency of the proposed algorithm, we try to use two service packages (GAMS and ILOG CPLEX) to solve the model and compare the results of different methods. Due to a large number of variables, we figure out the upper model using GAMS because CPLEX is not suitable to solve the complete model. We only list the calculation results of the upper model by GAMS, CPLEX and SA (γ=0). We employ the calculation results of GAMS as a benchmark and make comparative analysis. Table 4 lists the comparison of the different solving methods. Three cases
are employed: Line 2 on workdays, Line 5 on workdays and Line 10 on workdays. For a large-sized problem, our proposed algorithm outperforms the commercial software.

**TABLE 4 Comparison of different solving methods**

<table>
<thead>
<tr>
<th>Case</th>
<th>Number of Depot</th>
<th>Solver</th>
<th>obj_A</th>
<th>Comparison of object value</th>
<th>Computing time(s)</th>
<th>Comparison of computational time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line 2, workday</td>
<td>1</td>
<td>Upper level model/GAMS</td>
<td>1,267,350</td>
<td>1</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upper level model/CPLEX</td>
<td>1,267,350</td>
<td>1</td>
<td>124</td>
<td>10.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upper level model/SA</td>
<td>1,286,750</td>
<td>1.01</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bi-level model /SA+ neighbor hood searching</td>
<td>1,306,215</td>
<td>1.03</td>
<td>15</td>
<td>1.25</td>
</tr>
<tr>
<td>Line 5, workday</td>
<td>2</td>
<td>Upper level model/GAMS</td>
<td>3,967,667</td>
<td>1</td>
<td>4,239</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upper level model/CPLEX</td>
<td>unavailable</td>
<td>unavailable</td>
<td>unavailable</td>
<td>unavailable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upper level model/SA</td>
<td>3,788,936</td>
<td>0.95</td>
<td>95</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bi-level model /SA+ neighbor hood searching</td>
<td>3,808,960</td>
<td>0.96</td>
<td>126</td>
<td>0.03</td>
</tr>
<tr>
<td>Line 10, workday</td>
<td>3</td>
<td>Upper level model/GAMS</td>
<td>15,533,020</td>
<td>1</td>
<td>8,769</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upper level model/CPLEX</td>
<td>unavailable</td>
<td>unavailable</td>
<td>unavailable</td>
<td>unavailable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Upper level model/SA</td>
<td>13,119,341</td>
<td>0.84</td>
<td>814</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bi-level model /SA+ neighbor hood searching</td>
<td>13,358,397</td>
<td>0.86</td>
<td>1,004</td>
<td>0.11</td>
</tr>
</tbody>
</table>

The following are several observations based on the results presented in Table 4:

1) For the case of Line 2 on workdays, the results of the upper level model using GAMS, CPLEX and SA are similar, but the computing time by CPLEX is ten times faster than the computing time by GAMS, and SA is also much faster than GAMS. The computing time of the bi-level model by SA and neighborhood searching is slightly longer than the computing time of the bi-level model by GAMS and the results are worse than GAMS by 2% considering the lower level of the rolling stock schedule.

2) For the large-scale problems, such as cases 2 and 3, CPLEX cannot obtain a solution within a reasonable computing time. In the case of Line 5 with a time interval of 30s and two depots, GAMS can solve the model; however, the computing time exceeds 1 hour. SA and neighborhood searching can obtain a better solution within 3 minutes.

3) In the case of Line 10 with a time interval of 15s and 3 depots, GAMS can barely obtain a solution within 2 hours, whereas SA and neighborhood searching can obtain the solution within 16 minutes and 44 seconds. The objective function value is greater than 14%.
From these observations, it appears that SA, as a meta-heuristics, is fast but cannot guarantee global optimality. For a practical, large-scale problem, SA may be a promising approach to yield adequate enough solutions (may not be perfect) given a reasonable time span.

CONCLUSIONS

This paper proposes a new mathematical model for the optimization of train service plans, train timetables and rolling stock schedules for URT. This model can simultaneously reduce passenger waiting time and train operation costs while also improving the utilization of train sets.

First, we introduce three key elements in URT lines and propose a general train flow model, which can be employed across all topologies of train lines. Then we apply a bi-level programming model to formulate the scheduling problem for URT. The upper level model optimizes train timetables by minimizing waiting times for passengers and operation costs for URT systems. The lower level model is used to schedule rolling stock by minimizing the number of infeasible train paths. We use a simulated-annealing-based heuristic algorithm to solve the large-scale model. In a case study of the Beijing rail transit network, we test our optimization mathematical model and algorithm. The calculation results indicate that the proposed model and algorithm can obtain reasonable schedule planning. In particular, our new, integrated algorithm can rapidly obtain a near-optimal solution and best utilize rolling stocks, which renders it useful for complex real-world applications. A similar optimization methodology can also be adapted for high-speed and freight rail lines.

Our future research will focus on three major areas. First, the neighborhood searching method for vehicle scheduling is not a global optimization method; we aim to improve the algorithm performance. Second, the model can be extended to account for train skip-stop patterns and long/short trains. Last, we will refine and validate the proposed model with observed data from URT systems in Beijing.
REFERENCES


20. Yang, L., Qi, J., Li, S., Gao, Y. 2015. Collaborative optimization for train scheduling and train stop planning on high-speed railways. *Omega*.