

Bikeshare Pool Sizing for Bike-And-Ride Multimodal Transit

Guoming Tang, *Member, IEEE*, Srinivasan Keshav, *Senior Member, IEEE*, Lukasz Golab, *Member, IEEE*,
and Kui Wu, *Senior Member, IEEE*

Abstract—In shared bike-and-ride transit systems, commuters use shared bicycles for last-mile transport between transit stations and home, and between transit stations and work locations. This requires pools of bicycles to be located near each transit stop where commuters can drop off and pick up shared bikes. We study the optimal sizing of such bicycle pools. While various problems related to vehicle pool sizing have been studied before, to the best of our knowledge this is the first work that considers a multimodal transportation system with a regularly-scheduled public transportation backbone and shared bicycles for the first and last mile. We present two solutions that guarantee bicycle availability with high probability, and we empirically verify their effectiveness using Monte Carlo simulations. Compared to a baseline solution, our techniques reduce the size of the bikeshare pool at the public transit station by 39 to 75 percent in the tested scenarios.

Index Terms—Bikeshare pool sizing; multimodal transit.

I. INTRODUCTION

Due to increasing traffic congestion [1] and motivated by cost savings and environmental protection, many commuters have come to rely on public transit. If transit stations are outside the walkable range or have limited vehicle parking, then cycling is a convenient option for the first and last mile [2], [3], [4], [5]. We refer to this arrangement as bike-and-ride multimodal transit.

One problem with bike-and-ride is that buses and trains have limited on-board space for bicycles, preventing commuters from taking their bicycles to the final destination [6]. Bicycle sharing (bikeshare) programs, with bicycle stands located near public transit stations, can provide a solution to this problem [7], [8]. As of 2016, approximately 1000 cities around the world have bikeshare programs [9], some of which also include commuter-friendly electric bicycles [10].

A critical aspect of any successful bikeshare system, especially for commuters who cannot afford to be late for work, is bicycle availability. However, over-provisioning is not ideal at best and infeasible at worst, due to high cost and limited bicycle docking space in dense neighbourhoods. This motivates the need for bikeshare pool sizing techniques, which we study in this work.

G. Tang is with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, Hunan 410073, China (e-mail: gmtang@nudt.edu.cn).

S. Keshav and L. Golab are with the Cheriton School of Computer Science and the Department of Management Sciences, respectively, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: keshav@uwaterloo.ca; lgolab@uwaterloo.ca).

K. Wu is with the Department of Computer Science, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: wkui@uvic.ca).

Consider the following situation: a bikeshare customer routinely travels from home to a destination such as their workplace, and back home. That is, each customer cycles from home to their *origin* station using a shared bicycle, returns the bicycle to the pool located there, takes public transport to their *destination* station, rents another bicycle from a pool at this location, and cycles from the destination station to their final destination, e.g., workplace. The customer keeps ownership of the bicycle during the working day, then uses the same bicycle to return to the destination station, drops it off there, takes public transport to their home station, and picks up a different bicycle there to travel home, retaining ownership of the bike overnight. A naive solution is to allocate two bicycles per customer: one at their origin station, which will be taken home at the end of the workday (or returned to a bikeshare stand at the customer’s home if one exists), and one at their destination station, which will be parked at the workplace throughout the workday (or returned to a bikeshare stand at the customer’s workplace if one exists). While this solution is easy to implement, it is expensive. We present two probabilistic techniques that guarantee bicycle availability, with high probability, but using a smaller bicycle pool: i) a transient-state analysis based on the difference of random variables and ii) a steady-state analysis based on the Engset model.

There is prior work on optimal facility location and pool sizing for bikeshare systems [11], [12], [13], [14]. However, to the best of our knowledge, this paper is the first to study the bikeshare pool sizing problem in the context of bike-and-ride multimodal transit. What makes our problem challenging is the coupling of the non-stationary process of bicycle rentals and returns with the public transportation schedule. We focus on bike-and-ride multimodal transit systems with a public transportation backbone. The purpose of such systems is to serve commuters who need bikes both from their homes to local public stations and from remote public stations to their work places, and thus help them solve the so-called “first-mile” and “last-mile” problems. Our contributions are as follows:

- We formally define the bikeshare pool sizing problem for bike-and-ride multimodal transit systems with a public transportation backbone and shared bicycle stands located at public transit stations.
- We propose two bikeshare pool sizing techniques which guarantee bicycle availability with high probability. Given a user-supplied threshold ϵ , our techniques guarantee a request blocking probability of at most ϵ .

