A Data-Driven and Optimal Bus Scheduling Model With Time-Dependent Traffic and Demand

Yuan Wang, Dongxiang Zhang, Lu Hu, Yang Yang, and Loo Hay Lee, *Senior Member, IEEE*

***Abstract*— Urban bus companies have collected a tremendous amount of travel data from passengers in the past years. In spite of great value for bus schedule optimization, these data have not been fully exploited. In this paper, we leverage hundreds of millions of bus transaction records, generated when passengers board and alight, to infer time-dependent traffic and customer demand. When the traffic and demand information are available, we build an optimal model to schedule the departure time of each bus service with the objective of minimizing the average waiting time. Experimental results show that compared with the existing bus scheduling system, our model can help reduce the waiting time by a wide margin.**

***Index Terms*— Demand-driven, time-dependent traffic, bus scheduling.**

I. INTRODUCTION

TRAFFIC congestion has posed a substantial threat to urban cities in terms of tremendous lost time and productivity, air pollution and wasted energy. Promoting public transportations is an effective way to alleviate the headache. Since buses and metros provide much higher resource utilization than private cars, the number of vehicles on the road can be significantly reduced. With more and more buses in operation, passengers can enjoy much better quality of service, with shortened waiting time and fewer bus transitions for long journeys. However, it imposes a great burden on the bus service providers in controlling the operating budget.

We take SBS (Singapore Bus Service) Transit in Singapore, a pioneering country dedicated in promoting public transportations, as an example to show the dilemma between improving service quality and controlling operating budget, which many modern cities are confronting. In order to comply with the slogan “more buses, better rides” from the Land Transportation Authority in Singapore, more and more buses are put on the road to enhance connectivity and improve bus service levels. In consequence, the operating budget has been growing dramatically in recent years in terms of both hardware and manpower expenses.

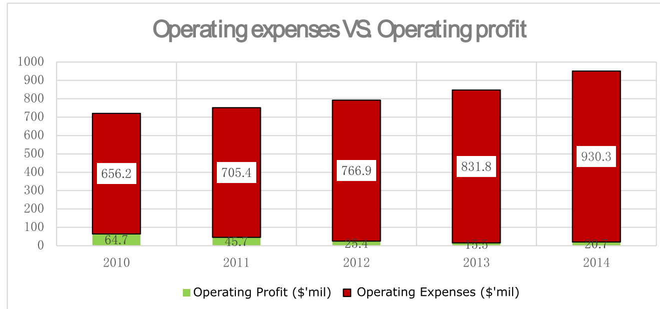


Fig. 1. Operating expenses and profit of SBS Transit from 2010 to 2014.

Figure 1 illustrates the operating expenses and profit of SBS Transit in the last 5 years.[1] As can be seen, to meet the Quality of Service (QoS) Standards in Singapore [2], the operating expenses increase dramatically year by year, resulting in a decreasing profit since 2010. Note that the slight rising of profit in 2014 is mainly due to the increasing of bus fare by 3.2% in that year [3].

The above statistics partly reveal that existing urban bus scheduling systems are neither scalable nor sustainable because they were designed to be

1) **Blindfold:** Existing bus scheduling systems are mainly driven by expertise instead of (real-time) data. In Singapore, millions of transaction records about the motion patterns of passengers are generated each day. The record consists of the spatial and temporal information about their boarding or alighting timestamps and stations. Such enormous volumes of information in the past history are not fully exploited for customer demand prediction and bus route optimization. Moreover, real-time customer travel requests become easier to collect due to the prevalence of smartphones. With the technologies, the whole taxi industry has been shifting gradually from moving blindly on streets to server precisely on demand. Analogously, existing bus scheduling systems can also benefit from such trend.
2) **Unoptimized:** Existing bus allocation and scheduling mechanisms are far from optimal for profitability. Based on the current scheduling system, more buses have to be put in operation to reduce the average waiting time and keep the degree of customer satisfaction.

Manuscript received October 22, 2015; revised January 12, 2016, March 17, 2016, June 13, 2016, August 24, 2016, and October 30, 2016; accepted December 8, 2016. Date of publication January 31, 2017; date of current version August 28, 2017. This work was supported in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions, in part by the Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, in part by the National Nature Science Foundation of China under Grant 61602087 and Grant 61632007, and in part by the MOE Academic Research Fund under Grant R-266-000-093-592 and Grant R-266-000-082-731. The Associate Editor for this paper was P. Ye.

Y. Wang, L. Hu, and L. H. Lee are with the Department of Industrial and Systems Engineering, National University of Singapore, Singapore 117576 (e-mail: iseway@nus.edu.sg; isehulu@nus.edu.sg; iseleelh@nus.edu.sg).

D. Zhang and Y. Yang are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: zhangdo@uestc.edu.cn; yang.yang@uestc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2644725

[1]The data is derived from the annual report of SBS Transit in the year of 2014, which is available online [1]

In this paper, we propose a more economic and sustainable strategy to enhance the profitability of bus companies. Our goal is to maintain the current service standard in terms of average waiting time with fewer buses, in the means of optimizing the scheduling algorithm from blindfold to data-driven. We leverage the historical bus operating data to construct a time-dependent bus network which is associated with the traffic and demand information in different periods of a day. The whole dataset contains hundreds of millions of records for all the people in the city and these data can be used to infer the historical traffic and demand patterns. More specifically, we split each day into a fixed number of time intervals. For each interval, we estimate the average travel time between each pair of neighbouring bus stations as well as the amount of waiting customers in each bus station. Based on the time-dependent bus network, we propose an optimal model to schedule the departure time of each bus service to minimize the average waiting time and the model can be solved by standard MIP solvers such as Cplex.

In summary, the contributions of this paper include

1) We propose the concept of time-dependent bus network augmented with traffic patterns and customer demands to drive the design of bus scheduling systems.
2) We propose effective algorithms to construct the time-dependent bus network using hundreds of millions of historical transaction records.
3) We devise an optimal scheduling model that takes into account the time-dependent traffic and demand information to minimize the average waiting time.
4) We conduct simulation experiments with real traffic conditions and customer demand. The experimental results show that our model can reduce the waiting time by a wide margin.

The remaining of the paper is organized as follows. In Section II, we review existing work on existing bus scheduling algorithms and traffic estimation techniques. We propose the time-dependent bus network and as well as its construction algorithms in Section IV-C. The optimal model is proposed Section IV. We introduce the experimental setup and report the extensive experiment results on real customer demand in Section V. Section VI concludes the paper.

II. Related Work

The early studies of bus scheduling design were focused on determining the optimal frequency with the assumption of equal intervals [4]–[8]. In the last century, Ceder proposed to determine the bus frequency based on passenger count data [5]. Salzborn proposed a continuous solution methodology to determine the optimal frequency with the objective of minimizing bus fleet size and passenger waiting time that can work in a single route [4] or multiple routes [6]. Verbas and Mahmassani [7] studied how to allocate the optimal number of buses for a service line as the best trade-off between maximizing the number of passengers and minimizing the operational cost. Martinez et al. proposed a new mixed integer linear programming (MILP) formulation and tested both exact and approximate methods to determine the time interval [8].

