COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING

Computer-Aided Civil and Infrastructure Engineering **32** (2017) 856–873

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

Yixiang Yue* & Juntao Han

School of Traffic and Transportation, Beijing Jiaotong University, Beijing, China

Shifeng Wang

China Railway Electrification Survey Design & Research Institute Co. Ltd., Tianjin, China

&

Xiang Liu

Department of Civil and Environmental Engineering, Rutgers, the State University of New Jersey, NJ, USA

Abstract: *The congestion of urban transportation is becoming an increasingly critical problem 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 a holistic modeling framework for train scheduling problems, this article 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 tradeoff between waiting times for passengers and the train frequency of the URT system. By using train paths and rolling stock indicators as decision variables, this problem is formulated as a bi-level programming model. A simulated-annealing (SA)-based heuristic algorithm is employed to solve the proposed model and generate approximate optimal solutions. In the case study of Line 10 of Beijing Subway, GAMS (The General Algebraic Modeling System) with the IBM ILOG CPLEX Optimization Studio (CPLEX) solver can barely obtain a solution in more than 2 hours, whereas the SA-based heuristic can obtain the solution within 16 minutes and 44 seconds with the objective value improved by more than 14%. The calculation results and comparisons indicate that the SA-based heuristic can efficiently produce approximate optimal scheduling strategies; these findings demonstrate the practical value of our proposed approaches.*

*To whom correspondence should be addressed. E-mail: *yxyue@bjtu.edu.cn*.

1 INTRODUCTION

With the emergence of a more social economy, many rail transit networks have been put into operation or construction in many metropolises. Urban Rail Transit (URT) systems play an important role in urban public transportation, as millions of passengers use them for their daily commute. The frequency of train operations is becoming very high, especially in large cities such as Beijing, Tokyo, New York, and Paris, where the minimum headway between trains is close to 2 minutes for certain busy lines. Therefore, the planning process for urban rail transit systems is becoming more and more significant for reducing the operation costs of rail operators and for guaranteeing passenger satisfaction.

The planning process for public transportation usually consists of several consecutive phases. The process begins with network design, typically followed by line planning, timetabling, and vehicle and crew scheduling.

DOI: 10.1111/mice.12300

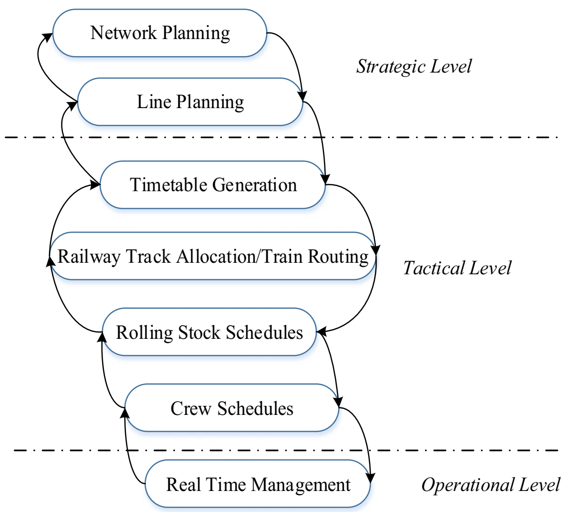


Fig. 1. Railway planning process.

Yang et al. (2016) present a detailed analytical process of railway planning in Figure 1. The process is divided into several step-by-step subproblems. In this process, once a prespecified train timetable plan is changed according to realistic requirements, a new rolling stock schedule for the railway needs to be regenerated to satisfy the varied train timetable constraints. This process necessarily increases the complexity of railway operations. Obtaining a high-quality railway operation plan requires several iterations. Although many models and algorithms are available to address each step, the entire multi-step process remains highly challenging and computationally cumbersome. Therefore, formulating and solving a large-scale integrated railway planning problem has been a long-term pursuit for academics and practitioners. To produce a comprehensive operational plan for the URT system, we are particularly interested in designing an optimization methodology that integrates both train timetabling and rolling stock scheduling based on time-dependent passenger flow demands. This topic has not attracted sufficient attention in the literature.

1.1 Literature review

Network design is the first stage of the railway planning process. Some scholars (Fan and Machemehl, 2006; Kermanshani et al., 2010) use simulated annealing algorithms to solve the transit route network design problem. The next railway operation planning process is line planning. Goossens et al. (2004, 2006) consider a model formulation of the line planning problem in which the total operating costs are to be minimized, and the model is solved with a branch-and-cut approach. Lin and Ku (2014) propose an integer program for the stopping pattern optimization problem and develop two genetic algorithms to solve the problem. Canca et al. (2017) develop an adaptive large neighborhood search algorithm for the integrated railway rapid transit network design and the line planning problem. These studies address the strategic problems of railway planning, whereas ours focuses on the tactical level. They lay the foundation for our study of timetables and rolling stock scheduling.

The development of optimization models for constructing timetables and synchronized schedules is another research direction within the field. Caprara et al. (2002) propose a graph theoretical formulation for the timetable problem and build an integer linear programming model that is relaxed in a Lagrangian way. Zhou and Zhong (2005, 2007) formulate train scheduling models that consider segment and station headway capacities as limited resources and develop algorithms to minimize both the expected passenger waiting times and the total train travel times. Carey and Crawford (2007) design a series of heuristics for identifying and resolving train conflicts to satisfy various operational constraints and objectives. Goverde (2007) describes a railway timetable stability measure using max-plus system theory and analyze the processes of train delay propagation. Wong et al. (2008) concentrate on the synchronization between the different lines of a URT network to minimize passengers' transfer times. Albrecht (2009) presents a two-level approach (computation of transport offer, timetable design) and the results obtained from fully automatic offer planning and timetabling for a suburban railway. Barrena et al. (2014) propose two mathematical programming formulations that generalize the nonperiodic train timetabling problem on a single line under a dynamic demand pattern and introduce a fast adaptive large neighborhood search metaheuristic to solve the problem. These papers focus on train timetabling without considering line planning and rolling stock scheduling.

Many other scholars (Corman et al., 2010; Min et al., 2011; Kecman and Goverde, 2012; Castillo et al., 2015, 2016) focus on train rescheduling and timetable optimization. Dollevoet et al. (2012, 2014b,b) have performed extensive research on the delay management problem. They model it with rerouting and develop several heuristics to tackle large-scale real-world instances. In addition, they propose an iterative optimization approach that iteratively solves a macroscopic delay management model and a microscopic train scheduling model. Corman et al. (2012) consider a bi-objective problem of minimizing train delays and missed connections to provide a set of feasible nondominated schedules to support this decision process. Another paper by Corman et al. (2014) studies the disturbance robustness of a timetable to assess the quality of a train schedule. Our paper focuses on timetable planning without accounting for train delay. Delay management will be the next focus of our research.

Numerous studies investigate rolling stock scheduling. Nielsen et al. (2012) address the real-time disruption management of railway rolling stock. Thorlacius et al. (2015) propose an integrated rolling stock planning model that simultaneously considers all practical requirements for rolling stock planning. Haahr et al. (2016) employ two approaches (mixed integer linear program and a column and row generation approach) to create daily schedules and test their real-time applicability through tests with different disruption scenarios.

A number of recent studies have focused on integrated optimization. Kaspi and Raviv (2013) formulate an integrated line planning and timetabling model with the objective of minimizing both user inconvenience and operational costs. Some others (Yang et al., 2016; Yue et al., 2016) focus on the integrated optimization of train stopping patterns and schedules. Niu and Zhou (2013) and Niu et al. (2015) concentrate on passenger flow and timetabling. Michaelis and Schöbel (2009) formulate an integrated model and present a heuristic to address three well-known problems (line planning, timetabling, and vehicle scheduling) in bus transportation. Schöbel (2017) presents a generic, bi-objective model for this problem and design iterative algorithms as heuristics for the integrated problem. However, the methods that they propose have not been applied to railways. Table 1 lists the mathematical formulations and solution algorithms of existing studies.

1.2 Knowledge gaps

Note that line planning, timetabling and rolling stock scheduling are often studied independently in the literature due to the complexity of each problem (see Figure 1). Although a few researchers have attempted to explore integrated line planning, most of them focus on the integration of line planning and timetabling for railways, whereas simultaneously accounting for line planning, timetabling and rolling stock scheduling for a railway system is rarely discussed in the literature. Moreover, the solution methods for integrated optimization are usually optimization software and have not been applied in large-scale, real-world daily transportation systems. In this article, we propose a new methodology to simultaneously account for total passenger waiting times, train timetabling costs and rolling stock usage for URT lines. The framework of our proposed methodology is illustrated in Figure 2. 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 the algorithm, the first step is to optimize train frequency while guaranteeing passenger flow constraints. The model simultaneously schedules train timetables and 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 and rolling stock scheduling.