- We validate our methods using Monte Carlo simulations under various realistic scenarios. Compared to the aforementioned baseline solution which requires two bicycles per customer, our solutions require 1.25 ~ 1.62 bicycles per customer in the tested scenarios. The reduction of bicycles per customer results in 39% ~ 75% smaller bicycle pools at public transit stations.

The remainder of this paper is organized as follows. We discuss related work in Section II, and formally state our problem and the assumptions in Section III. We model our customer arrival process in Section III, followed by our two bikeshare pool sizing techniques in Section V. We discuss simulation results in Section VI and conclude the paper in Section VIII.

II. RELATED WORK

Bike-and-ride multimodal transit has been studied in several works. Rietveld [5] reported that in the Netherlands, 23% of train passengers arrived to and from train stations on bicycle. Krizek & Stonebraker [6] pointed out the limited space for bicycles aboard transit vehicles, and explored alternative solutions for integrating cycling and transit such as bikeshares and bicycle parking at transit stops. Similarly, Pucher and Buehler [4] argued for more bicycle-carrying capacity on trains and more secure and sheltered bicycle parking at railway stations.

Bikeshare pool sizing has been studied but usually not in the context of multimodal transit [11], [12], [13], [14]. One exception is the work by Chen et al. [15], which considered bicycle docking stations in residential neighbourhoods, near workplaces, and at a single metro station which can move public bicycles parked there to other docking stations. Given a set of origins and destinations, they focused on bikeshare pool sizing at residential docking stations and bicycle parking sizing at other stations, without taking the metro schedule into account. In contrast, we focus on bikeshare pool sizing at public transport stations.

From a technical standpoint, our work is closest to that of Carpenter et al. [16], who focused on sizing vehicle rental fleets with a finite population of subscribers. Two main differences are: 1) Carpenter et al. allowed only one pool whereas we consider multiple pools, one at each public transit stop, and 2) Carpenter et al. did not consider multimodal transit, whereas we incorporate public transit schedules into our pool sizing solutions.

Finally, we mention recent work on predicting the number of available bicycles in a bikeshare based on the time of day and day of the week [17]. Again, multimodal transit was not considered.

III. PROBLEM STATEMENT AND ASSUMPTIONS

In this section, we define our problem and explain our assumptions. Table I explains the symbols used in the remainder of this paper. To simplify the presentation, we refer to the public transport line as *train line* or *railway*, but our solution also applies to other regularly-scheduled modes such as buses or subways.

TABLE I
SUMMARY OF SYMBOLS

Symbol	Meaning
N	Number of bikeshare stations
ϵ	Desired blocking probability threshold
$B_n(t)$	Number of available bikes at station n at time t
t_0/t_e	Start/end time of a daily cycle
t_r	Arrival time of train r
X_r	Inter-arrival time between train $r - 1$ and r
$\Delta_{i,j}$	Travel time by train from station i to j
S	Population vector
H	Population partition matrix
V/Θ	Customer departure/return (time) matrix
$\mu_n(t)$	Bike return rate at station n at time t
$\lambda_n(t)$	Bike demand rate at station n at time t
$\hat{S}_n(t)$	Max. #bikes returned to station n after time t
$\check{S}_n(t)$	Max. #bikes demanded at station n after time t
$C_n(t)$	Cumulative difference between bike returns and demands at station n at time t
U_n	Number of bike returns at station n
D_n	Number of bikes rented at station n
f_n	Difference between U_n and D_n at station n
$Y_n(t)$	Number of bikes rented at station n at time t
$\zeta_n(t)$	Satisfaction degree at station n at time t
bpc	Performance metric of bikes per customer
bsr	Performance metric of bike saving ratio
aba	Performance metric of actual bike availability

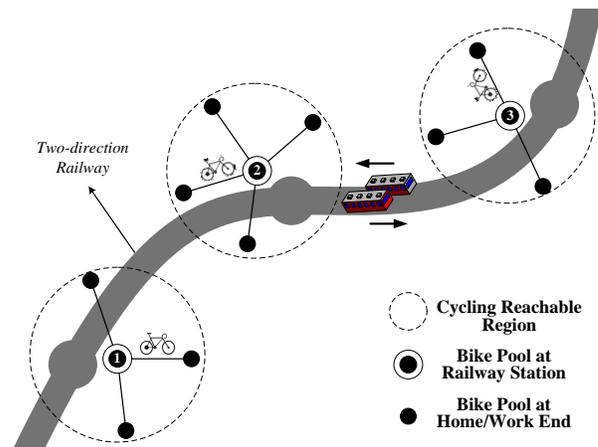


Fig. 1. Illustration of bikeshare stations located at three train stations.