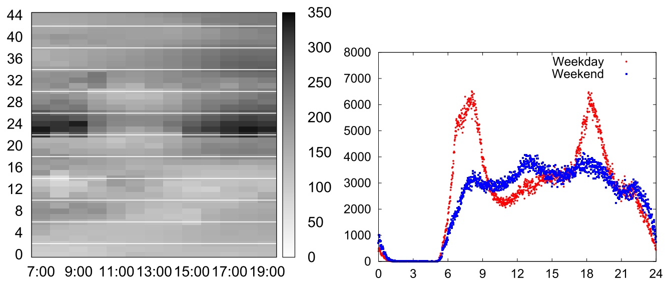
With the development of computing power and optimization techniques, a line of work on the departure time optimization with various objectives have been proposed [9]–[17]. Ceder and Philibert [9] considered bus timetable design problem with the objective of minimizing the maximum load upon the vehicles so as to avoid overcrowding. In [10] and [11], bus scheduling interval and the associated bus model (e.g., single decker or double decker bus) were considered as two factors to optimize in the objective function. [12] attempted to optimize the timetable by employing deterministic parameters within each planning period such as morning rush hour with the assumption of evenly spaced headway. In [15], runtime optimization was investigated in order to improve the punctuality of routes. Moreover, some studies [13], [14] considered the synchronization in the transfer bus stations to enable the transfer of passengers from one route to another with minimum waiting time.

Timetable optimization has also attracted great interest in the railway systems. Zhou and Zhong [18] studied double-track train scheduling problem minimize both the expected waiting times for high-speed trains and the total travel times of high-speed and medium-speed trains. In [19], a bi-objective evolutionary approach was proposed to cater for energy saving. Interested readers can refer to [20] for more work in this area.

However, the time-dependent customer demand was rarely considered in the bus scheduling optimization. An exceptional case is [21], in which passenger count information is extracted to determine the scheduling frequency of bus services. For railway systems, there exist several demand-driven work in the timetable design with the objective of minimizing the total waiting time [22], [23]. Sun *et al.* [24] designed timetable for metro or railway services by considering passenger demand. An integer linear programming model was proposed for exact solutions. Niu *et al.* [25] considered skip-stop patterns when designing timetables for express trains and proposed several nonlinear integer programming models with linear constraints. In these work, the travel time between two consecutive stations is modelled to be constant, while in bus scheduling system, the travel time is time-dependent and affected by the traffic patterns.

III. Time-Dependent Bus Network

A typical bus scheduling system in a modern city consists of hundreds to thousands bus services operating on the well planned routes in a city. Each bus service may be associated with two bus routes in opposite directions or simply a single route in a closed loop. We can treat the former case as two bus routes that can be optimized independently. The reason is that most buses have the LED to dynamically display the service number. Thus, a physical bus is not bound to a particular service. It can be assigned to any service operated in a bus terminal station. All the bus routes form a network, in which the node is a bus station. Two nodes are connected by an edge as long as they appear consecutively in a bus route. The weight of each edge is the average travel time between the two connected bus stations. It is worth noting that data anomalies are common in automated data and fortunately, such anomalies and missing data have been cleaned by the data provider.



(a) (b)

Fig. 2. Time-dependent traffic and demand. (a) traffic. (b) demand.

A. Bus Dataset Description

The bus dataset used in this paper was provided by the public transportation agency in a modern city in China. It contains more than 0.3 billion transaction records for a period of six months. When a user boards a bus from a station, he/she will tap a card on an on-board device and the system keeps a record of card id, bus service id, bus station id and boarding timestamp. When the user alights at another station, he/she will tap again and similar information is recorded. In the dataset, the identities of passengers have been anonymized for privacy protection. Each record contains the unique customer id as well as his complete information of the ride, including the boarding and alighting bus stations. The record also preserves the travel distance between these two stations and the corresponding travel time. In this paper, we study the extraction of time-dependent traffic and demand from bus data. Various other applications derived from smart card data can be found in [26]–[29].

B. Time-Dependent Traffic and Demand

In most of the previous bus scheduling systems, the travel time between two bus stations is considered to be constant. The optimization based on such assumption is not realistic and may lead to misleading insights. In fact, the travel time between two bus stations varies significantly in different periods of a day, which consequently affect the estimated arrival time at each bus station. When developing the bus scheduling optimization model, we argue that such important factor should be considered. For example, we plot the traffic pattern for a bus service in Figure 2(a) in both temporal and spatial dimensions. The x-axis represents the time slots in a day (temporal dimension) and y-axis represents the sequence of bus stations in the route (spatial dimension). For each time slot and each bus station, we plot the average travel time from the station to the next one in that period. The travel time is estimated in the unit of second and plotted by color. The light color means the traffic is clear and the dark area means heavy traffic. We can see that the bus stations exhibit different spatial-temporal traffic patterns, which should be taken into account in the optimization model.

The customer demand also varies dramatically in different periods of a day. Figure 2(b) shows the total number of waiting customers, derived from the bus dataset, in different periods of a day. We can see that in weekdays, there are two spikes in the morning and evening peak hours; the amount of demand drops significantly after 7pm. In weekends, the demand vary slightly in the day time, with two non-significant peaks in the lunch and dinner hours.

The lack of considering time-dependent traffic and demand renders the previous bus scheduling systems fail to work well in real-life scenarios. To bridge the gap between ideal models and practice in real system, we propose the concept of time-dependent bus network augmented with traffic patterns and customer demands to drive the design of proof-of-concept bus scheduling models.

C. Extracting Time-Dependent Traffic

For each edge connecting two consecutive bus stations s_i and s_{i+1} in the bus network, we split a day into Z intervals. Each edge in the bus network is thus augmented with a Z-dimensional vector in the form of

$$(\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_Z)$$

as a discrete representation of time-dependent traffic patterns. We assume that the travel cost between two bus stations remains consistent in the same time interval and is denoted by λ_z, where z is an index for the temporal partition. Z is a parameter used to adjust the granularity and control the precision of the bus scheduling system. For example, if we set $Z = 288$, each interval is 300 seconds and the whole day is split into $[00:00, 00:05), [00:05, 00:10), \ldots, [23:55, 00:00)$. From the arrival time at bus station s_i, which is estimated by the tapping time of the first boarding passenger at that station, we can use the vector as a reference to infer the travel cost from s_i to the next station s_{i+1}. When Z is set to be a large value, we can capture more precise traffic patterns but with more computation cost in the optimization model as more input variables are involved.

However, it is a challenging task to fill all the variables λ_z from the historical transactions of the bus dataset. Although each bus record provides the time cost between two stations in a route, these two stations are not necessarily consecutive. Many of the variables cannot be directly extracted from the historical dataset. To solve the issue, we propose an effective traffic condition estimation method as well as an interpolation technique to fill the missing variables.