We intend to address the following specific objectives:

(1) Analyze the time-dependent passenger flow, train timetabling and rolling stock scheduling in URT systems and discuss their interactions. This article develops an optimization methodology that enables the integration of train timetabling, line planning and rolling stock scheduling.
(2) URT lines include both loop and linear lines. To formulate an integrated optimization model, this article 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, minimizing passenger waiting times and operation costs. The lower level model schedules rolling stock to minimize the number of infeasible train paths and proposes a simulated-annealing (SA)-based heuristic algorithm to solve the model.
(4) Illustrate the use of an integrated optimization model and SA-based algorithm to improve the timetable of three typical real-world URT lines in Beijing, China.

The remainder of this article is structured 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, an SA-based algorithm is proposed to solve the model. Subsequently, we generate computational results from real-world instances of the Beijing URT and demonstrate the effectiveness of our proposed model. Finally, the principal conclusions are presented, and possible future research directions are suggested.

2 PROBLEM DESCRIPTION

2.1 URT timetable and key elements

A train timetable defines train departure and arrival times at each station and is an essential plan for the operation of a railway system. We obtained

Table 1
Mathematical formulations and solution algorithms of existing studies

Topic		*Publication*	*Model type*	*Objective*	*Solution method*
Network design		Fan and Machemehl (2006)	NMIP	Minimize total cost	SA
		Kermanshani et al. (2010)	LMIP	Maximize network coverage	SA
Line planning		Goossens et al. (2004, 2006)	ILP	Minimize total operation costs	Branch-and-cut method
		Lin and Ku (2014)	IP	Maximize general profit	Genetic algorithms
		Canca et al. (2017)	NLP	Maximize general profit	Adaptive neighborhood search algorithm
Timetable		Caprara et al. (2002)	ILP	Maximize total profits	Lagrangian relaxation
		Zhou and Zhong (2005, 2007)	MIP	Minimize total train travel time	Lagrangian relaxation, branch and bound
		Carey and Crawford (2007)	–	Identify and resolve conflicts	Heuristics
		Goverde (2007)	Analytical	Timetable robustness	Max-plus system theory
		Wong et al. (2008)	MIP	Minimize interchange waiting times	Optimization-based heuristic
		Barrena et al. (2014)	NLP	Minimize average waiting time	Neighborhood search metaheuristic
		Min et al. (2011)	MIP	Minimize total weighted deviation	Column-generation-based algorithm
		Corman et al. (2010, 2012, 2014)	Alternative graph model	Minimize train delay	Tabu search algorithm, branch and bound algorithm
		Dollevoet et al. (2012, 2014b,b)	IP	Minimize travel time	Heuristics & CPLEX
Rolling stock schedule		Nielsen et al. (2012)	Simulation	Minimize total cost	Rolling horizon algorithm
		Thorlacius et al. (2015)	–	Maximize benefits	Hill climbing heuristic
		Haahr et al. (2016)	MILP	Minimize total cost	GAMS & column generation approach
Integrated model	Line planning and timetable	Kaspi and Raviv (2013)	MIP	Minimize both user inconvenience and operational costs	Cross-entropy metaheuristic
	Timetable and passenger demand	Niu and Zhou (2013)	BIP	Minimize total number of waiting passengers	Genetic algorithm
		Niu et al. (2015)	NLIP	Minimize total waiting time	GAMS/AlphaECP
	Train scheduling and train stop planning	Yang et al. (2016)	MIP	Minimize total dwell time and delay	CPLEX solver
		Yue et al. (2016)	MIP	Maximize total profit	Column generation algorithm
	Line planning, timetabling, and vehicle scheduling (bus transportation)	Michaelis and Schöbel (2009)	IP	Maximize attractiveness of the bus transportation system	Heuristics
		Schöbel (2017)	Bi-objective model	Multiobjective	Iterative heuristics

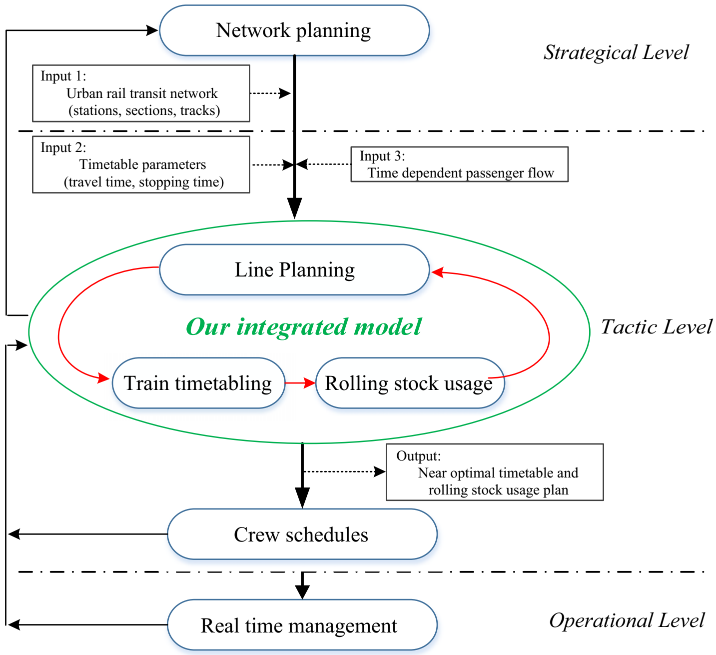


Fig. 2. Framework of integrated URT planning methodology.

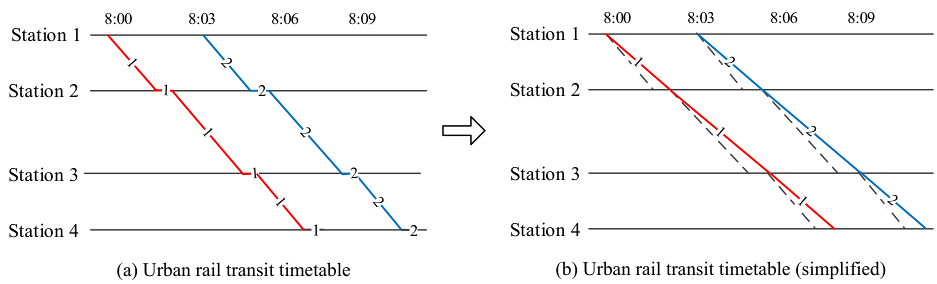


Fig. 3. Illustration of a train timetable along a rail transit line.

passenger data from Beijing Subway Company's operation database, which includes the number of passengers arriving and departing each station in time units of minutes. Figure 3a provides an illustration of a URT timetable. In most URT systems, train types are similar, and trains stop at each station. We focus only on the train frequency in line planning. Trains experience similar dwell times without overtaking operations, which exceed acceptable waiting times. Therefore, arrival and departure times can be defined by the departure time at the origin station or the arrival time at the terminal station, as shown in Figure 3b. In this simplification, the decision variables for train timetables are the departure times at the origin station instead of the departure and arrival times at every station. Thus, the problem is simplified, and the calculation amount is dramatically reduced.

Each train departs from a depot when service begins and returns to a depot when service terminates. Stations that connect directly to depots have greater importance and are named "D-stations" in the model. For example, in Figure 4, Xizhimen and Jishuitan are D-stations for Line 2; Tiantongyuanbei and Songjiazhuang are D-stations for Line 5; and Bagou, Chedaogou and Songjiazhuang are D-stations for Line 10.

The URT lines have three key elements: passenger flow, train flow and rolling stock. In a URT system, the movement of people is referred to as passenger flow. Passengers' arrival times are determined by their individual travel purposes (e.g., home-to-work or home-to-shopping) and are affected by external random factors such as freeway conditions and the walking distances between homes or offices and rail stations. Passenger flows are time dependent.