A. Overview

Figure 1 illustrates a bike-and-ride transport system with $N = 3$ train stations, providing bike-to-train and train-to-bike service to a finite set of customers. We assume that each customer follows a daily routine: at some point, the customer cycles from home to their *local station*, returns a bike there, takes the train to a *remote station*, and borrows a bike from the remote station to cycle to their final destination. Later in the day, we assume that each customer returns home by following these steps in reverse. The bikeshare thus provides a feeder system to public transport within the cycling reachable region of each station (see, e.g., [5] for estimating cycling reachable regions).

As shown in Figure 1, we assume that there is a bikeshare pool at every train station, and there may be local bikeshare

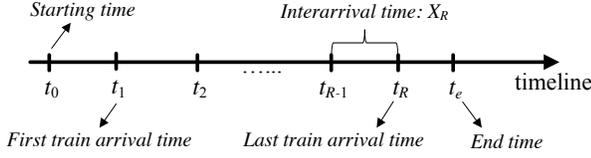


Fig. 2. Example of a train schedule at one station.

pools near customers' homes and workplaces (or customers may park bikes at home or at work). Previous work on bikeshare pool sizing has focused on pools near homes or workplaces [11], [13], [14], [15]. In contrast, we focus on the new problem of bike pool sizing at railway stations, which is a harder problem because train schedules must be taken into account. From now on, unless otherwise specified, we use the terms bike pool and bike station to refer to those at railway stations.

Let $B_n(t)$ be the number of available bikes at station n at time t . This number may fluctuate throughout the day as customers borrow and return bikes. Our goal is to ensure that there are always enough bikes for customers getting off the train. To do so, for each station n , we will calculate a minimum value for $B_n(0)$, the number of available bikes at the beginning of a day (before the arrival of the first train), that guarantees bike availability to customers throughout the day with a small blocking probability.

After calculating the initial $B_n(0)$ values, we need to make sure that this number of bikes is available at each station at the beginning of each day. A simple solution, *static dispatching*, is to move bikes among stations at the beginning or end of each day [18], [19]. With this solution, to make sure there are enough parking spaces for all returned bikes at all stations at any time, we set the number of bike parking spaces (or bike docks) at station n to $\max_t B_n(t)$. *Dynamic dispatching* can reduce the required number of docks by repositioning bikes throughout the day [20], [21]. In this paper, we focus on bike pool sizing rather than dispatching, but we will revisit the issue of dynamic dispatching in Section V-D.

B. Train Schedules

For any train station, the daily train schedule is a deterministic arrival (and departure) process $\psi = \{t_r : r = 1, 2, \dots, R\}$, in which t_r represents the arrival time of train r and R is the total number of trains. We denote the inter-arrival times of adjacent trains as $\{X_r : r = 1, 2, \dots, R\}$, where $X_r = t_r - t_{r-1}$. Figure 2 gives an example of a train schedule, with t_0 and t_e being the start and end times of a daily cycle, respectively.

We make two simplifying assumptions about train arrivals and departures. First, we ignore the relatively short train stopping times to load and unload passengers. Second, we assume that the train schedule is "symmetric". This means that whenever a train from station $i - 1$ arrives at station i , another train from station $i + 1$ arrives at the same time from the opposite direction. In practice, such arrivals may

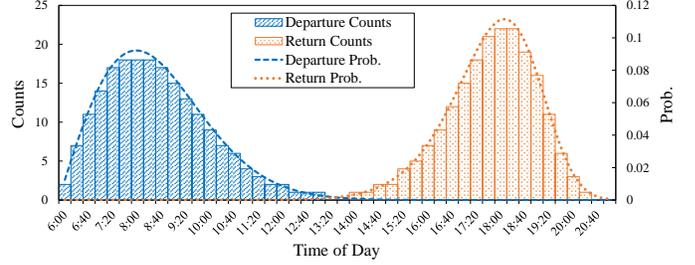


Fig. 3. An example of the distributions of the random variables corresponding to departure and return times at a particular station. This illustrates our approach to obtaining the departure and return time distributions from the collected data.

be several minutes apart, but, again, this time difference is relatively small.