Let t_s denote the boarding time and t_e denote the alighting time. If the boarding bus station and the alighting bus station happen to be two adjacent nodes in the bus network, we can set $\lambda_z = t_e - t_s$, where z is the interval in which the boarding time is located and $z = \lceil \frac{Z \cdot t_s}{86400} \rceil$. This is because each day has 86400 seconds. When we split it into Z intervals, each interval contains $\frac{86400}{Z}$ seconds. It is possible that multiple evidences for λ_z are found in the dataset. For example, there may be multiple customers travelling from s_i to the next bus station s_{i+1} in different bus services, generating multiple transactions with boarding station s_i and alighting station s_{i+1}. In this case, we use linear regression methodology to find λ_z with the minimum mean square error (MSE) among all the instances. In other words, given a set of $\{t_e^1 - t_s^1, t_e^2 - t_s^2, \ldots, t_e^n - t_s^n\}$, we want to find λ_z such that

$$\sum_{j=1}^{n} (t_e^j - t_s^j - \lambda_z)^2 \tag{1}$$

Algorithm 1 Time-Dependent Traffic Estimation

1: **for** each edge (b_s, b'_s) in the bus network **do**
2: **for** $z \leftarrow 0; z < Z; z$++ **do**
3: init an empty list L_z and set c_z to 0
4: **end for**
5: **end for**
6: init a priority queue pq
7: **for** each cell record $R = (t_s, t_e, b_s, b_e)$ in the dataset **do**
8: $z \leftarrow \lceil \frac{Z \cdot t_s}{86400} \rceil$
9: **if** (b_s, b_e) is an edge in the bus network **then**
10: $\lambda_z \leftarrow \frac{t_e - t_s + c_z \lambda_z}{c_z + 1}$
11: $c_z \leftarrow c_z + 1$
12: **else**
13: insert R into the sorted list L_z in bus station b_s
14: insert R into the priority queue pq
15: **end if**
16: **end for**
17: **while** pq is not empty **do**
18: $(t_s, t_e, b_s, b_e) \leftarrow pq$.pop()
19: $z \leftarrow \lceil \frac{Z \cdot t_s}{86400} \rceil$
20: **UpdateList**(t_s, t_e, b_s, b_e)
21: **end while**

is minimized. Obviously, the optimal λ_z should be set to

$$\lambda_z = \frac{1}{n} \sum_{j=1}^{n} (t_e^j - t_s^j) \tag{2}$$

When $t_e - t_s$ covers the travelling time across multiple bus stations $s_i, s_{i+1}, \ldots, s_j$, we only know the total cost of the whole trip but our goal is to infer the travelling time for each pair of neighbouring stations. To leverage such information, we propose a divide-and-conquer strategy. The intuitive idea is that if we know the time cost from s_i to s'_i and the cost from s_i to s_j, both starting from the same time interval, then we can infer the time cost from s'_i to s_j starting from the time interval in which the bus arrives at s'_i. Furthermore, if (s'_i, s_j) happens to be an edge in the bus network, i.e., they appear consecutively in a bus route, we have successfully inferred the expected travel time of an edge in a certain time period. The pseudo codes of time-dependent traffic estimation are presented in Algorithm 1 and Algorithm 2.

In the initialization steps, for each edge (b_s, b'_s) in the bus network, we maintain an array λ associated with a counter array c, both with length Z. We use c_z to denote the z-th entry of the array. Each list L_z stores all the records whose boarding station is b_s and boarding time is located in the z-th time interval. The consecutive boarding-alighting patterns are directly used for traffic estimation and the remaining patterns are inserted into a priority queue. Then, we iteratively pop the record from the top of the heap. For each record, we call the recursive function **UpdateList** to check if the travel time between any sub-segment of the original trip can be inferred. The function **UpdateList** works by scanning all the records starting from the same bus station and time interval. If we meet a record (t'_s, t'_e, b'_s, b'_e) such that $t'_e - t'_s < t_e - t_s$, we can update the travel cost from b_e to b'_e to be $t'_e - t_e$.

Algorithm 2 UpdateList(t_s, t_e, b_s, b_e)

1: **if** (b_s, b_e) is an edge in the bus network **then**
2: $\lambda_z \leftarrow \frac{t_e - t_s + c_z \lambda_z}{c_z + 1}$
3: $c_z \leftarrow c_z + 1$
4: **else**
5: $z \leftarrow \lceil \frac{Z \cdot t_s}{86400} \rceil$
6: **for** each record (t'_s, t'_e, b'_s, b'_e) in L_z **do**
7: **if** $t'_e - t'_s < t_e - t_s$ **then**
8: $z' \leftarrow \lceil \frac{Z \cdot t_e}{86400} \rceil$
9: insert (t'_e, t_e, b_e, b'_e) into $L_{z'}$ in bus station b_e
10: **UpdateList**(t'_e, t_e, b_e, b'_e)
11: **end if**
12: **end for**
13: **end if**

Note that **UpdateList** is defined as a recursive function and all the possible sub-segments will be processed in the same way. If the sub-segment contains only two bus stations forming an edge in the network, we can use the inferred time cost to estimate the corresponding λ, as shown in lines 1-3 in Algorithm 2. The whole procedure terminates when all the records in the priority queue have been processed and pq becomes empty.

It is possible that not all the entries can be derived from the above algorithms. We propose a k-nearest neighbour interpolation method [30] to fill those missing entries by utilizing the travel time information that is spatially and temporally close. Intuitively, the travel speed from s_i to its neighbouring station s_{i+1} is similar to the travel speed from s_{i-1} to s_i or from s_{i+1} to s_{i+2} in similar time periods. Thus, we can expand from station b_s and the z-th time interval both spatially and temporally until we meet the first non-empty entry, which will be used as an estimation of the missing entry.

D. Extracting Time-Dependent Demand

Even though the bus dataset contains all transactions of city-scale passengers, extracting time-dependent demand, i.e., the arrival time distribution of passengers at each bus station waiting for a particular bus service, is not a trivial task. This is because the bus records only incorporate the boarding time and alighting time, but the information of arrival time is not available. Since our final objective is to minimize the total waiting time, it is challenging to estimate the waiting time of passengers if their arrival time is unknown.

To solve the issue, our idea is to assume that the passenger waiting time follows roughly the same distribution among different bus stations of a particular bus service. Note that different bus services can be associated with different waiting time distributions as each service has its own scheduling policy. Under the assumption, our task is simplified to estimating the waiting time distribution at one bus station for a particular service. Then, we can detect the transition patterns from the bus dataset. If a passenger first takes bus service A at time t_1, alights at a certain bus station s_m at time t_2 and takes another service B at time t_3 to the destination, then $t_3 - t_2$ is the waiting time at bus station s_m for service B.