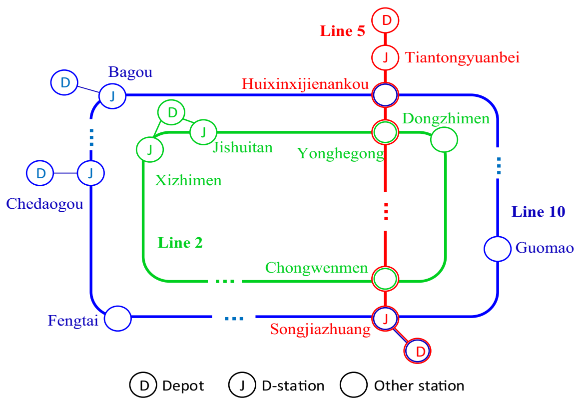


Fig. 4. Illustration of rail transit network in our cases.

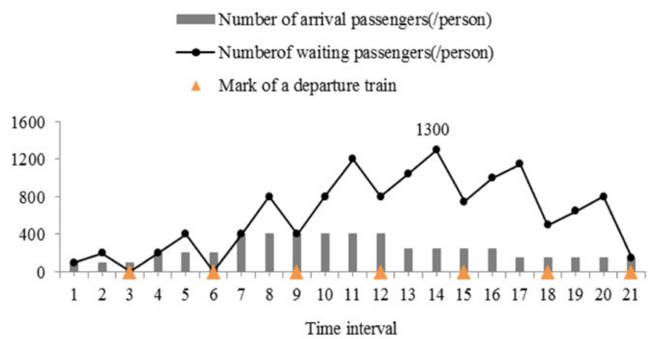


Fig. 6. Relationship between passenger flow and train flow.

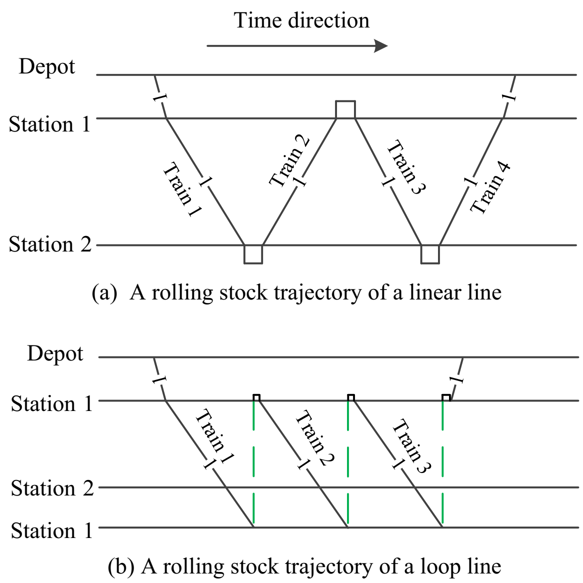


Fig. 5. Illustration of rolling stock trajectory.

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. For linear lines, the trajectory of one rolling stock is illustrated in Figure 5a. After completing a service run, the rolling stock must turn around on a lead track or a station track before it begins the next run. In a loop line, rolling stock can perform multiple service runs without reversing direction, as shown in Figure 5b. Each rolling stock may have several trajectories in a day; each trajectory includes a path from the depot to a D-station, a number of train paths and a path from a D-station to the depot. Each train path corresponds to a train in the timetable. We consider the number of available rolling stocks in each depot and the minimum time span a rolling stock needs for maintenance once it reaches the depot.

2.2 Relationships between key elements

Figure 6 shows the connection between passenger flows and train flows. In Figure 6, we show a time span with 21 time intervals, each of which indicates a possible train departure time. The bar chart indicates the number of arriving passengers, whereas the line chart shows the number of passengers waiting on the station platform. When the peak-hour demand temporarily exceeds the maximum loading capacity of a train, passengers may not be able to board the next train and may be forced to wait in queues for succeeding trains. This condition is referred to as saturation. Conversely, when the number of passengers at a station is less than the maximum loading capacity of a train, the resulting condition is referred to as nonsaturation. For example, in Figure 6, the time period before interval "6" involves a period of nonsaturation, whereas the time period following interval "6" shows a saturated condition. During saturation, urban rail operators must enter as many trains as possible into operation. During nonsaturation, operators can decrease the train frequency to reduce operation costs.

Passengers perceive service coverage and frequency level as the most important factors in service quality. The main objective of operators, however, is to maximize profit. The primary challenge in transit planning is to discover tradeoffs between these conflicting objectives.

We illustrate the relationships between train flow and rolling stock in Figure 7. Each solid line represents a train, and lines of the same color denote a rolling stock trajectory. Assume that three rolling stocks exist in depot 1. The number of trains between station 1 and station 2 is determined by the train intervals (passenger flow). Due to constraints on the number of available rolling stock, some trains (denoted by solid lines) can be placed into service, whereas other trains (as denoted by dashed lines) cannot. Dashed lines must be deleted in the final output train timetable.

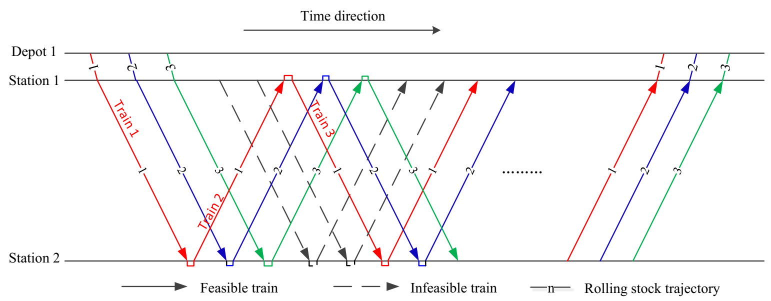


Fig. 7. Relationship between train flow and rolling stock.

Table 2
General subscripts and sets

C	Set of stations: $1, \ldots, c$
D	Set of D-stations: $1, \ldots, d$, $D \in C$
E	Set of time intervals: $1, \ldots, e$
F	Set of trains
G	Set of rolling stock
$i,\ i\prime,\ j,\ j\prime$	Station indices, $i,\ i\prime,\ j,\ j\prime \in C$
$k,\ k\prime$	D-station indices, $k,\ k\prime \in D$
$t,\ s, r$	Time interval indices, $t,\ s, r \in E$
f	Train index, $f \in F$
g	Rolling stock index, $g \in G$

Table 3
Parameters

$o_{i,j}^{t}$	Number of passengers who leave station i for station j at time t
$q_{i,j}^{t}$	Passenger volume between station i and station j at time t
$L^{\max}$	Maximum number of passengers for a train
L^{expt}	Expected number of passengers for a train
F_k^{num}	The number of train paths of D-station k
$d(f)$	Origin station of a train, $d(f) \in D$
h	Minimum departure and arrival headway (time interval) between two consecutive train paths at a station
$d(g)$	Depot connected to the origin station at which rolling stock g starts to perform transportation task
$e_{i,j}$	Running time between station i and station j
$e_{run}^{\max}$	Maximum running time for a rolling stock in a day
$e_{dwell}^{\max}$	Maximum waiting time at a D-station for a rolling stock
$e_{stop}^{\min}$	Minimum stopping time in the depot for a rolling stock
w_i^t	Waiting passengers of station i at time t
ϕ	Value of generalized cost for each waiting passenger
φ	Value of generalized cost for each running of a train
η	Value of generalized cost for each infeasible train path

3 MATHEMATICAL FORMULATION

3.1 Notations

The general subscripts, input parameters, and decision variables that are employed in our mathematical formulations are listed in Tables 2–4.

3.2 General train flow model for urban transit lines

Each rail passenger trip consists of an origin station, a destination station, and a travel time. We obtain passenger data from Beijing Subway Company's operation database, which includes the number of passenger arrivals and departures at station in time units of minutes. We can obtain the approximate passenger volume between two adjacent D-stations based on the number of passengers for each origin–destination matrix. 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.

$$q_{i,i+1}^{t} = \sum_{j\prime \in [i+1,c]} \sum_{i\prime \in [1,i]} o_{i\prime,j\prime}^{t+e_{i\prime,i}} \quad \forall t \in E,\ i \in C \qquad (1)$$

Using Equation (1), we can obtain the passenger volume between two adjacent stations as illustrated in Figure 8a. 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 successive sections.

$$q_{i,j}^{t} = \max\left\{q_{i,i+1}^{t},\ q_{i+1,i+2}^{t+e_{i,i+1}},\ \ldots,\ q_{j-1,j}^{t+e_{i,j-1}}\right\}$$
$$\forall t \in E,\ i,\ j \in C \qquad (2)$$

For example, in Figure 8b, assume that $e_{1,2} = 3$, $e_{2,3} = 2$, $e_{3,4} = 2$, $q_{1,4}^{1}$ indicates the passenger

Table 4
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
$x_{k,k\prime}^{t,s}(g)$	1, if rolling stock g departs from D-station k at time t and arrives at D-station $k\prime$ at time s 0, otherwise
$y_{k,k}^{t,s}(g)$	1, if rolling stock g waits at D-station k from time t to time s 0, otherwise and $s = t + 1$
$z_{k,k}^{t,s}(g)$	1, if rolling stock g stops at depot connected to D-station k from time t to time s 0, otherwise

volume between station 1 and station 4. According to the passenger volume between two adjacent stations in Figure 8a, we can calculate the maximum passenger volume between station 1 and station 4. $q_{1,4}^1 = \max\{q_{1,2}^1, q_{2,3}^4, q_{3,4}^6\}, q_{1,4}^2 = \max\{q_{1,2}^2, q_{2,3}^5, q_{3,4}^7\}$.