Finally, let Δ be a travel time matrix, where $\Delta_{i,j} = \Delta_{j,i}$, $1 \leq i, j \leq N$, is the travel time of a train between stations i and j .

C. Customer Traffic Matrices

Let S be a *population vector* whose n th coordinate, S_n , denotes the number of customers whose local station is n .

$$S := [S_1, S_2, \dots, S_N]^T. \quad (1)$$

The population vector can be determined, e.g., by examining customer addresses. For customers with multiple nearby stations, we choose the closest one.

Next, we define a *population partition matrix* H

$$H := \begin{bmatrix} H_{1,1} & H_{1,2} & \dots & H_{1,N} \\ H_{2,1} & H_{2,2} & \dots & H_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ H_{N,1} & H_{N,2} & \dots & H_{N,N} \end{bmatrix} \quad (2)$$

in which $H_{i,j}$ is the fraction of population travelling from local station i to remote station j , where $0 \leq H_{i,j} \leq 1$ and $\sum_{j=1}^N H_{i,j} = 1$. This information can be estimated from customer surveys or train station data (such as the number of passengers entering and exiting a station)¹.

Using S and H , we can compute customer traffic flows: the number of customers with local station i and remote station j is $S_i H_{i,j}$.

In addition to knowing how many customers use different stations, we need to know approximately *when* they travel. Although most studies of transit behaviour are based on trip-level information, this data is unavailable for bikeshare commuters. Hence, we are forced to make an approximation, based on expected arrival and departure times, rather than trip statistics. Accordingly, let V be a *departure distribution matrix*

$$V := \begin{bmatrix} v_{1,1}(t) & v_{1,2}(t) & \dots & v_{1,N}(t) \\ v_{2,1}(t) & v_{2,2}(t) & \dots & v_{2,N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ v_{N,1}(t) & v_{N,2}(t) & \dots & v_{N,N}(t) \end{bmatrix} \quad (3)$$

¹Note that this modeling approach allows us to easily account for more complex commute patterns, since their only impact is to modify the values in this matrix. For simplicity, we work with a simple home/work commute model in this paper.

where $v_{i,j}(t), t_0 \leq t \leq t_e$, represents the probability density function (PDF) of the departure time distribution at local station i for customers traveling to remote station j . Similarly, let Θ be a *return distribution matrix*

$$\Theta := \begin{bmatrix} \theta_{1,1}(t) & \theta_{1,2}(t) & \cdots & \theta_{1,N}(t) \\ \theta_{2,1}(t) & \theta_{2,2}(t) & \cdots & \theta_{2,N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{N,1}(t) & \theta_{N,2}(t) & \cdots & \theta_{N,N}(t) \end{bmatrix} \quad (4)$$

where $\theta_{i,j}(t), t_0 \leq t \leq t_e$, denotes the PDF of the return time distribution at remote station i for customers returning to their local station j . Again, these matrices can be estimated from customer surveys or train station data.

Example 1: Figure 3 shows an example of a departure and return distribution between two stations, where most customers depart the local station in the morning (and take the train to work) and return to the remote station in the afternoon (and take the train home). Note that the distributions we can learn from surveys or historical data are discrete, and need to be converted to continuous PDFs for our analysis, as shown in the figure.

The matrices defined in this section allow us to compute two important pieces of information. Consider the time period between the departure of train $r-1$ at time t_{r-1} and the arrival of train r at time t_r . For any train station n :

- The expected number of customers arriving by bike and dropping it off (i.e., whose local station is n) is

$$\sum_{k=1}^N S_n H_{n,k} \int_{t_{r-1}}^{t_r} v_{n,k}(t) dt = S_n \sum_{k=1}^N \int_{t_{r-1}}^{t_r} v_{n,k}(t) dt. \quad (5)$$

- The expected number of customers arriving by train who will need a bike (i.e., whose remote station is n) is

$$\sum_{k=1}^N S_k H_{k,n} \int_{t_{r-1}}^{t_r} \theta_{n,k}(t) dt. \quad (6)$$

D. Problem Statement

The number of available bikes at each station changes throughout the day as customers arrive and depart by train. A customer is said to be *blocked* if he or she finds the bike pool empty upon arrival; if so, we assume that the customer will leave without waiting. Given a train schedule and the traffic matrices defined earlier in this section, we want to minimize the initial bike pool size at each station, $B_n(0), 1 \leq n \leq N$, such that the *blocking probability* at each station is less than a user-specified threshold ϵ .