TABLE I
INPUT VARIABLES

M	the number of stations in the bus route for scheduling optimization
P	the number of waiting passengers derived from the bus dataset
K	the number of buses allocated to the bus route for scheduling optimization
Z	the number of intervals for time-dependent traffic estimation
m	the index of bus station ($1 \leq m \leq M$)
k	the index of the bus ($1 \leq k \leq K$)
p	the index of passenger ($\leq p \leq P$)
z	the index of interval ($1 \leq z \leq Z$)
D_f	the departure time of the first bus
D_l	the departure time of the last bus
λ_{mz}	the expected travel time from bus route from bus station s_m to s_{m+1} at the z-th interval
B_p	the bus station at which the p-th passenger is waiting ($1 \leq p \leq P$)
A_p	the arrival time of the p-th passenger ($1 \leq p \leq P$)

In our implementation of detecting such transition patterns, we set two strict requirements in order to achieve high precision. First, we set the time interval between two successive bus transaction records for the same customer id in the same day to be less than 30 minutes. Second, such a pattern must occur more than 10 times within one month. In this way, each detected waiting time is considered as one sample and we use empirical probability [31], which is the ratio of the number of outcomes in which a specified event occurs to the total number of trails, to estimate the probability of waiting for k seconds at a bus station. With sufficient number of samples, the empirical probability can be close to the true distribution.

Even though the customer demand may be underestimated because passengers may use coins or evade fare [32], [33], our data-driven technique can still provide a rough estimation for customer demand and waiting time. More careful estimation strategies are encouraged. With the provenance of bus mobile apps, passengers can adapt to the bus scheduling by sending a request for the next bus arrival time so as to minimize the waiting time on their own efforts. Each enquiry specifies a concrete demand at a timestamp from a certain bus station. By collecting and analyzing a sufficient amount of such enquiry data, we can get a more accurate picture about periodic user demand to complement the optimization model proposed in this paper.

IV. Optimal Bus Scheduling Model

In this section, we study the optimal bus scheduling modelling with the objective of minimizing the total waiting time when the time-dependent traffic and demand information are available. Without loss of generality, we use a bus route $s_1 \rightarrow s_2 \rightarrow \ldots \rightarrow s_M$ with M bus stations as an instance. The remaining bus routes can be processed in the same way to obtain their optimal schedules.

We first formulate the *input variables* of our model, as shown in Table I. These variables fall into three categories: traffic-related, passenger-related and scheduling-related. The traffic variables include Z and λ and their derivation have been introduced in Section IV-C. The passenger-related variables include P, B_p and A_p, where P refers to the total demand, B_p is the boarding station and A_p is the arrival time of a passenger. The bus-related variables include M, K, A_f and A_l, where M is the number of bus stations in the route instance, A_f and A_l are the departure time of the first and last bus for the route. K determines the number of buses allocated to the route.

TABLE II
DECISION VARIABLES AND DEPENDENT VARIABLES

x_k	the departing time of the k-th bus from the terminal station
V_{mk}	the arrival time of the k-th bus at the bus station s_m
τ_p	the bus onto which the p-th passenger boards
w_p	the waiting time of the p-th passenger

Table II illustrates the *decision variable* x_k and several dependent variables used in the optimization model. Given K allocated buses, the task is to determine the departure time for each bus. Let x_k denote the departure time of the k-th bus and V_{mk} denote the arrival time of the k-th bus at station s_m. Since time-dependent traffic is considered in the model, the value of V_{mk} is affected by the bus departure time x_k and the traffic conditions in the corresponding time periods. τ_p determines the boarding bus for a waiting passenger. In our model, we assume that the bus always has capacity for the waiting passengers. Thus, each passenger will be assigned to the first bus that arrives subsequently. Finally, w_p is a variable to preserve the actual waiting time of each customer.

Up to here, we can model the optimization problem as an MIP (mixed integer programming) problem with the objective to minimize the total waiting time:

$$\min \sum_{p=1}^{P} w_p$$

subject to :

$$x_1 = D_f \tag{3}$$
$$x_K = D_l \tag{4}$$
$$x_k \leq x_{k+1} \tag{5}$$
$$V_{1k} = x_k \tag{6}$$
$$V_{mk} = V_{(m-1)k} + \lambda_{(m-1)j} \quad \text{where} \quad j = \left\lceil \frac{Z \cdot V_{(m-1)k}}{86400} \right\rceil \tag{7}$$
$$\tau_p = \min_k (V_{B_p k} > A_p) \tag{8}$$
$$w_p = V_{B_p \tau_p} - A_p \tag{9}$$

Since the departure time of the first and last bus are important information we need to comply with the scheduling policy, the values of x_1 and x_K have in fact been determined. There are $K - 2$ values of x_k remaining to be optimized. Since the arrival time of the k-th bus at the terminal station is essentially V_{1k}, we have Constraint 6. In Constraint 7, we define the relationship between V_{mk} and $V_{(m-1)k}$ based on the time-dependent traffic, i.e, when the k-th bus arrives at the station s_{m-1}, the expected arrival time at the next bus station is dependent on the traffic condition. Since the travel time between each pair of consecutive bus stations is available, we can use $V_{(m-1)k}$ to infer the expected travel time from s_{m-1} to s_m. When V is determined, we can infer which bus each passenger boards in Constraint 8 and finally determine the waiting time in Eqn. 9.

A. Reformulation of the Model

Although the above integer programming model can be directly solved using a standard optimization package, it could take days to solve a moderate size of the problem because many of the variables are dependent on the departure variables x_i and difficult to be optimized.

To make the problem algorithmically tractable, we need to reformulate the problem using binary variables. In our new model, we use *minute* as the finest granularity to reduce the computation complexity. We split a day of 24 hours into 1440 minutes and use the n-th minute ($1 \le n \le 1440$) to model the arrival time of a passenger or bus at a bus station. The passengers arrive at the same minute are grouped together for optimization. For example, if there are 5 passengers arriving at the same bus station from 8:00am to 8:01am, they will form a group of passengers arriving at the 481-th minute. In this way, the number of passengers is reduced from P persons to G groups, where $G < P$.

Based on the new temporal partitioning, we need to transform the temporal dimensions in the input variables λ, B and A. Originally, a day is split into T_t intervals and each interval contains $1440/T_t$ minutes. For the minutes in the same interval, we can simply assign them with the same traffic condition for the matrix expansion. For matrices B and A, the temporal granularity is updated from second to minute. After these transformations, we introduce a set of new binary variables. The first binary variable is used to identify whether the k-th bus arrives at the m-th station in the n-th minute.

$$u_{mkn} = \begin{cases} 1, & \text{if bus } k \text{ arrives at station } m \text{ in the } n\text{-th minute} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

The second binary variable v models the boarding time of the g-th group of passengers.

$$v_{gn} = \begin{cases} 1, & \text{if the } g\text{-th group board at the } n\text{-th minute} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