As previously mentioned, the line topology structure can be divided into two main categories: linear lines and loop lines. Figure 9a 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 modeling purposes, we modify linear lines to loop lines by treating tracks in different directions as virtually different stations. Thus, we can transform linear lines into a looped line topology with one or more depots. A rolling stock departs from a depot, travels some loops along the stations and returns to the origin depot, as shown in Figure 9b. In this example, type 1 has one depot, type 2 has two depots, and type 3 has three depots. Multiple depot line trajectories must be coordinated to satisfy the minimum train headway constraints at D-stations.

3.3 Bi-level mathematical model

In the traditional serial process of railway planning, timetable scheduling is higher priority than rolling stock assignment, the objective of timetable scheduling usually is satisfying traffic demand, and the objective of rolling stock assignment is to minimize the number of rolling stocks in use or the number of infeasible trains with given rolling stocks. Our integrated model aims to find tradeoffs between these two objectives simultaneously and analyze the interaction between timetabling and the rolling stock schedule. Thus, we use a bi-level programming model to describe the integrated model for URT. The objective of the upper level model is to optimize train timetables, including minimizing waiting times for passengers and reducing operating costs. 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 all feasible, the problem will be solved.

3.3.1 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. The values α, β are weights for the cost of passengers and the operation cost.

$$obj\_up = Min\ (\alpha \times obj\_w + \beta \times obj\_u) \tag{3}$$

$$obj\_w = \sum_{k\in D}\sum_{t\in E} \phi w_k^t \tag{4}$$

$$obj\_u = \sum_{f\in F}\sum_{k\in D}\sum_{t\in E} \varphi u_k^t(f) \tag{5}$$

(1) "Passenger flow" constraints

The number of passengers waiting at time t is the number of waiting passengers at the prior time $t-1$ plus the number of arriving passengers at time t minus the number of departing passengers at time t. At

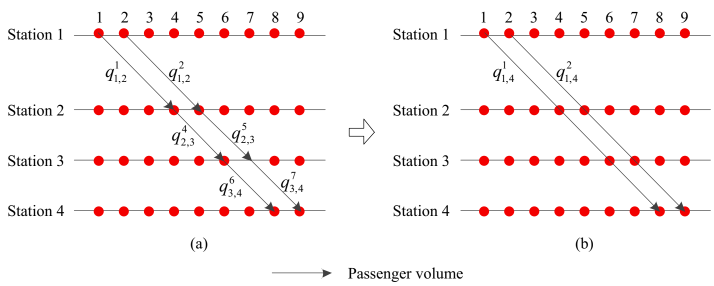
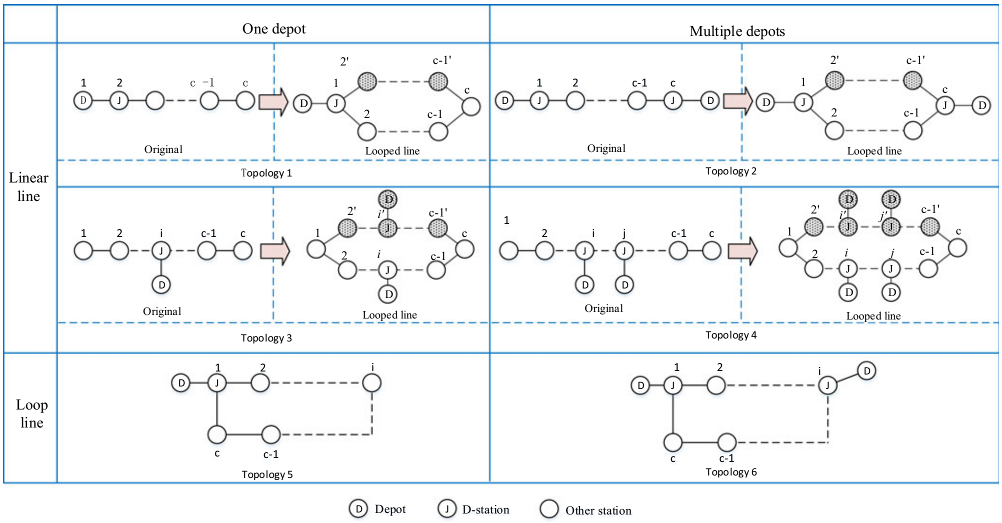
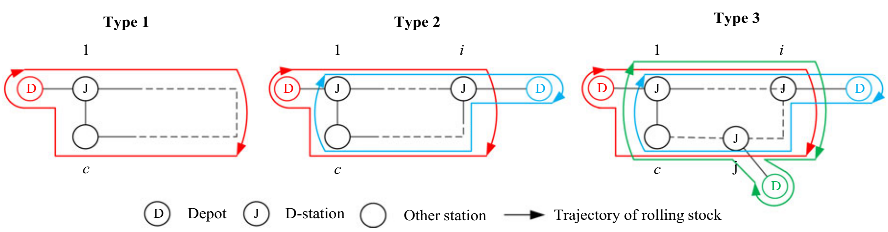


Fig. 8. Illustration of passenger volume in sections. (a) Original and modified topologies of URT lines. (b) Trajectories of rolling stocks.



(a) Original and modified topologies of URT lines



(b) Trajectories of rolling stocks

Fig. 9. Illustration of topological structure of URT lines and rolling stock arrangement.

saturation, the number of passengers on the train is equal to $L^{\max}$, and at nonsaturation, the number of passengers on the train is less than $L^{\max}$. Considering cases of saturation and nonsaturation, we use "greater than" in Equations (6)–(8).

$$w_k^t \geq 0 \quad \forall t \in E,\ k \in D \tag{6}$$

$$w_k^1 \geq q_{k,k'}^1 - \sum_{f\in F} u_k^1(f) \times L^{\max} \quad \forall k,\ k' \in D \tag{7}$$

$$w_k^t \geq w_k^{t-1} + q_{k,k'}^t - \sum_{f\in F} u_k^t(f) \times L^{\max}$$

$$\forall t \in E/\{1\},\ k, k' \in D \tag{8}$$

(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 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.

$$\sum_{f\in F} \sum_{t\in[r,r+h]} u_k^t(f) \leq 1 \quad \forall r \in E, k \in D \tag{9}$$

$$\sum_{f\in F} \sum_{t\in[r,r+h]} v_k^t(f) \leq 1 \quad \forall r \in E, k \in D \tag{10}$$

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

$$u_k^t(f) = v_{k'}^{t+e_{k,k'}}(f) \quad \forall t \in E,\ k,\ k' \in D,\ f \in F \tag{11}$$

$$v_k^t(f) = u_k^t(f) \quad \forall t \in E,\ k \in D/\{d(f)\},\ f \in F \tag{12}$$

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

$$\sum_{k\in D} \sum_{t\in E} v_k^t(f) = \sum_{k\in D} \sum_{t\in E} u_k^t(f) \quad \forall f \in F \tag{13}$$

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

$$obj\_low = Min\ obj\_x \tag{14}$$

$$obj\_x = \sum_{k\in D}\sum_{t\in E}\eta\left[u_k^t(f) - \sum_{k'\in D}\sum_{s\in E}x_{k,k'}^{t,s}(g)\right] \quad \forall u_k^t(f)=1,\ f\in F,\ g\in G \tag{15}$$

(1) "Flow conservation" constraints

For any space–time node (k,t), the number of outflows is not more than one.

