An Effective Simulation Model for Multi-line Metro Systems Based on Origin-destination Data

Jiaxuan You, Wei Guo, Yi Zhang, Jianming Hu

Abstract—**Metro systems are playing an increasing role in modern cities, and various management solutions are proposed to improve their efficiency. When analyzing the effects of those solutions, simulation proves to be an effective and cost-saving method. This paper presents an effective simulation model for multi-line metro systems based on the OD (origin-destination) data and the network connection data. The model is validated in the scenario of Beijing metro system, which proves its effectiveness in large-scale empirical implementation. We further apply our model in analyzing the policy of staggered shifts, and prove the convenience of its application.**

I. Introduction

Due to the rapid development of urbanization, urban residents suffer from more and more serious traffic congestion. Thus, metro systems are playing an increasing role in urban transportation, owing to its congestion free properties. To improve the operating efficiency and service level of metro systems, various management solutions and policies were proposed and examined. However, the high cost of field test and the possible risk of system failure may hinder us from carrying out a real-time experiment. Therefore, implementing simulation method would be more cost-saving and effective [1-2].

In fact, metro scenario based simulation methods were applied in a wide range. However, many studies are faced with the following three problems.

The first problem is that few studies propose an explicit framework for metro systems. In [3], a multi-agent system was proposed, where environment, passenger agent and train agent were defined. However, the work did not explicitly reveal the structure behind each agent and the interaction between them. Heimburger et al. proposed a framework incorporating important train and passenger variables, yet was limited to single-line metro systems [4].

The second problem is that they try to precisely reproduce the operation status of metro systems, yet the involved parameters are hard to obtain. Sun et al. [5] established a metro simulation model, with exact train schedule parameters involved. In [2], the metro system simulator set up many train parameters including speed, weight and etc. A comprehensive model in [6] also included train occupancy data which had to be acquired by special circuit. The commonly faced problem is that those train parameters are hard to acquire in reality, which makes validation with empirical data challenging.

The third problem is that many studies do not consider the decision making of passengers. The simulator in [2] did not differentiate between passengers, and assumed that all passengers are distributed according to the probability in the OD matrix. However, each passenger may choose a different route according to his destination and the generalized travel cost for each possible path. Therefore, omitting the decision making of passengers deviates from reality.

To sum up, current metro simulation models lack an explicit framework and incorporate hard-to-obtain parameters, while many of them do not take the decision making of passengers into account. When facing large-scale empirical implementation, current metro simulation models are restricted by the above drawbacks. Thus, many studies only examine their models in a single metro line [1] and lack overall examination on its results [3].

To overcome the above-mentioned problems, this paper presents a hierarchical, empirical and adaptive simulation model for multi-line metro systems. The proposed model has the following three characteristics.

First, it has three hierarchical layers that are explicitly described. In addition, we use the graph structure to store most information, which facilitates the further implementation of algorithms.

Second, the model utilizes easily available data including the OD data and the network connection data. We successfully apply our model in the scenario of Beijing metro system and prove the effectiveness of the model.

Third, our model is adaptive in formulating the travel preferences of passengers. The model can describe various travel preferences of passenger, by simply altering the parameters in the formulation of generalized travel cost and choosing an appropriate path choice algorithm. Those travel preferences can be mined from the smart card data if needed [7].

The remaining of the paper is outlined as follows. In *Section 2*, the theoretical framework for our model is proposed. *Section 3* builds the model with empirical data and verifies the effectiveness of the model. *Section 4* proposes the potential application of the model. Conclusions are made and discussed in *Section 5* finally.

J. You is with Department of Automation, Tsinghua University, Beijing, China 100084. (e-mail: youjx13@mails.tsinghua.edu.cn).

W. Guo is with Department of Automation, Tsinghua University, Beijing, China 100084. (e-mail: guo-w11@mails.tsinghua.edu.cn).

Y. Zhang is with Department of Automation, TNList, Tsinghua University, Beijing, China 100084. (corresponding author, phone: +86(10)62796832, e-mail: zhyi@mail.tsinghua.edu.cn).

J. Hu is with Department of Automation, TNList, Tsinghua University, Beijing, China 100084. (e-mail: hujm@mail.tsinghua.edu.cn).