The waiting time for the g-th group is reformulated as

$$w_g = \sum_{n=1}^{1440} (v_{gn} \cdot n - A_g) \tag{12}$$

Let η_g denotes the number of passengers in group g and the new objective function becomes

$$\min \sum_{g=1}^{G} \eta_g \cdot w_g$$

subject to :

$$\sum_{n=1}^{1440} u_{mkn} = 1 \tag{13}$$

$$\sum_{n=1}^{1440} v_{gn} = 1 \tag{14}$$

$$\sum_{n=1}^{1440} u_{11n} \cdot n = D_f \tag{15}$$

$$\sum_{n=1}^{1440} u_{1Kn} \cdot n = D_l \tag{16}$$

$$\sum_{n=1}^{1440} u_{1kn} \cdot n \ge \sum_{n=1}^{1440} u_{1(k-1)n} \cdot n \tag{17}$$

$$\sum_{n=1}^{1440} u_{mkn} \cdot n = \sum_{n=1}^{1440} u_{(m-1)kn} \cdot n + \sum_{n=1}^{1440} u_{(m-1)kn} \cdot \lambda_{mn} \tag{18}$$

$$\sum_{n=1}^{1440} v_{gn} \cdot n \ge A_g \tag{19}$$

$$\sum_{k=1}^{K} u_{mkn} = v_{gn} \tag{20}$$

In the above constraints, Eqns 13 and 14 render u and v to be binary variables. The departure time of the first and the last bus are determined by Constraints 15 and 16. In Constraint 17, the departure time of the $(k-1)$-th bus is required to be earlier than the k-th bus. The time-dependent traffic constraint is reflected in Eqn. 18 to capture the relationship between the arrival time at the $(m-1)$-th and m-th bus stations for the same bus. Constraint 19 requires the passenger boarding time to be larger than or equal to his/her arrival time and the boarding time must be consistent with one of the bus arrival time at the same station, as shown in Constraint 20.

B. Capacity Constrained Optimization

The optimization model aforementioned assumes unlimited seats. However, in reality, we need to consider the bus capacity constraint and assume that each bus can accommodate a maximum number of passengers, denoted by CAP. If a bus already has CAP passengers on boarding, then the passengers waiting at the bus station have to continue to wait for the next coming bus. When taking into account the capacity constraint, the group-based optimization model cannot be applied directly because users within the same group may not take the same action, i.e., boarding the same bus or alighting at the same station. To leverage the reformulated model using binary variables, we treat each passenger as a single group and still set the temporal granularity to be one minute. In this way, each passenger is associated with a unique group index g and there are $G = P$ groups in total. The passengers and groups have one-for-one mapping and in the capacity constrained model, it is possible for multiple groups to arrive in the same minute. Let s_{gk} be a binary variable such that $s_{gk} = 1$ if the g-th group board the k-th bus. Then, we have

$$\sum_{k=1}^{K} s_{gk} = 1 \tag{21}$$

$$\sum_{k=1}^{K} s_{gk} \cdot u_{m'kn} = v_{gn} \tag{22}$$

where m' is an input variable referring to the waiting bus station of the g-the group. This constraint means if the

g-th group boards the k-th bus, the bus arrival time at station m' is the same with boarding time v_{gn}. The constraint is linear because for any three binary variables x_k, y_k and z_k, the equation $z_k = x_k y_k$ can be linearized by three inequalities: $z_k \leq x_k$, $z_k \leq y_k$ and $z_k \geq x_k + y_k - 1$. Consequently, constraint 22 can be transformed into a group of linear constraints.

Let b_{gm} and d_{gm} be two binary input variables, denoting the boarding and alighting bus station for the g-th group. If the g-th group boards (or alights) at the m-th station, we set GB_{gm} (or GA_{gm}) to be 1. Let c_{mk} be a decision variable to denote the number of passengers on the k-th bus when the bus arrives at station s_m. We have

$$c_{mk} \leq CAP \tag{23}$$

$$c_{1k} = 0 \tag{24}$$

$$c_{mk} = c_{(m-1)k} + \sum_{g=1}^{G} s_{gk} GB_{g(m-1)} - \sum_{g=1}^{G} GA_{gm} \tag{25}$$

In Constraint 25, $\sum_{g=1}^{G} s_{gk} b_{g(m-1)}$ is the number of groups boarding on the k-th bus from bus station s_{m-1} and $\sum_{g=1}^{G} s_{gk} d_{gm}$ is the number of groups alighting from the k-th bus at the station s_m.

C. Network-Based Optimization

The above models considers the optimization upon a single service route and the bus company can determine any number of buses allocated to that service. In practice, the total number of available buses is limited and it is crucial to examine the bus-network-level optimization such that the optimal number of buses allocated to each service can be determined in a data-driven way.

Given a bus network with a set of terminal stations, two terminals are connected if there exists a bus service starting from one and arriving at the other. Suppose in the initial state, there are I_t buses at terminal t and $\sum_t I_t$ is the total number of operating buses owned by the company. We use R_{tz} to denote the number of buses available at terminal t in the z-th time interval. Then, we have

$$R_{t1} = I_t \tag{26}$$

Suppose bus service s is assigned with a schedule plan with N_s buses, where N_s is the decision variable in our new optimization problem. We can adopt the proposed MIP model in Section IV-A and IV-B to find the optimal schedule for different instances of N_s. The output is the optimal departure schedule for a service s given N_s buses. Let A_s be the departure terminal and B_s be the arrival terminal of service s. We define a variable $D_{N_s i}$ referring to the optimal departure time from A_s for the i-th bus ($i \leq N_s$) allocated for service s and the expected arrival time at the counterpart terminal B_s is stored in variable $E_{N_s i}$. In addition, we are aware of the waiting time generated by the optimal schedule, which is denoted by $W(s, N_s)$. These three variables, pre-computed for different values of N_s, will then be fed into the network-based optimization model as input variables.

We are now ready to define the network-based optimization model as follows:

$$\min \sum_{s} W(s, N_s)$$

subject to :

$$R_{A_s D_{N_s i}} = R_{A_s (D_{N_s i}+1)} + 1 \tag{27}$$

$$R_{B_s (E_{N_s i}-1)} = R_{A_s E_{N_s i}} - 1 \tag{28}$$

$$R_{tz} \geq 0 \tag{29}$$

Constraints 27 and 28 are used to capture the temporal dynamism when buses move between terminals. When a bus moves from A_s (with departure time $D_{N_s i}$) to B_s (with estimated arrival time $E_{N_s i}$), the number of available buses in A_s decreases by 1 and that in B_s increases by 1. Constraint 29 is used to guarantee there always exists at least one available bus for each departure time in the schedule. Otherwise, the number of available buses would decrease to be a negative value for the next time interval, conflicting with the constraint.

V. Experiments

In this section, we first introduce the setup of the simulation experiments driven by real data and then evaluate the performance of our proposed model.

A. Setup of Simulation Experiments

In our case study, we randomly choose two bus services S_A and S_B that cover at least 25 bus stations for our case study. To show the effect of time-dependent traffic and demand, we select three time intervals in weekdays for our experiments: [8am, 9am] (morning peak), [2pm, 3pm] (off-peak) and [6pm,7pm] (evening peak).