$$\sum_{s\in E}\left[\sum_{k'\in D}x_{k,k'}^{t,s}(g) + y_{k,k}^{t,s}(g) + z_{k,k}^{t,s}(g)\right] \le 1 \quad \forall t\in E,\ k\in D,\ g\in G \tag{16}$$

For any space–time node (k,t), the difference between outflows and inflows should be equal to 1 or –1 if the node is a source or a sink node, respectively; otherwise, it should be equal to 0.

$$\begin{aligned}&\sum_{s\in E}\left[\sum_{k'\in D}x_{k,k'}^{t,s}(g) + y_{k,k}^{t,s}(g) + z_{k,k}^{t,s}(g)\right] - \sum_{s\in E}\left[\sum_{k'\in D}x_{k',k}^{s,t}(g) + y_{k,k}^{s,t}(g) + z_{k,k}^{s,t}(g)\right]\\ &= \begin{cases}1 & t=1, k=d(g)\\ -1 & t=e, k=d(g)\\ 0 & \text{otherwise}\end{cases} \quad \forall t\in E,\ k\in D,\ g\in G\end{aligned} \tag{17}$$

(2) "Train flow" constraints

For any space–time node (k,t), the number of train trajectories is no greater than the number of train paths.

$$\sum_{k'\in D}\sum_{s\in E}x_{k,k'}^{t,s}(g) \le u_k^t(f) \quad \forall t\in E,\ k\in D,\ d(f)=d(g) \tag{18}$$

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.

$$\sum_{g\in G}\sum_{k'\in D}\sum_{s\in E}x_{k,k'}^{r,s}(g) + \sum_{g\in G}\sum_{t\in E}y_{k,k}^{r,t}(g) \le 1 \quad \forall r\in E,\ k\in D \tag{19}$$

(3) "Rolling stock" constraints

For any space–time node (k,t), a train cannot run more than $e_{run}^{\max}$ per day.

$$\sum_{k\in D}\sum_{k'\in D}\left[e_{k,k'} \times \sum_{t\in E}\sum_{s\in E}x_{k,k'}^{t,s}(g)\right] \le e_{run}^{\max} \quad \forall g\in G \tag{20}$$

For any space–time node (k,t), a train cannot wait in a D-station for more than $e_{dwell}^{\max}$ each time.

$$\sum_{t\in E}\sum_{s\in[r,r+e_{dwell}^{\max}]}y_{k,k}^{t,s}(g) \le e_{dwell}^{\max} \quad \forall r\in E,\ k\in D,\ g\in G \tag{21}$$

For any space–time node (k,t), a train cannot stop in a depot for less than $e_{stop}^{\min}$ each time.

$$\sum_{t\in E}\sum_{s\in[t,t+e_{stop}^{\min})}z_{k,k}^{t,s}(g) = 0 \quad \forall k\in D,\ g\in G \tag{22}$$

4 SIMULATED ANNEALING ALGORITHM

4.1 Algorithm feature and selection

The problem we have proposed is a large-scale combinatorial optimization problem including a large number of constraints. The decision variables of the upper level model are $u_i^t(f)$, $v_i^t(f)$, and w_i^t, and the number of decision variables is $2\times e\times d\times f + e\times d$. In the lower level model, the decision variables are $x_{k,k'}^{t,s}(g)$, $y_{k,k}^{t,s}(g)$, and $z_{k,k}^{t,s}(g)$, and the number of decision variables is $3\times g\times e\times e\times d\times d$. Assume that the operation time of one subway line is 20 hours, a train runs every 5 minutes, and there are 2 depots and 50 rolling stocks. Then, one day can be divided into 1,200 time intervals (of 1 minute each). The number of decision variables for the upper level model is 1,154,400, and the number of decision variables for the lower level model is approximately 8.64×10^8. The sheer size of the timetable optimization problem for a typical URT line necessitates an algorithm that can solve the problem effectively in a time-efficient manner.

To solve this problem, two types of algorithms can be considered: (1) exact algorithms, such as the Lagrangian relaxation algorithm and the column generation algorithm and (2) heuristic algorithms, such as the genetic algorithm and the ant colony algorithm. Note the following important considerations in our proposed models: the decision variables of train paths are based on the space–time network; train scheduling and rolling stock assignment will interact with each other; a train path will affect passenger flow and other train paths; and the number of variables increases rapidly. The application of divide-and-conquer algorithms (such as the Lagrangian relaxation algorithm and column generation algorithm) is difficult because they require decomposing the primal problem into subproblems, and expressing the relationship of the subproblems. The ant colony algorithm or genetic algorithm can hardly address the complex constraints of this model.

SA has been adopted widely to solve engineering problems, for example, the train platform problem (Kang et al., 2012), transit network optimization problem (Zhao and Zeng, 2006), and bottleneck routing problem at railway stations (Wu et al., 2013), among others. It is a local search-based algorithm that has a mechanism to escape from local optima with the purpose of finding a global optimum. The essential feature of the SA algorithm is allowing a hill-climbing movement by introducing an acceptance criterion. The calculation process of SA is simple and robust. Moreover, it is suitable for parallel processing and can be used to solve complex nonlinear optimization problems. Considering these factors, we utilize the SA metaheuristic to solve our proposed models and to derive a near-optimal solution. We also tried GAMS with the CPLEX solver to validate the model and compare the calculation results to the results of our algorithm. The evaluation function for solution A is $obj(\mathrm{A}) = \alpha \times obj\_w + \beta \times obj\_u + \gamma \times obj\_x$, where γ is the penalty for the number of infeasible train paths.

4.2 Procedures for the simulated annealing algorithm

The main process of the algorithm is using the train timetable from the upper level as the input for the lower level model. Some train paths will be deleted after optimizing the rolling stock schedule. We will iterate this procedure until a good solution is obtained. In this section, we provide the detailed procedures of the algorithm. The parameters of the algorithm are shown in Table A1.

Step 1. Initialize parameters and variables.

Input data on line and passenger volume, initialize parameters of the algorithm, and initialize variables $u_i^t(f)$, $v_i^t(f)$ and w_i^t, obj_x, $obj(A)$.

Step 2. Obtain an initial solution F of the upper level model, and compute obj_w, obj_u.

We use a blank train timetable ($F = \Phi$) as the initial feasible solution.

Step 3. Impose a disturbance and generate a new solution F' of the upper level model.

To generate a new timetable solution, we choose the existing train path randomly and use a greedy method to enumerate possible neighboring feasible train path and add to the existing train list.

Step 4. Neighborhood search method for rolling stock assignment.

Step 4.1. Choose first train path without rolling stock assigned, $f \in F'$.

Step 4.2. Check the availability of rolling stock using the FIFO (First In, First Out) principle. Available rolling stocks are the rolling stocks which are located at a specified depot and also satisfy constraints 20–22.

If the rolling stock is available, go to Step 4.5: assign the rolling stock to a train path. If not, go to Step 4.3.

Step 4.3. Find next available rolling stock for train path f till the last rolling stock in the depot.

Step 4.4. Was that entry the end of the rolling stock stack?

If not, go to Step 4.5. Otherwise, all rolling stocks are unavailable. This train path is infeasible and will be deleted from the train path list; $obj\_x' += 1$, go to Step 4.1.

Step 4.5. Assign the rolling stock to train path f and go to Step 4.1.

Step 4.6. If $f > F_k^{num}$, obtain a solution A' and compute the evaluation function: $obj(A)$, $obj(A')$.

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 $\Delta obj < 0$, the new solution must be accepted; otherwise, the new solution is accepted with a probability of p^{accept}.

$$p^{accept} = \begin{cases} 1 & \Delta obj < 0 \\ \exp\left(\frac{-\Delta obj}{M}\right) & \Delta obj \geq 0 \end{cases}$$

Step 6. Stop condition.