II. Theoretical Framework

In this section, we propose a hierarchical simulation model which includes three layers, i.e. infrastructure layer, passenger layer and operation layer. Each layer is built on the output of the previous layer. The input data can be categorized into the network connection data and the OD data. The network connection data is imported into the infrastructure layer, and is then saved using the graph structure. The passenger layer takes attributes of stations and paths provided by the infrastructure layer, together with the OD data, to formulate the decision making process of passengers. Built on the basis of previous layers, the operation layer treats passengers as entities with planned path, and simulates the real transportation process. The overall structure of the model is shown in Fig. 1.

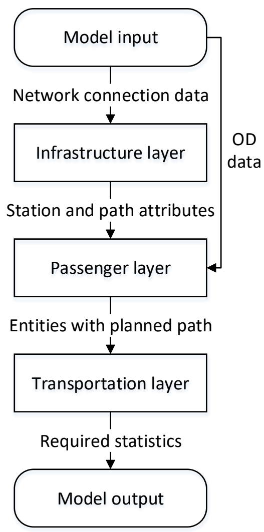

Figure 1. Overall structure of the metro system simulation model

For presentation simplicity, the symbols and notations are enumerated in Table I below.

TABLE I. Nomenclature List of the Metro Simulation Model

Parameters	Interpretations
The symbols below are variables of the infrastructure layer	
ID_i	ID for station i
$Line_i$	The line that station i belongs to
t_i^e	Entrance time for station i
t_i^s	Train stop time for station i
t_i^x	Exit time for station i
$\overline{t_j^t}$	Expected travel time for path j
t_j^t	Real travel time for path j
t_j^b	Transfer boarding time for transfer path j
c_j	Capacity for each train in path j
h_j	Headway for each train in path j
M_j	Maximum capability for transportation in path j
N_j	Current passenger number in path j
The symbols below are variables of the passenger layer	
V_{ij}^k	Generalized travel cost for the k th travel path from station i to station j
T_{ij}^k	Total travel time cost for the k th travel path from station i to station j
C_{ij}^k	Total comfort cost for the k th travel path from station i to station j
E_{ij}^k	Total travel expense cost for the k th travel path from station i to station j
P_{ij}^k	The set of paths that constitute the k th travel path from station i to station j
λ	Transform coefficient of total transfer time
σ	Transform coefficient of the number of times for transferring
U_{ij}^k	Total travel utility for the k th path station i to station j
p_{ij}^k	The probability for a passenger to select the k th path from station i to station j
The symbols below are variables of the operation layer	
TTL	Time to live for the entity in current path
N	Number of passengers in the entity
$T_{theoretical}$	Theoretical total travel time for the entity
T_{real}	Real travel time for the entity
L	Planned path for the entity.
$Station$	Used in the algorithm when traversing all stations
$Entity$	Used in the algorithm when traversing all entities
$Path$	Used in the algorithm when traversing all paths
$Path_{next}$	The path to be entered for an entity

A. Infrastructure Layer

The infrastructure layer builds the basic structure for a metro system which consists of stations and paths, and stores all the network connection data. The whole layer is built on the graph structure. Therefore, it is convenient for applying further path choice algorithm, such as the shortest path algorithm. In the graph structure, each node contains information related to stations, including ID_i, $Line_i$, entrance time t_i^e, stop time t_i^s and exit time t_i^x. Meanwhile, each arc represents information related to paths, including expect travel time $\overline{t_j^t}$, real travel time t_j^t and maximum capacity M_j.

Notice that the transfer information has not been stored in the graph. Thus, we perform a special procedure, called *remapping*, over all the stations. Normally, each transfer station is viewed as a single station. In the remapping procedure, each transfer station is divided into different stations, and the transfer information can be stored in arcs between remapped stations. Fig. 2 illustrates the effect of remapping. Notice that the station with $ID = 2$ is now remapped into stations with $ID = 2$ and $ID = 3$, and network connection is therefore changed.

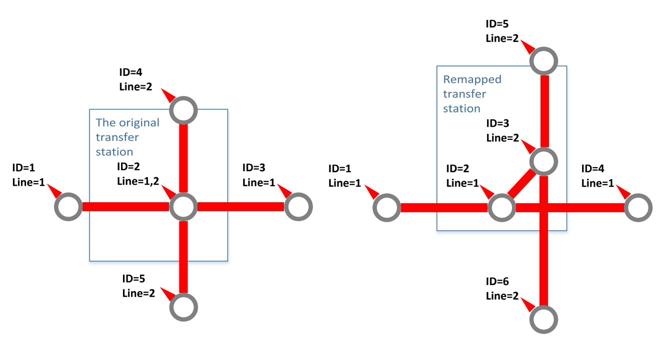

Figure 2. An illustrative example of the remapping process

We then need to calculate real travel time t_j^t . To consider different scenarios, we categorize arcs into normal arcs and transfer arcs. Arcs between two metro stops are defined as normal arcs, while arcs between remapped transfer stations are defined as transfer arcs. Two types of arcs model the trip on train and the trip in transfer station individually, and are depicted in Fig. 3.