The time-dependent traffic conditions for the two selected bus services in the simulation experiments are derived from all the relevant records in the bus dataset. For each edge in the bus route, we maintain an array λ of size T_t to capture the expected travel time from station s_i to its next station s_{i+1} in different periods of a day. Thus, in our simulation experiments, if a bus departures the terminal station s_1 at timestamp t_1. Its arrival time at station s_2 is estimated as $t_2 = t_1 + \lambda[\lceil \frac{T_t \cdot t_1}{86400} \rceil]$. Similarly, the arrival time at station s_{i+1} is $t_{i+1} = t_i + \lambda[\lceil \frac{T_t \cdot t_i}{86400} \rceil]$. Then, we have the arrival time at all the bus stations as long as we know the departure time.

To generate the real customer demand, we shift the boarding time of each passenger on the two selected bus services by an amount of waiting time randomly generated from the empirical probability distribution derived from historical data. Figure 3 shows the customer demand generated before and after the empirical probability distribution. The x-axis means the sequence of bus stations. Each dot represents one customer. Its two-dimensional coordinates (x,y) refer to the bus station id (spatial attribute) and arrival time (temporal attribute). For example, $x = 3$ refers to the customer demand at the third station of the bus route. We can see that, the passengers are originally aggregated into clusters as they board the arriving bus. The timestamp of each cluster denotes the boarding time of these passengers. After applying the empirical distribution of waiting time, the time-dependent customer demand becomes scattered and more realistic.

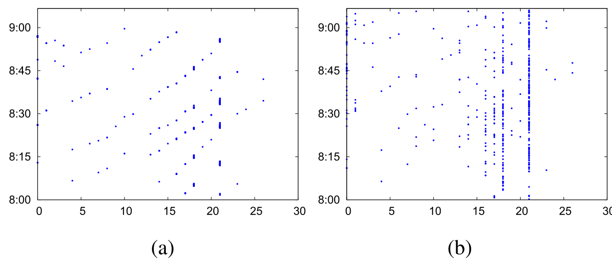


Fig. 3. Customer demand of service S_A. (a) Before Poisson (8am-9am). (b) After Poisson (8am-9am).

TABLE III

RUNNING TIME OF CPLEX MODELS IN PERIOD 8AM-9AM

Service	Model	K=3	K=4	K=5	K=6
S_A	Model-T-UN	0.06h	0.15h	0.36h	0.89h
	Model-D-UN	0.01h	0.04h	0.15h	0.55h
	Model-DT-UN	0.14h	0.48h	1.02h	5.76h
	Model-DT-CAP	0.46h	3.2h	10.7h	-
S_B	Model-T-UN	0.09h	0.21h	0.45h	1.17h
	Model-D-UN	0.02h	0.05h	0.17h	0.62h
	Model-DT-UN	0.19h	0.57h	1.32h	8.12h
	Model-DT-CAP	0.59h	5.28h	15.83h	-

B. Optimization Model Evaluation

We compare the optimization models with different settings. Specifically, the following models are of interest in our experimental study.

- Model-T-UN: the method assumes unlimited seats and only considers time-dependent traffic, without capturing the characteristic of time-dependent variation of customer demand. It assumes the demand at a bus station is fixed in different time windows. In other words, it assumes the arrival of passengers follows uniform distribution.
- Model-D-UN: the method only assumes unlimited seats and considers time-dependent customer demand. It assumes the average travel time between two consecutive bus stations is a constant.
- Model-DT-UN: the method assumes unlimited bus capacity and considers both time-dependent traffic and demand, as presented in Section IV-A.
- Model-DT-CAP: the method assumes limited bus capacity (60 seats in our setting) as well as the time-dependent traffic and demand.

In the naming, T is an abbreviation for traffic estimation, D for demand, UN for unlimited-seats assumption and CAP for capacity-constrained model. We implement the above models with IBM ILOG Cplex 12.6. The optimization experiments are conducted on a server with 128GB memory, 64KB L1 cache and 512KB L2 cache, running Centos 5.6. We evaluate the performance w.r.t. different K and report the running time and average waiting time.

Table III shows the running time of Cplex models for two bus services in the selected period of 8am-9am with varying K. We can see that as K increases from 3 to 6, the running time grows dramatically and it takes several hours for Model-DT-UN to find the optimal solution. Since the capacity constrained model Model-DT-CAP is the most difficult to solve as there are too many decision variables, its running time is not reported when $K = 6$. Model-T-UN and Model-D-UN take much less time because they only consider either time-dependent traffic or demand, which reduces the number of decision variables in Model-DT-CAP and facilitates the pruning when searching the optimal solution by Cplex. In our future work, we will examine how to further improve the efficiency with heuristic algorithms to support real-time optimization scenarios.

TABLE IV

AVERAGE WAITING TIME (IN SECONDS) OF DIFFERENT CPLEX MODELS DERIVED FROM THE SIMULATION ENVIRONMENT

Service	Model	K=3	K=4	K=5	K=6
S_A	Model-T-UN	615	476	334	276
	Model-D-UN	647	502	343	289
	Model-DT-UN	607	467	316	252
	Model-DT-CAP	598	443	305	-
S_B	Model-T-UN	653	515	378	301
	Model-D-UN	674	559	398	315
	Model-DT-UN	635	502	368	292
	Model-DT-CAP	623	487	341	-

TABLE V

DEPARTURE HEADWAY IN THE SELECTED TIME PERIODS

Service	8am - 9am	2pm - 3pm	6pm-7pm
S_A	10min - 21min	18min - 32min	12min - 22min
S_B	10min - 22min	20min - 35min	11min - 24min

The output of these optimization models is in the form of a scheduling timetable a specific bus service, which is then fed into our simulation environment to obtain the average waiting time for all the passengers taking the bus service. We use the simulation results to evaluate the model effectiveness and the average waiting time of the four models in the period of 8am-9am are reported in Table IV. We can see that accurate demand estimation plays a more important role than the time-dependent traffic and Model-T-UN beats Model-D-UN in all cases. Model-DT-UN achieves superior performance if both time-dependent factors are considered. Finally, when capacity constraint is considered, the waiting time can be further reduced but in expense of much more optimization time cost.

C. Comparing With Empirical Bus Scheduling

In the last experiment, we compare our proposed optimal model with simulated scheduling that complies with the guideline provided by the operating bus company. The departure headways of the two services are illustrated in Table V. To evaluate the performance within a selected time period $[t_{min}, t_{max}]$, we assume that the first bus departures at timestamp t_{min} and the last bus departures at t_{max}, with K being at least 2. Then, we start from t_{min} and sample the headway for the next bus following the guideline in Table V. When we obtain the next departure time $t_{min} + t_1$, where t_1 is the sampled headway, we continue the iteration until the last headway to t_{max} also follows the guideline. Note that these sampling procedures may lead to different values of K and we will report all the possible results.