Two conditions may terminate the algorithm: (1) the iteration number reaches the maximum limit and (2) the stable iteration number reaches the maximum limit, and the number of infeasible train paths is "0." If one condition 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. The temperature decreases in two stages. The temperature drop curve of each stage is an exponential function. Upon reaching the specified number

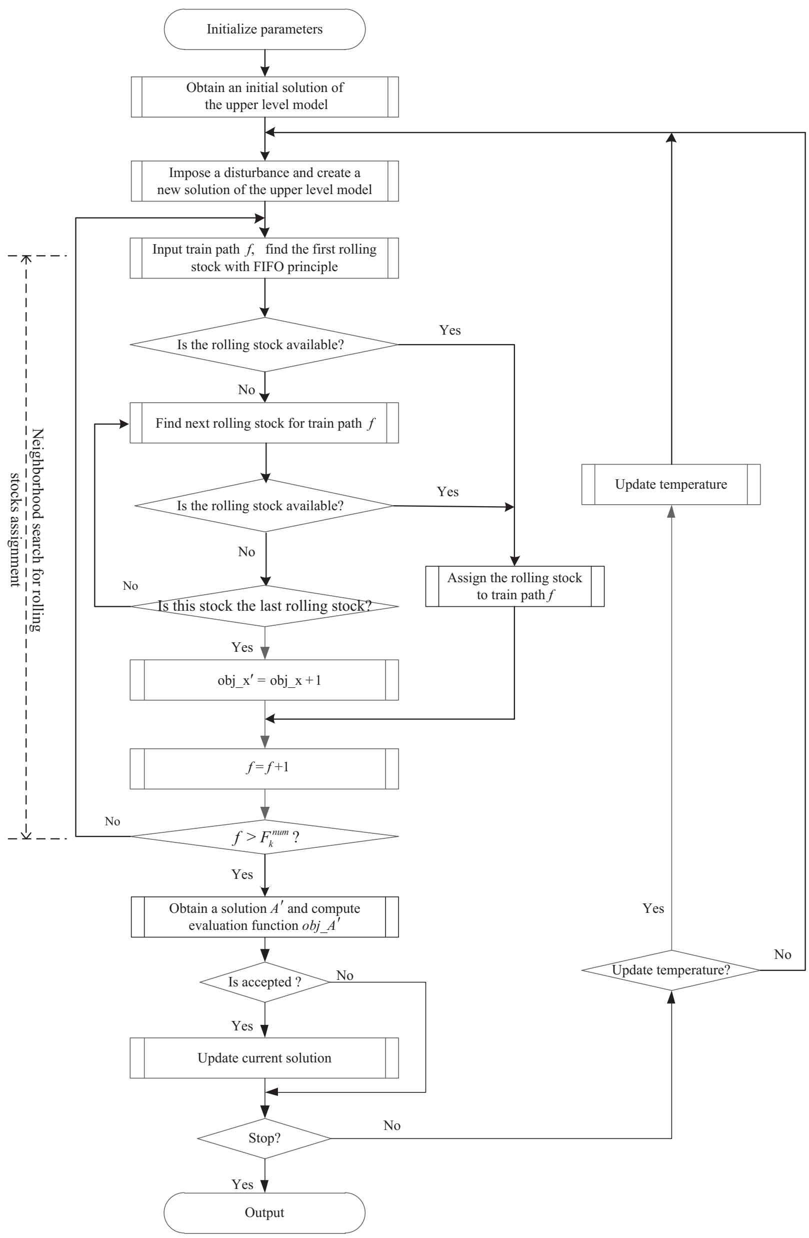


Fig. 10. Solution methodology procedures.

of iterations, the temperature is increased by a process of "warming up," and then the temperature decreases exponentially.

$$M = \begin{cases} M_0 & n = 0 \\ \rho^{rise} \times M_0 & n = N^{rise} \\ \rho \times M' & \text{otherwise} \end{cases}$$

where M is the current temperature; M_0 is the initial temperature; M' is the temperature in the previous state; N^{rise} is the number of iterations to increase temperature. The value $\rho\in(0,1)$ is a constant close to 1, and $\rho^{rise}\in(0,1)$.

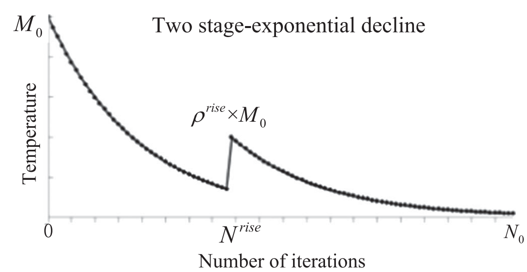


Table 5
Model parameter definitions

Parameters	*Value*	*Meaning*
h	2 minutes and 15 seconds	Minimum departure and arrival headway (time interval) between two consecutive train paths at a station
α	1	Weight of passenger waiting time
β	9,600	Weight of train frequency
γ	1,500,000	Weight of infeasible trains
$e_{run}^{\max}$	2,880	Maximum running time
$e_{dwell}^{\max}$	12	Maximum dwelling time in D-stations
$e_{stop}^{\min}$	360	Minimum stopping time in depots

Table 6
Date of Beijing rail transit Line 10

Line	*Number of depots*	*Number of rolling stocks*	*Number of stations*	*Running time (minutes/lap)*	*Line length (km/lap)*
10	3	10, 24, 24	45	107	57.1

Step 8. Output the result.

The following iterative procedure was implemented to solve the problems, as summarized in Figure 10.

5 CASE STUDY

We test the proposed algorithm on three real-world lines (Line 2, Line 5, and Line 10) of Beijing's railway transit network in two situations (weekdays and weekends). Due to the limitations of this article, we only list the case study of Subway Line 10 on a workday 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 5 shows the parameter definitions: d is the number of depots, and b represents the number of time intervals of 1 minute. (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 article is 20 hours (1,200 minutes), 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 minutes). Table 6 lists the data of Beijing rail transit Line 10.

5.1 Results and analysis of Line 10

Line 10 of Beijing's railway transit network is a ring route. Three depots (refer to blue line in Figure 4) connect with the Bagou station, Chedaogou station and Songjiazhuang station, respectively. These three stations are D-stations in our case. The minimum departure headway between two consecutive train paths is 2 minutes and 15 seconds. The time interval is 15 seconds. In this case, $b = 4$, $d = 3$. We consider the counter-clockwise direction of this ring line. The passenger demand of Beijing rail transit Line 10 on workdays is shown in Table A2. These data are real data obtained for the Beijing Subway.

After 10 rounds of calculations, we obtain the results (refer to Table 7) for Beijing rail transit Line 10 for workdays. The calculation time for these cases is approximately 18 minutes. On workdays, the mean value

Table 7
Results for Beijing rail transit Line 10 (workdays)

Calculation times	*obj_A*	*obj_w*	*obj_u*	*obj_x*	*Number of iterations*	*Computing time*
1	15,811,540	7,286,740	888	0	204,487	0:19:05
2	15,958,062	7,346,862	897	0	282,806	0:15:01
3	16,102,443	7,548,843	891	0	247,470	0:20:27
4	15,866,348	7,399,148	882	0	272,230	0:18:26
5	16,058,657	7,562,657	885	0	213,494	0:20:19
6	15,798,396	7,215,996	894	0	251,425	0:17:04
7	15,886,218	7,303,818	894	0	239,748	0:15:33
8	15,774,205	7,163,005	897	0	228,145	0:19:45
9	15,971,274	7,388,874	894	0	260,432	0:16:12
10	16,028,443	7,359,643	903	0	247,070	0:16:17
Mean	15,925,559	7,357,559	893	0	244,731	0:17:49
STDV	115,406	127,923	6.2	–	–	–