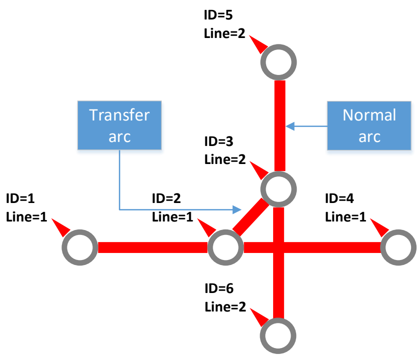

Figure 3. Two types of arcs in the simulation model

For normal arc, we assume that trains are on schedule. Thus, the real travel time is the expected travel time, which can be described as

$$t_j^t = \overline{t_j^t} \tag{1}$$

For transfer arc, the real travel time is the sum of transfer walking time and transfer boarding time, which can be written as

$$t_j^t = \overline{t_j^t} + t_j^b \tag{2}$$

where $\overline{t_j^t}$ reflects the average transfer walking time which can be obtained from empirical data. The transfer boarding time t_j^b is estimated as half of the headway in many studies [8]. However, such an approximation fails to reflect the queuing phenomenon in peak hours.

To solve this problem, we assume that each normal arc has a maximum capability for carrying passengers. The capability for transportation M_j for a normal arc is defined as follows.

$$M_j = \frac{c_j \overline{t_j^t}}{h_j} \tag{3}$$

A normal arc is defined as overloaded, if current passenger number N_j exceeds M_j . If the arc to board is overloaded, a passenger will wait until the arc has enough space, which simulates passengers waiting on the platform in peak hours. The corresponding algorithm will be discussed in the operation layer.

C. *Passenger Layer*

The passenger layer mainly formulates the decision making process of the passengers. We first formulate the cost for all potential travel paths, then apply algorithms to select the desired travel path for each passenger.

The generalized travel cost is proposed to model the cost of travel, which is a typical method adopted by many studies [3]. In metro system, The generalized travel cost V_{ij}^k can be written as the sum of time cost T_{ij}^k , comfort cost C_{ij}^k and travel expense cost E_{ij}^k , where C_{ij}^k and E_{ij}^k has been transformed into the unit of time:

$$V_{ij}^k = T_{ij}^k + C_{ij}^k + E_{ij}^k \tag{4}$$

For a typical trip, time cost consists of station entrance time t_i^e , real travel time t_j^t , train stop time t_i^s and station exit time t_i^x . The information can be obtained from the infrastructure layer. P_{ij}^k denotes the set of arcs that constitute the k th travel path from station i to station j .

$$T_{ij}^k = t_i^e + \sum_{p \in P_{ij}^k} t_p^t + t_j^x \tag{5}$$

Comfort cost is typically determined by the congestion status on the train and the transfer activity between lines. In our model, we assume that the congestion status on each train is similar, given any specific time. Thus, we use transfer activity to formulate the comfort cost. A study conducted in [9] shows that the cost of transfer activity can be viewed as a linear function of total transfer time and the number of times for transferring. The equation can be shown as follows, where λ and σ transform the unit of comfort cost into time.

$$C_{ij}^k = \lambda \sum_{m \in T_{ij}} t_m^c + \sigma n_{ij} \tag{6}$$

Travel expense cost is fixed for a given origin-destination pair, considering the fair policy in most metro systems. Therefore, this term will not affect the selection of path for passengers.

For path choice decision, two models are commonly used. The first model assumes the absolute rationality of passengers. It implements the shortest path algorithm and found the optimal solution. Although it slightly deviates from reality, it can be viewed as an adequate estimation. The second model assumes that passengers choose among possible paths based on probability, which is determined by the cost of that path. It better describes the behavior of passengers, and is adopted in our model.