Remark 1: Equations (5) and (6) cannot be used directly to size the bike pool at the given station. These are **expected** values which may be different from **actual** values. For example, given that the expected number of arrivals during some short period of time is 9.5, it is *not* correct to conclude that a pool of ten bikes at this time is sufficient. Instead, we will apply probabilistic analysis to calculate blocking probabilities.

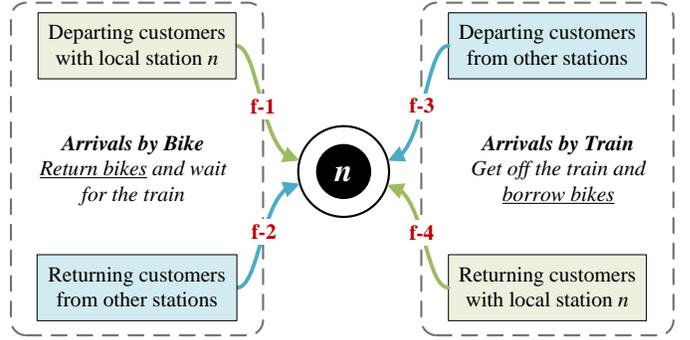


Fig. 4. Customer arrival flows at an arbitrary station.

IV. ARRIVAL ANALYSIS

To analyze how bikes are borrowed and returned at a particular station n , we consider four customer flows, as illustrated in Figure 4. Flows **f-1** and **f-2** correspond to customers arriving at station n by bike and returning their bikes, while flows **f-3** and **f-4** correspond to customers arriving at station n by train and borrowing bikes. More precisely:

- **f-1:** customers arriving at station n from home (i.e., n is their local station) and taking a train to their final destination, with departure time distributions $v_{n,j}, 1 \leq j \leq N$, given by equation (3).
- **f-2:** customers arriving at station n from their final destination (i.e., n is their remote station) and taking a train to return home, with return time distributions $\theta_{n,i}, 1 \leq i \leq N$, given by equation (4).
- **f-3:** customers arriving at station n from another station and borrowing a bike to get to their final destination (i.e., n is their remote station), with departing time distributions $v_{i,n}, 1 \leq i \leq N$, given by equation (3),
- **f-4:** customers arriving at station n from another station and borrowing a bike to get home (i.e., n is their local station), with returning time distributions $\theta_{j,n}, 1 \leq j \leq N$, given by equation (4).

A. Expected Short-Term Bike Rentals and Returns

To capture the number of bikes returned at station n during any short time interval $[t-\delta, t]$, we model the flows **f-1** and **f-2** as Poisson point processes, which are widely used in customer-server queue systems. In our context, the processes may not be stationary as the arrival rate changes over time (recall Figure 3 with morning/evening arrival peaks). Nevertheless, for a short time period, the arrival rate can be regarded as constant and thus a Poisson process is applicable [16].

Let $\mu_n^{(f-1)}(t)$ and $\mu_n^{(f-2)}(t)$ be the rates of the arrival flows **f-1** and **f-2**, respectively:

$$\mu_n^{(f-1)}(t) = S_n \sum_{k=1}^N \int_{t-\delta}^t v_{n,k}(t) dt, \quad (7)$$

$$\mu_n^{(f-2)}(t) = \sum_{k=1}^N S_k H_{k,n} \int_{t-\delta}^t \theta_{n,k}(t) dt. \quad (8)$$

Corollary 1: Given Equations (7) and (8), the aggregated arrivals by bike (i.e., bike returns) during time interval $[t - \delta, t]$ at station n can be modelled by another Poisson process (as the sum of independent Poisson processes is Poisson as well) with the following rate:

$$\mu_n(t) = \mu_n^{(f-1)}(t) + \mu_n^{(f-2)}(t).$$

To capture the demand for bikes at station n during $[t - \delta, t]$, we model the flows **f-3** and **f-4** as aggregated results of independent Poisson processes originating at other stations. Moreover, the processes are *delayed* by the time it takes the train to travel to station n , defined in the travel time matrix Δ (recall Section III-B).