Table 8
Rolling stock schedule of depot for workdays

No.	*Running time (minutes)*	*Rolling stock scheduling*
1	642	[26 454][975 1,403][1,408 1,836][1,843 2,271][2,963 3,391][3,791 4,219]
2	642	[161 589][597 1,025][1,026 1,454][1,869 2,297][2,297 2,725][3,148 3,576]
3	642	[177 605][608 1,036][1,593 2,021][2,024 2,452][2,460 2,888][3,271 3,699]
4	642	[203 631][631 1,059][1,718 2,146][2,146 2,574][3,000 3,428][3,854 4,282]
5	535	[229 657][1,057 1,485][1,939 2,367][2,368 2,796][3,243 3,671]
6	535	[253 681][685 1,113][1,750 2,178][2,573 3,001][3,426 3,854]
7	642	[283 711][711 1,139][1,967 2,395][2,400 2,828][2,829 3,257][3,665 4,093]
8	642	[299 727][732 1,160][1,167 1,595][2,056 2,484][2,485 2,913][3,292 3,720]
9	535	[445 873][873 1,301][2,075 2,503][2,514 2,942][3,357 3,785]
10	642	[470 898][899 1,327][2,175 2,603][2,606 3,034][3,035 3,463][3,868 4,296]
11	642	[497 925][925 1,353][1,354 1,782][1,789 2,217][2,220 2,648][3,023 3,451]
12	535	[554 982][989 1,417][1,420 1,848][2,390 2,818][3,442 3,870]
13	642	[585 1,013][1,014 1,442][1,446 1,874][2,446 2,874][2,877 3,305][3,679 4,107]
14	642	[656 1,084][1,085 1,513][1,514 1,942][2,500 2,928][2,929 3,357][3,722 4,150]
15	535	[720 1,148][1,153 1,581][2,596 3,024][3,395 3,823][3,823 4,251]
16	642	[1,112 1,540][1,543 1,971][1,981 2,409][2,413 2,841][2,841 3,269][3,882 4,310]
17	428	[1,138 1,566][1,573 2,001][2,721 3,149][3,522 3,950]
18	535	[1,259 1,688][1,688 2,116][2,122 2,550][2,911 3,339][3,948 4,376]
19	535	[1,298 1,726][2,817 3,245][3,256 3,684][3,694 4,122][4,122 4,550]
20	535	[1,326 1,754][2,865 3,293][3,304 3,732][3,736 4,164][4,164 4,592]
21	535	[1,395 1,823][1,830 2,258][2,265 2,693][3,102 3,530][3,979 4,407]
22	321	[2,889 3,317][3,317 3,745][3,750 4,178]
23	321	[2,900 3,328][3,333 3,761][3,763 4,191]
24	321	[2,938 3,366][3,366 3,794][4,287 4,715]

Fig. 11. Iterative solution process of the evaluation function.

of obj_A is 15925559, the average number of waiting passengers is 567 persons/time interval (obj_w/(b $\times$ d $\times$ 1,080)), and the average train frequency is 297 trains/day (obj_u/d).

Table 8 shows the rolling stock schedule of the depot that is connected to Chedaogou station for workdays. The first column represents the rolling stock number. The second column shows the total number of running time intervals for each rolling stock. In this case, the time interval is 15 seconds. The third column represents the timetable of every rolling stock. For example, [26 454] indicates that the first rolling stock begins its first service at the 26th time interval and ends at the 454th time interval (service time interval is 107 minutes). Similarly, its last service starts at the 3,791st time interval and ends at the 4,219th time interval. The total running time for the first rolling stock is 642 minutes. This depot has a total of 24 rolling stocks, and all are used in the schedule. As previously mentioned, the time span of the rolling stock operation is 1,200 minutes. The results reveal that the proposed model and the algorithm apply to the rail transit ring line of multiple depots and can achieve an approximate optimal solution. The train timetable for Beijing rail transit Line 10 during workdays is shown in Figure A1.

5.2 Efficiency analysis of the algorithm

Figure 11 illustrates the iterative solution process of the evaluation function for Beijing rail transit Line 10 on a workday. The horizontal axis represents the number of iterations, and the vertical axis represents the evaluation function. The value of the evaluation function decreased with further iterations and stabilized before the maximum number of iterations had been completed.

Table 9
Comparison of different solving methods

Case	*Number of depot*	*Solver*	*obj_A*	*Comparison of object value*	*Computing time (seconds)*	*Comparison of computational time*
Line 2, workday	1	Upper level model/CPLEX	1,268,350	1	8	1
		Upper level model/SA	1,286,750	1.01	12	1.5
		Bi-level model/ SA+neighborhood searching	1,292,500	1.02	15	1.875
Line 5, workday	2	Upper level model/CPLEX	3,810,410	1	4,239	1
		Upper level model/SA	3,788,936	0.99	95	0.02
		Bi-level model/ SA+neighborhood searching	3,808,960	1	126	0.03
Line 10, workday	3	Upper level model/CPLEX	18,624,221	1	9,369	1
		Upper level model/SA	15,542,351	0.83	682	0.07
		Bi-level model/ SA+neighborhood searching	15,958,062	0.86	1,004	0.11

We also apply a similar optimization approach to Line 2 and Line 5 in Beijing. To evaluate the efficiency of the proposed algorithm, we try to use GAMS with the CPLEX solver to solve the model and compare the results of different methods. Due to the scale of the lower level model, the CPLEX solver cannot solve the lower level model. We only list the calculation results of the upper model solved by CPLEX and SA (γ=0) and the bi-level model solved by SA. We employ the calculation results of the CPLEX solver as a benchmark and perform a comparative analysis. Table 9 compares 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.

The following observations are based on the results presented in Table 9:

(1) For Line 2 on workdays, the results of the upper level model using CPLEX and SA are similar, but the computing time for the CPLEX solver is faster than the computing time for SA. SA and neighborhood searching can obtain the solution for the bi-level model within 15 seconds.
(2) For Line 5 with a time interval of 30 seconds and two depots, the results of the upper level model using CPLEX and SA are similar, but the computing time for the CPLEX solver exceeds 1 hour, and the computing time for SA is less than 2 minutes. SA and neighborhood searching can obtain a better solution of the bi-level model within 3 minutes.
(3) For Line 10 with a time interval of 15 seconds and 3 depots, the CPLEX solver can barely obtain a solution of the upper level model in more than 2 hours, whereas SA and neighborhood searching can obtain the solution within 16 minutes and 44 seconds. The objective function value is improved by more than 14%.

From these observations, it appears that SA, as a meta-heuristic approach, is fast but cannot guarantee global optimality. For a practical, large-scale problem, SA may be a promising approach to yield adequate solutions (but not necessarily perfect) given a reasonable time span.

6 CONCLUSIONS

This article 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 improving the utilization of train sets.

First, we introduce three key elements in URT lines and propose a general train flow model that 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. We test our optimization mathematical model and algorithm using a case study of the Beijing rail transit network. The calculation results indicate that the proposed model and algorithm can obtain reasonable schedule planning. In particular, our newly integrated algorithm can rapidly obtain a near-optimal solution and best utilize rolling stock, which renders it useful for complex real-world applications. The application of this method in the real world can help subway operators make decisions quickly and precisely. In addition, although the methodology and optimization model are presented in the context of a rail transit train timetable and rolling stock schedule in this article, they can also be applied to 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's performance. Second, the model can be extended to account for train skip-stop patterns and long/short trains. Finally, we will refine and validate the proposed model with observed data from URT systems in Beijing.

ACKNOWLEDGMENTS

This work was financially supported by the Fundamental Research Funds for the Central Universities (No. 2017JBM317, No. 2014JBZ008) provided by the Ministry of Education of the People's Republic of China.

REFERENCES

Albrecht, T. (2009), Automated timetable design for demand-oriented service on suburban railways, *Public Transport*, **1**(1), 5–20.

Barrena, E., Canca, D., Coelho, L. C. & Laporte, G. (2014), Single-line rail rapid transit timetabling under dynamic passenger demand, *Transportation Research Part B*, **70**(C), 134–50.

Canca, D., De-Los-Santos, A., Laporte, G. & Mesa, J. A. (2017), An adaptive neighborhood search metaheuristic for the integrated railway rapid transit network design and line planning problem, *Computers & Operations Research*, **78**, 1–14.

Caprara, A., Fischetti, M. & Toth, P. (2002), Modeling and solving the train timetabling problem, *Operations Research*, **50**(5), 851–61.

Carey, M. & Crawford, I. (2007), Scheduling trains on a network of busy complex stations, *Transportation Research Part B*, **41**(2), 159–78.

Castillo, E., Gallego, I. Sánchez-Cambronero, S., Menéndez, J. M., Rivas, A., Nogal, M. & Grande, Z. (2015), An alternate double-single track proposal for high speed peripheral railway lines, *Computer-Aided Civil and Infrastructure Engineering*, **30**, 181–201.

Castillo, E., Grande, Z., Moraga, P. & Sánchez-Vizcaíno, J. (2016), A time partitioning technique for railway line design and timetable optimization, *Computer-Aided Civil and Infrastructure Engineering*, **31**(8), 599–616.

Corman, F., D'Ariano, A. & Hansen, I. A. (2014), Evaluating disturbance robustness of railway schedules, *Journal of Intelligent Transportation Systems*, **18**(1), 106–20.

Corman, F., D'Ariano, A., Pacciarelli, D. & Pranzo, M. (2010), A tabu search algorithm for rerouting trains during rail operations, *Transportation Research Part B*, **44**, 175–92.

Corman, F., D'Ariano, A., Pacciarelli, D. & Pranzo, M. (2012), Bi-objective conflict detection and resolution in railway traffic management, *Transportation Research Part C*, **20**(1), 79–94.