Figure 4 shows the waiting time of Model-DT-UN compared to the synthetic simulation that follows the guideline in Table V. For the bus service S_A, when the time period is [7am, 8am], the minimum and maximum headway are required to be between $[10, 21]$ minutes. Hence, the number of allocated buses K could be from 4 to 7. When the time period is [2pm, 3pm], the headway interval is $[18, 32]$ and its possible number of allocated buses is from 3 to 4. For each selected

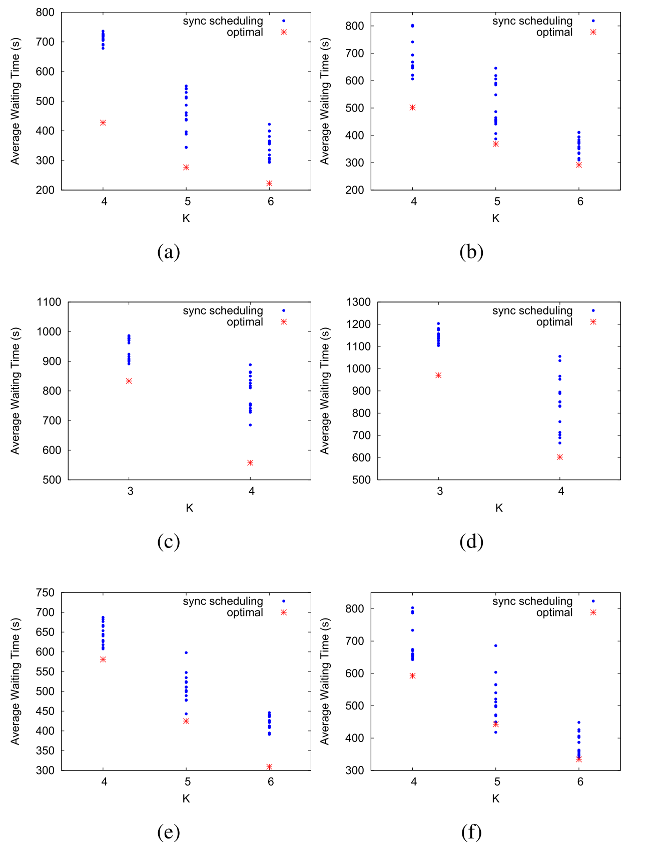


Fig. 4. Average waiting time for services S_A and S_B. (a) S_A (7am-8am). (b) S_B (7am-8am). (c) S_A (2pm-3pm). (d) S_B (2pm-3pm). (e) S_A (6pm-7pm). (f) S_B (6pm-7pm).

time period of a bus service, we randomly generate 30 solutions with the departure time following the guideline and select the top 15 results as our comparison method.

From the figures, we have three important observations. 1) As more buses put into operation for one service line, the headway can be reduced and thus the average waiting time is shortened. The effect is more remarkable when there are few buses. For example, in the morning peak-hour operating, the waiting time is reduced more than 2 minutes when the number of buses increases from 4 to 5. But when it continues to be increased to 6, the amount of gain in reducing waiting time is less significant. Thus, our optimization model has a side product to examine optimal bus resource allocation to different bus services. 2) The variance of the synthetic methods is quite large. Its current timetable scheduling is derived empirically. Without data-driven techniques, the scheduling may be arbitrarily bad. In contrast, our technique is data-driven and takes into account time-dependent traffic and demand. Thus, it provides more reliable scheduling services. 3) The scheduling generated by our optimal model is much more effective than the synthetic method. The reason is that it can foreseen the customer demand and traffic conditions. With the help of optimization models, it can reduce the waiting time systematically. Thus, it reduces the waiting time significantly, meaning that we can use fewer number of buses to provide similar average waiting time, which is an important factor in the service standard measurement.

TABLE VI
PERFORMANCE OF BUS NETWORK-BASE OPTIMIZATION

	$I_t = 3$	$I_t = 4$	$I_t = 5$	$I_t = 6$
Running time	1s	1s	3s	5s
Average waiting time	657s	498s	365s	278s

D. Evaluation of Bus Network-Based Optimization

To evaluate the performance of bus network-based optimization, we build a network with 20 bus terminals with 37 bus routes connecting them. We use the time-dependent traffic and demand in the period of 8am to 9am. We vary the number of available buses I_t in the initial state of each terminal from 3 to 6 and examine the running time of the network-based optimization model as well as the average waiting time. The optimization on each instance of N_s of a bus service is pre-computed and its running time is not counted in the experimental results in Table VI. We can see that as more buses available, it takes longer time to conduct the network-level optimization. However, this part is conducted rather efficiently because the only decision variable is N_s for each service. The average waiting time decreases significantly as more bus resources are available. The pattern is similar to that in a single-service optimization.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a data-driven model to optimize the bus scheduling system in a modern city in China. In our model, we took into account the time-dependent passenger demands and traffic patterns as distinguishing features from previous work. We leveraged hundreds of millions of bus transaction records to infer the time-dependent traffic and passenger demand. The experimental results on a simulated environment showed that our model can reduce the waiting time by a wide margin.

This work only sheds certain initial efforts on the bus scheduling optimization. From the perspective of customer demand estimation, even though smart-card is an important data source, it may incur potential bias issues. For instance, certain passengers may use coins rather than smart card. When integrating with other techniques, such as video analysis from bus CCTV or service enquiries directly sent from mobile apps, the estimation could be more accurate and robust. Each enquiry could specify a concrete demand at a timestamp from a certain bus station. On the one hand, by collecting and analyzing a sufficient amount of such enquiry data, we can get a more accurate picture about periodic user demand to complement the current optimization model proposed in this paper. On the other hand, a real-time scheduling system that can respond promptly to on-the-fly passenger demands indicated from mobile phones is an important direction of our future work.

REFERENCES

[1] *SBS Transit Annual Report 2014*, accessed on Sep. 17, 2015. [Online]. Available: http://www.sbstransit.com.sg/download/SBST_AnnualReport14_a.pdf

[2] *Quality of Service (QOS) Standards for Basic Bus Services*, accessed on Sep. 17, 2015. [Online]. Available: http://www.ptc.gov.sg/FactsAndFigures/QOS.htm#OPS

[3] *Fare Review: Bus, Train Fares to Increase by 3.2% This Year*, accessed on Sep. 17, 2015. [Online]. Available: http://www.todayonline.com/singapore/public-transport-fares-increase

[4] F. J. M. Salzborn, "Optimum bus scheduling," *Transp. Sci.*, vol. 6, no. 2, pp. 137–148, May 1972.

[5] A. Ceder, "Bus frequency determination using passenger count data," *Transp. Res. A, General*, vol. 18, nos. 5–6, pp. 439–453, Oct. 1984.

[6] F. J. M. Salzborn, "Scheduling bus systems with interchanges," *Transp. Sci.*, vol. 14, no. 3, pp. 211–231, Aug. 1980.

[7] I. Verbas and H. Mahmassani, "Optimal allocation of service frequencies over transit network routes and time periods: Formulation, solution, and implementation using bus route patterns," *Transp. Res. Rec.*, vol. 2334, pp. 50–59, Dec. 2013.

[8] H. Martínez, A. Mauttone, and M. E. Urquhart, "Frequency optimization in public transportation systems: Formulation and metaheuristic approach," *Eur. J. Oper. Res.*, vol. 236, no. 1, pp. 27–36, 2014.

[9] A. Ceder and L. Philibert, "Transit timetables resulting in even maximum load on individual vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2605–2614, Dec. 2014.