We first implement the KSP (K shortest paths) algorithm to provide first K possible paths that minimize the generalized travel cost. The utility of a given path is in negative relationship with the cost, which is described as follows

$$U_{ij}^k = \frac{1}{V_{ij}^k} \tag{7}$$

Then the probability of choosing the k th path among K possible paths can be written as follows.

$$p_{ij}^k = \frac{\exp(U_{ij}^k)}{\sum_{k=1}^{K} \exp(U_{ij}^k)} \tag{8}$$

D. *Operation Layer*

The operation layer simulates the real transport operation, which is built on the basis of the passenger layer and the infrastructure layer. This layer will carry out real-time simulation, and provide outputs including transfer passenger flow, exit passenger flow and etc.

The operation layer mainly has three functions: packing passengers into entities with their path choice, accepting entities from stations and transporting entities through paths. The layer will update on each simulation step.

In each simulation step, the operation layer first packs each group of departure passengers into entities. Each entity is defined to include the following information: Time to live in current path TTL, number of passengers N, theoretical total travel time $T_{theoretical}$, real total travel time T_{real}, planned path L. Those entities are stored in stations, waiting for the acceptance of operation layer.

Entities are then accepted by the operation layer, and are stored in paths. The algorithm can be shown in Table II.

TABLE II. THE ALGORITHM FOR ACCEPTING ENTITIES

Algorithm 1 Accepting entities

Foreach *Station*
  Foreach *Entity*
    1. **If** $Path_{next}$ has enough space, **then**
      i. Copy *Entity* into $Path_{next}$, set TTL as the expected travel time $\overline{t_j^t}$ of $Path_{next}$
      ii. Delete *Entity* in *Station*
    2. **If** $Path_{next}$ will be overloaded if *Entity* is to be accepted, **then**
      i. Skip *Entity*
    3. Update the number of entities of $Path_{next}$, increase the real travel time for all the entities by a simulation step
  EndFor
EndFor

Entities are then transported within the operation layer, based on the following algorithm in Table III.

TABLE III. THE ALGORITHM FOR TRANSPORTING ENTITIES

Algorithm 2 Transporting entities

Foreach *Path*
  Foreach *Entity*
    1. **If** $TTL \le 0$, **then**
      a) **If** $Path_{next}$ is null, i.e. *Entity* is to exit the metro system, **then**
        i. Collect the information of *Entity*
        ii. Delete *Entity* in the *Path*
      b) **If** *Path* is a transfer path, and $Path_{next}$ will be overloaded if *Entity* is to be accepted, **then**
        i. Skip *Entity*
      c) **else**
        i. Copy *Entity* into $Path_{next}$, set the new TTL as the expected travel time $\overline{t_j^t}$ of that path
        ii. Delete *Entity* in *Path*
    2. **If** $TTL > 0$, **then**
      Decrease TTL by a simulation step
    3. Update the number of entities of $Path_{next}$, increase the real travel time for all the entities by a simulation step
  EndFor
EndFor

III. EMPIRICAL VERIFICATION

In this section, we examined our simulation model in a typical scenario of Beijing metro system. We verify our model based on the real exit passenger flow data, which is independent from model input.

The input data are collected on January 9th, 2013. The metro system includes 14 lines and 220 stations. The simulation model is programmed on MATLAB platform. The sketch map for Beijing metro system is depicted in Fig. 4.

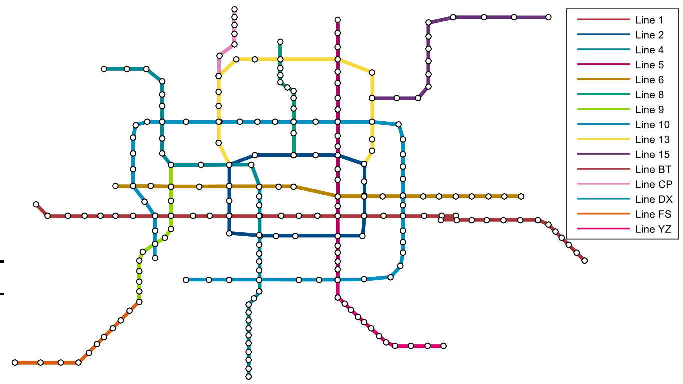

Figure 4. The sketch map for Beijing metro system on January 9th, 2013

Input data includes the OD data and the network connection data, which are obtained from the official source of Beijing metro system. The OD data records the origin, destination and departure time of each passenger. The network connection data includes connection state between stations, average travel time between stations, average transfer time, station entrance time and station exit time. Model parameters are set according to [10], and is finally adjusted as $\lambda = 1.5$ and $\sigma = 3\,\text{min}$ by grid search based on the model performance. Besides, we simplify the KSP algorithm by applying the shortest path algorithm to save computational cost.