Dollevoet, T., Corman, F., D'Ariano, A. & Huisman, D. (2014a), An iterative optimization framework for delay management and train scheduling, *Flexible Services and Manufacturing Journal*, **26**(4), 490–515.

Dollevoet, T., Huisman, D., Kroon, L., Schmidt, M. & Schöbel, A. (2014b), Delay management including capacities of stations, *Transportation Science*, **49**(2), 185–203.

Dollevoet, T., Huisman, D., Schmidt, M. & Schöbel, A. (2012), Delay management with rerouting of passengers, *Transportation Science*, **46**(1), 74–89.

Fan, W. & Machemehl, R. B. (2006), Using a simulated annealing algorithm to solve the transit route network design problem, *Journal of Transportation Engineering*, **132**(2), 122–32.

Goossens, J. W., van Hoesel, S. & Kroon, L. (2004), A branch-and-cut approach for solving railway line-planning problems, *Transportation Science*, **38**(3), 379–93.

Goossens, J. W., van Hoesel, S. & Kroon, L. (2006), On solving multi-type railway line planning problems, *European Journal of Operational Research*, **168**(2), 403–24.

Goverde, R. M. P. (2007), Railway timetable stability analysis using max-plus system theory, *Transportation Research Part B*, **41**(2), 179–201.

Haahr, J. T., Wagenaar, J. C., Veelenturf, L. P. & Kroon, L. G. (2016), A comparison of two exact methods for passenger railway rolling stock (re) scheduling, *Transportation Research Part E*, **91**, 15–32.

Kang, L., Wu, J. & Sun, H. (2012), Using simulated annealing in a bottleneck optimization model at railway stations, *Journal of Transportation Engineering*, **138**(11), 1396–402.

Kaspi, M. & Raviv, T. (2013), Service-oriented line planning and timetabling for passenger trains, *Transportation Science*, **47**(3), 295–311.

Kecman, P. & Goverde, R. M. P. (2012), Process mining of train describer event data and automatic conflict identification, *Computers in Railways XIII, WIT Transactions on the Built Environment*, **127**, 227–38, https://doi.org/10.2495/CR120201.

Kermanshani, S., Shafahi, Y., Mollanejad, M. & Zangui, M. (2010), Rapid transit network design using simulated annealing, *12th World Conference on Transport Research, WCTR*, (p. 15), Lissabon.

Lin, D. Y. & Ku, Y. H. (2014), Using genetic algorithms to optimize stopping patterns for passenger rail transportation, *Computer-Aided Civil and Infrastructure Engineering*, **29**(4), 264–78.

Michaelis, M. & Schöbel, A. (2009), Integrating line planning, timetabling, and vehicle scheduling: a customer-oriented approach, *Public Transport*, **1**(3), 211–32.

Min, Y. H., Park, M. J., Hong, S. P. & Hong, S. H. (2011), An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network, *Transportation Research Part B*, **45**(2), 409–29.

Nielsen, L. K., Kroon, L. G. & Maróti, G. (2012), A rolling horizon approach for disruption management of railway rolling stock, *European Journal of Operational Research*, **220**(2), 496–509.

Niu, H. & Zhou, X., (2013), Optimizing urban rail timetable under time-dependent demand and oversaturated conditions, *Transportation Research Part C*, **36**(11), 212–30.

Niu, H., Zhou, X. & Gao, R. (2015), Train scheduling for minimizing passenger waiting time with time-dependent demand and skip-stop patterns: nonlinear integer programming models with linear constraints, *Transportation Research Part B*, **76**, 117–35.

Schöbel, A. (2017), An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation, *Transportation Research Part C*, **74**, 348–65.

Thorlacius, P., Larsen, J. & Laumanns, M. (2015), An integrated rolling stock planning model for the Copenhagen suburban passenger railway, *Journal of Rail Transport Planning & Management*, **5**(4), 240–62.

Wong, C. W., Yuen, T. W. Y., Fung, K. W. & Leung, M. Y. (2008), Optimizing timetable synchronization for rail mass transit, *Transportation Science*, **42**(1), 57–69.

Wu, J., Kang, L., Sun, H. & Jia, X. (2013), Track allocation optimization in railway station: mean-variance model and case study, *Journal of Transportation Engineering*, **139**(5), 540–47.

Yang, L., Qi, J., Li, S. & Gao, Y. (2016), Collaborative optimization for train scheduling and train stop planning on high-speed railways, *Omega*, **64**, 57–76.

Yue, Y., Wang, S., Zhou, L., Tong, L. & Saat, M. R. (2016), Optimizing train stopping patterns and schedules for high-speed passenger rail corridors, *Transportation Research Part C*, **63**, 126–46.

Zhao, F. & Zeng, X. (2006), Simulated annealing–genetic algorithm for transit network optimization, *Journal of Computing in Civil Engineering*, **20**(1), 57–68.

Zhou, X. & Zhong, M. (2005), Train scheduling for high-speed passenger railroad planning applications, *European Journal of Operational Research*, **167**(3), 752–71.

Zhou, X. & Zhong, M. (2007), Single-track train timetabling with guaranteed optimality: branch and bound algorithms with enhanced lower bounds, *Transportation Research Part B*, **41**(3), 320–41.

APPENDIX A

Table A1
SA algorithm parameter definitions

Parameter	*Value*	*Meaning*
Number of time intervals	4,800	Number of time intervals
M_0	6,000,000	Initial temperature
ρ^{rise}	0.3	Heating coefficient
ρ	0.95	Cooling coefficient
N_0	720,000	Maximum number of iterations
N^{stop}	3,600	Iteration number of terminating algorithm for no longer improvement
$N_{1\to 2}$	72	Iteration number in initial stage
$N_{2\to 3}$	1,440	Iteration number in medium term
N^{update}	3,600	Iteration number for temperature to update
N^{rise}	180,000	Iteration number for temperature to increase
p_1^{change}	1/600	Update probability in initial stage
p_2^{change}	1/2,400	Update probability in medium term
p_3^{change}	1/12,000	Update probability in later stage

Table A2
Passenger demand of Line 10 on workdays

Scenarios	*Section*	*5:00–7:00*	*7:00–9:00*	*9:00–11:00*	*11:00–17:00*	*17:00–19:00*	*19:00–22:00*	*22:00–23:00*
Workday (persons/min)	Bagou→Chedaogou	50	400	200	200	400	200	50
	Chedaogou→Songjiazhuang	50	500	450	400	500	400	50
	Songjiazhuang→Bagou	50	500	450	400	500	400	50

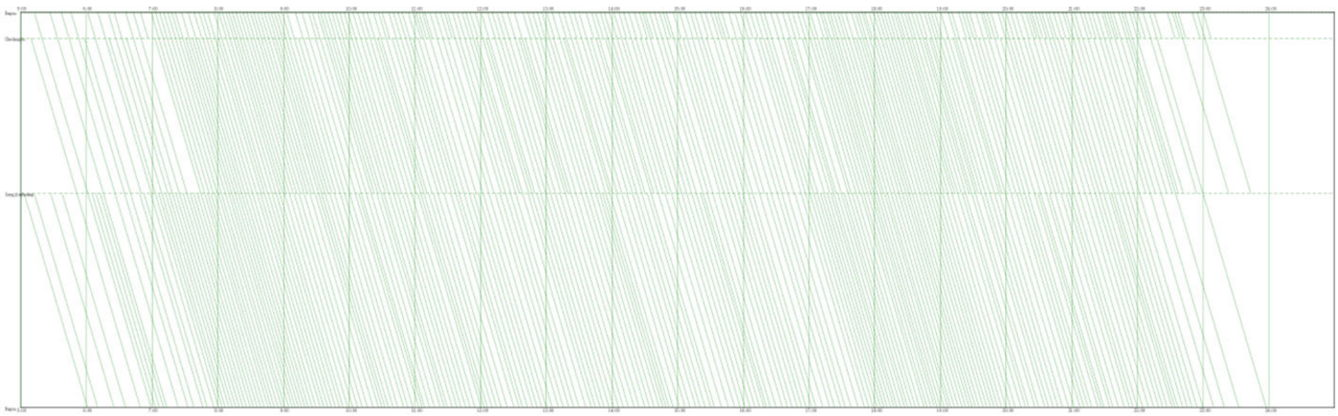


Fig. A1. Train timetable of Beijing rail transit Line 10 on workdays.