[10] D. J. Sun, Y. Xu, and Z.-R. Peng, "Timetable optimization for single bus line based on hybrid vehicle size model," *J. Traffic Transp. Eng.*, vol. 2, no. 3, pp. 179–186, Jun. 2015.

[11] S. Hassold and A. Ceder, "Multiobjective approach to creating bus timetables with multiple vehicle types," *Transp. Res. Rec.*, vol. 2276, pp. 56–62, Dec. 2012.

[12] O. J. Ibarra-Rojas, R. Giesen, and Y. A. Rios-Solis, "An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks," *Transp. Res. B, Methodol.*, vol. 70, pp. 35–46, Dec. 2014.

[13] A. Ceder, B. Golany, and O. Tal, "Creating bus timetables with maximal synchronization," *Transp. Res. A, Policy Pract.*, vol. 35, no. 10, pp. 913–928, Dec. 2001.

[14] O. J. Ibarra-Rojas and Y. A. Rios-Solis, "Synchronization of bus timetabling," *Transp. Res. B, Methodol.*, vol. 46, no. 5, pp. 599–614, 2012.

[15] M. Salicrú, C. Fleurent, and J. Armengol, "Timetable-based operation in urban transport: Run-time optimisation and improvements in the operating process," *Transp. Res. A, Policy Pract.*, vol. 45, no. 8, pp. 721–740, 2011.

[16] D. Yang, D. Zhang, K. L. Tan, J. Cao, and F. L. Mouël, "CANDS: Continuous optimal navigation via distributed stream processing," *Proc. Endowment*, vol. 8, no. 2, pp. 137–148, Oct. 2014.

[17] Q. Liu, W. Cai, J. Shen, Z. Fu, X. Liu, and N. Linge, "A speculative approach to spatial-temporal efficiency with multi-objective optimization in a heterogeneous cloud environment," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 4002–4012, 2016.

[18] X. Zhou and M. Zhong, "Bicriteria train scheduling for high-speed passenger railroad planning applications," *Eur. J. Oper. Res.*, vol. 167, no. 3, pp. 752–771, 2005.

[19] R. Chevrier, P. Pellegrini, and J. Rodriguez, "Energy saving in railway timetabling: A bi-objective evolutionary approach for computing alternative running times," *Transp. Res. C, Emerg. Technol.*, vol. 37, pp. 20–41, Dec. 2013.

[20] O. J. Ibarra-Rojas, F. Delgado, R. Giesen, and J. C. Muñoz, "Planning, operation, and control of bus transport systems: A literature review," *Transp. Res. B, Methodol.*, vol. 77, pp. 38–75, Jul. 2015.

[21] B. Barabino, M. D. Francesco, and S. Mozzoni, "An offline framework for handling automatic passenger counting raw data," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2443–2456, Dec. 2014.

[22] J. E. Fernández L., J. de Cea Ch., and R. H. Malbran, "Demand responsive urban public transport system design: Methodology and application," *Transp. Res. A, Policy Pract.*, vol. 42, no. 7, pp. 951–972, 2008.

[23] A. S. Mohaymany and S. M. M. Amiripour, "Creating bus timetables under stochastic demand," *Int. J. Ind. Eng.*, vol. 20, no. 3, pp. 83–91, 2009.

[24] L. Sun, J. G. Jin, D.-H. Lee, K. W. Axhausen, and A. Erath, "Demand-driven timetable design for metro services," *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 284–299, Sep. 2014.

[25] H. Niu, X. Zhou, and R. Gao, "Train scheduling for minimizing passenger waiting time with time-dependent demand and skip-stop patterns: Nonlinear integer programming models with linear constraints," *Transp. Res. B, Methodol.*, vol. 76, pp. 117–135, Jun. 2015.

[26] M.-P. Pelletier, M. Trépanier, and C. Morency, "Smart card data use in public transit: A literature review," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 4, pp. 557–568, 2011.

[27] Y. Wang, D. Zhang, Q. Liu, F. Shen, and L. H. Lee, "Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions," *Transp. Res. E, Logistics Transp. Rev.*, vol. 93, pp. 279–293, Sep. 2016.

[28] D. Zhang, C. Y. Chan, and K. Tan, "An efficient publish/subscribe index for e-commerce databases," *Proc. VLDB Endowment*, vol. 7, no. 8, pp. 613–624, 2014.

[29] L. Guo, D. Zhang, G. Li, K. L. Tan, and Z. Bao, "Location-aware pub/sub system: When continuous moving queries meet dynamic event streams," in *Proc. SIGMOD*, Jun. 2015, pp. 843–857.

[30] P. Jonsson and C. Wohlin, "An evaluation of k-nearest neighbour imputation using Likert data," in *Proc. 10th Int. Symp. Softw. Metrics*, Sep. 2004, pp. 108–118.

[31] *Empirical Probability*, accessed on Aug. 20, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Empirical_probability

[32] Y. Zhang, X. Sun, and B. Wang, "Efficient algorithm for k-barrier coverage based on integer linear programming," *China Commun.*, vol. 13, no. 7, pp. 16–23, Jul. 2016.

[33] B. Barabino, S. Salis, and B. Useli, "Fare evasion in *proof-of-payment* transit systems: Deriving the optimum inspection level," *Transp. Res. B, Methodol.*, vol. 70, pp. 1–17, Dec. 2014.

Yuan Wang received the Ph.D. degree in operations research, with a specialization in maritime transportation optimization and simulation, from the National University of Singapore. She is a Research Fellow with the Department of Industrial and Systems Engineering, National University of Singapore. Her research interests include mathematical modeling, complex system simulation, and optimization heuristics.

Dongxiang Zhang received the B.Sc. degree from Fudan University, China, in 2006 and the Ph.D. degree from National University of Singapore in 2012. He was a Research Fellow with the NeXT Research Center, Singapore, from 2012 to 2014, where he was a Senior Research Fellow in 2015. He is currently a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include cloud computing and big data analytic.

Lu Hu received the B.Sc. and Ph.D. degrees from Southwest Jiaotong University, China, in 2009 and 2014, respectively. He is a Research Fellow with the Department of Industrial and Systems Engineering, National University of Singapore. His research interests include urban logistics, vehicle sharing, urban rail transit, pedestrian traffic, queueing theory, and simulation optimization.

Yang Yang received the bachelor's degree from Jilin University in 2006, the master's degree from Peking University in 2009, and the Ph.D. degree from The University of Queensland, Australia, in 2012. He was a Research Fellow under the supervision of Prof. T.-S. Chua, with the National University of Singapore, from 2012 to 2014. He is currently with University of Electronic Science and Technology of China.

Loo Hay Lee (M'97–SM'04) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1992, and the M.S. and Ph.D. degrees from Harvard University, Cambridge, MA, USA, in 1994 and 1997, respectively. He is an Associate Professor with the Department of Industrial and Systems Engineering, National University of Singapore, Singapore. His research interests include simulation-based optimization, maritime logistics, and port operations and supply chain modeling.