Many outputs can be obtained from our simulation model. Macro statistics including the exit passenger flow, the transfer passenger flow and the section passenger flow can be obtained. Micro statistics such as the real travel time for each passengers can also be collected.

Statistics listed above are hard to collect in reality, except for the exit passenger flow. Thus, real value of the exit passenger flow, provided by Beijing Metro Network Control Center, is compared with the simulation value for the purpose of verification. The real value of the exit passenger flow is collected in each station, and is independent from the OD data used as our model input. Note that the collecting method of exit passenger flow is different from the OD data, and for some newly built stations the deviation can be great. The simulation results show that the proposed model can provide high performance, with an average error rate less than 8.8%. The average exit flow from all the stations is depicted in Fig. 5 and Fig. 6.

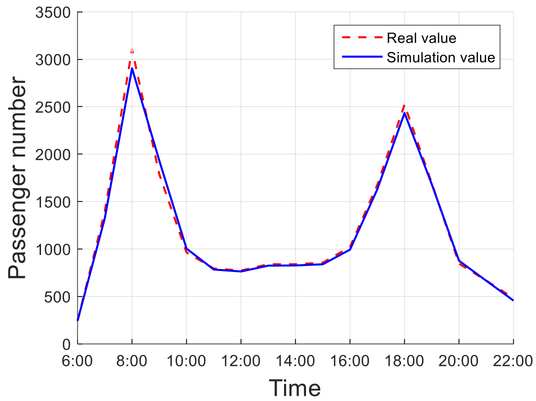

Figure 5. The average exit flow from all stations

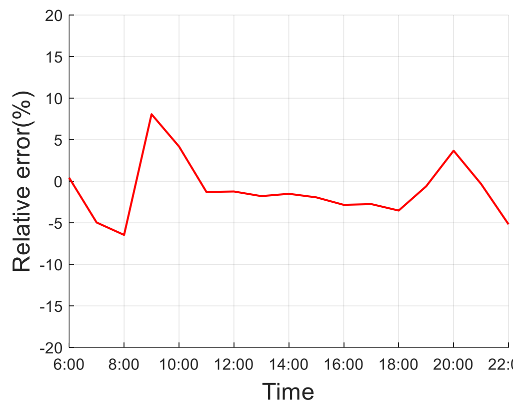

Figure 6. The relative error of the average exit flow from all stations

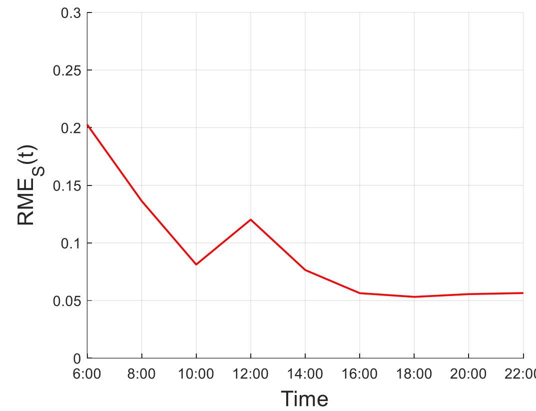

Figure 7. Simulated results for $RME_S(\mathrm{t})$

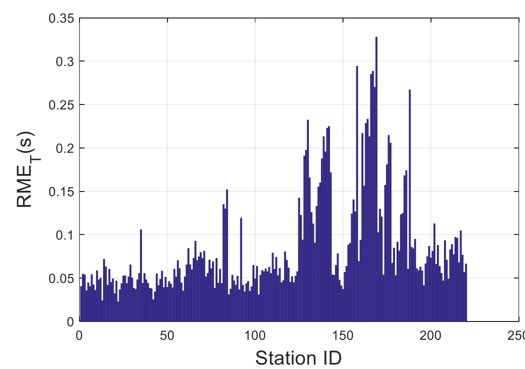

Figure 8. Simulated results for $RME_T(\mathrm{s})$

Relatively mean error (RME) is used to further reveal the spatial and temporal bias of our model output. Suppose there are S stations and T time periods, n_s^t and $\overline{n_s^t}$ denotes the real and simulated exit flow of station s in time t respectively. Then the $RME_S(\mathrm{t})$ (RME for all stations) and the $RME_T(\mathrm{s})$ (RME for all time periods) is defined as follows.

$$RME_S(\mathrm{t}) = \frac{1}{S}\sum_{s=1}^{S}\frac{\left|\overline{n_s^t} - n_s^t\right|}{n_s^t} \tag{9}$$

$$RME_T(\mathrm{s}) = \frac{1}{T}\sum_{t=1}^{T}\frac{\left|\overline{n_s^t} - n_s^t\right|}{n_s^t} \tag{10}$$

The $RME_S(\mathrm{t})$ shows that our model exhibits higher bias in peak hours, around 6:00 to 9:00 and 19:00 to 22:00. An explanation is that the congestion in peak hours prolongs the train stop time and passenger walking time, thus delay the exit of passengers. The delay in peak hours is support in Fig. 6, where the simulated exit flow has a negative bias before peak times and has a positive bias after peak times. The $RME_T(\mathrm{s})$ shows that higher bias exists on stations with ID higher than 100. Those stations are typically newly built suburban stations and have less passenger flow, which is more random and may account for the high bias. We can adjust the bias in peak hours in further application, and smooth the randomness in suburban stations by using the OD data in more days.

IV. Application

To alleviate the congestion level in peak hours, a policy of staggered shifts is commonly carried out. However, the quantitative effect of the policy on metro systems is hard to analyze. Instead of carrying out field test, we can use our metro simulation model to give a numerical analysis for its effect, which is convenient and cost-saving.

The scenario is simplified as follows. Suppose the central business district (CBD) in Beijing delays its office hour for one hour. We then assume that the departure time for commuters whose destination is GuoMao, the center of CBD, is delayed for one hour. The effect can be formulated by altering the OD data. The proportion of commuters is defined as α, which is set as $\alpha = 0.8$ in our scenario. The change should only take effect in peak hours, which is set from 6am to 10am. To examine the effect of this policy, we analyze the exit flow from GuoMao. In addition, the transfer flow of nearby transfer stations is also discussed, including JianGuomen and DongDan. Their relationship is shown in Fig. 9.

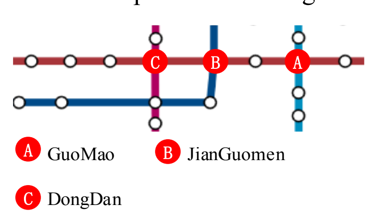

Figure 9. The spatial relationship of GuoMao, JianGuomen and DongDan

We compare the statistics before and after adjusting office hour. The peak of exit passenger flow of GuoMao is delayed and reduced as expected, which is shown in Fig. 10. In addition, the pressure for nearby transfer stations is also relieved, and our model is able to yield a quantitative prediction, shown in Fig. 11 and Fig. 12. It is shown that the nearer the transfer station is, the greater the effect is.

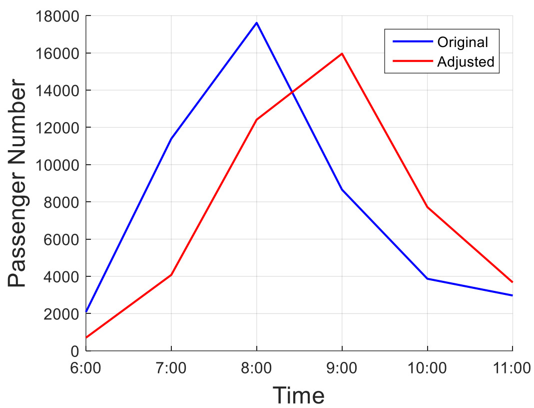

Figure 10. The exit passenger flow of GuoMao before and after adjusted

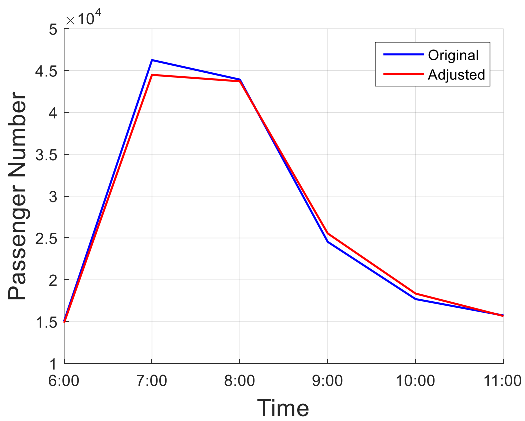

Figure 11. The transfer passenger flow of JianGuomen before and after adjusted

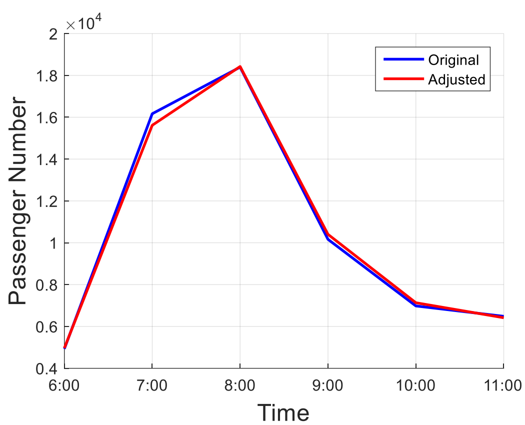

Figure 12. The transfer passenger flow of DongDan before and after adjusted

V. Conclusion

This article presents a hierarchical, empirical and adaptive simulation model for multi-line metro systems. Its performance is validated by real data from Beijing metro system. Our work shows that

- The proposed three layer framework can explicitly describe the operation of metro systems and adapt various settings.
- By only using the OD data and the network connection data, our model can carry out effective simulation, even in large-scale systems such as Beijing metro system.
- Our model can serve as a convenient tool for analyzing the effect of management policies on metro systems.

Acknowledgment

This work was supported by National Basic Research Program of China (973 Project) 2012CB725405, the National Science and Technology Support Program (2014BAG03B01), National Natural Science Foundation China 71301083, 61273238, and Beijing Municipal Science and Technology Project (D15110900280000). We sincerely thank Prof. Enjian Yao, Yongsheng Zhang and Shasha Liu who are with Beijing Jiaotong University for providing the empirical data and offering constructive suggestions.

References

[1] A. Cipriano and F. Reyes, “On-line passenger estimation in a metro system using particle filter,” *IET Intell. Transp. Syst.*, vol. 8, no. 1, pp. 1–8, 2014.

[2] P. Grube, F. Núñez, and A. Cipriano, “An event-driven simulator for multi-line metro systems and its application to Santiago de Chile metropolitan rail network,” *Simul. Model. Pract. Theory*, vol. 19, no. 1, pp. 393–405, 2011.

[3] X. Yao, P. Zhao, K. Qiao, and W. Li, “Modeling on Dynamic Passenger Flow Distribution in Urban Mass Transit Network,” *LISS 2012*, pp. 469–476, 2013.

[4] D. Heimburger, A. Herzenberg, and N. Wilson, “Using simple simulation models in operational analysis of rail transit lines: Case study of Boston’s Red line,” *Transp. Res. Rec. J. Transp. Res. Board*, no. 1677, pp. 21–29, 1999.

[5] X. Sun, S. Zhang, H. Dong, and H. Zhu, “Optimal train schedule with headway and passenger flow dynamic models,” *IEEE ICIRT 2013 - Proc. IEEE Int. Conf. Intell. Rail Transp.*, vol. 1, no. 1, pp. 307–312, 2013.

[6] H. Koutsopoulos and Z. Wang, “Simulation of urban rail operations: Application framework,” *Transp. Res. Rec. J. Transp. Res. Board*, 2007.

[7] J. Zhao, C. Tian, and F. Zhang, “Understanding temporal and spatial travel patterns of individual passengers by mining smart card data,” *17th Int. IEEE Conf. Intell. Transp. Syst.*, pp. 2991–2997, 2014.

[8] Z. Mingbao, C. Hui, Y. Liqiang, and C. Ying, “Public transport connecting preference model based on generalized travel utility,” *2009 2nd Int. Conf. Intell. Comput. Technol. Autom. ICICTA 2009*, vol. 3, no. 3, pp. 673–676, 2009.

[9] Y. Zheng, W. Guo, Y. Zhang, and J. Hu, “A Generalized Comfort Function of Subway Systems Based on a Nested Logit Model,” vol. 19, no. 3, pp. 300–306, 2014.

[10] S. S. Liu, E. J. Yao, and Y. S. Zhang, “Personalized route planning algorithm for urban rail transit passengers,” *J. Transp. Syst. Eng. Inf. Technol.*, vol. 14, no. 5, 2014.