Let $\mu_n^{(f-3)}(t)$ and $\mu_n^{(f-4)}(t)$ be the rates of the arrival flows **f-3** and **f-4**, respectively:

$$\lambda_n^{(f-3)}(t) = \sum_{k=1}^N S_k H_{k,n} \int_{t-\delta-\Delta_{k,n}}^{t-\Delta_{k,n}} v_{k,n}(t) dt, \quad (9)$$

$$\lambda_n^{(f-4)}(t) = S_n \sum_{k=1}^N \int_{t-\delta-\Delta_{k,n}}^{t-\Delta_{k,n}} \theta_{k,n}(t) dt. \quad (10)$$

Corollary 2: Given Equations (9) and (10), the aggregated arrivals by train (or demand for bikes) during time interval $[t - \delta, t]$ at station n can be modelled by a Poisson process with the following rate:

$$\lambda_n(t) = \lambda_n^{(f-3)}(t) + \lambda_n^{(f-4)}(t).$$

B. Expected Bike Rentals and Returns in the Future

Next, at any station n , we compute the expected number of bikes returned between some time t_x and the end of the daily cycle (t_e). For flows **f-1** and **f-2**, respectively, we get:

$$\hat{S}_n^{(f-1)}(t_x) = S_n \sum_{k=1}^N \int_{t_x}^{t_e} v_{n,k}(t) dt, \quad (11)$$

$$\hat{S}_n^{(f-2)}(t_x) = \sum_{k=1}^N S_k H_{k,n} \int_{t_x}^{t_e} \theta_{n,k}(t) dt. \quad (12)$$

Corollary 3: Given Equations (11) and (12), during the time interval $[t_x, t_e]$, the expected maximum number of bikes returned to station n is:

$$\hat{S}_n(t_x) = \hat{S}_n^{(f-1)}(t_x) + \hat{S}_n^{(f-2)}(t_x).$$

Similarly, for the expected number of bikes borrowed from station n during the time interval $[t_x, t_e]$, we get the following sums, corresponding to flows **f-3** and **f-4**, respectively:

$$\hat{S}_n^{(f-3)}(t_x) = \sum_{k=1}^N S_k H_{k,n} \int_{t_x}^{t_e} v_{k,n}(t) dt, \quad (13)$$

$$\hat{S}_n^{(f-4)}(t_x) = S_n \sum_{k=1}^N \int_{t_x}^{t_e} \theta_{n,k}(t) dt. \quad (14)$$

Corollary 4: Given Equations (13) and (14), during the time interval $[t_x, t_e]$, the expected maximum number of bikes borrowed at station n is:

$$\check{S}_n(t_x) = \hat{S}_n^{(f-3)}(t_x) + \hat{S}_n^{(f-4)}(t_x).$$

C. Expected Bike Rentals and Returns in the Past

Finally, for any time t_x , we compute the expected number of bikes remaining at any station n . To do this, we need to know how many local customers have departed their home station n and not yet returned, and how many remote customers have arrived at station n but not yet returned home.

First, the expected number of local customers who have not returned to station n by time t_x is:

$$C_n^{(\text{out})} = S_n \sum_{k=1}^N \int_{t_0}^{t_x} (v_{n,k}(t) - \theta_{k,n}(t)) dt \quad (15)$$

Then, the expected number of remote customers who have not yet arrived at station n to return home by time t_x is:

$$C_n^{(\text{in})} = \sum_{k=1}^N S_k H_{k,n} \int_{t_0}^{t_x} (v_{k,n}(t) - \theta_{n,k}(t)) dt. \quad (16)$$

Corollary 5: At any time t_x , given Equations (15) and (16), the (expected) cumulative difference between bikes returned and borrowed at station n is:

$$C_n(t_x) = C_n^{(\text{out})} - C_n^{(\text{in})} (1 - \bar{\epsilon}(t_x))$$

where $\bar{\epsilon}(t_x)$ is the expected blocking probability of bike requests till time t_x . The value of $C_n(t_x)$ is negative when there is a shortage of available bikes (more bikes expected to be borrowed than returned, cumulatively, by time t_x).

V. BIKE POOL SIZING

In this section, we develop two methods for bike pool sizing at individual stations: i) a transient-state approach of analyzing differences of random variables (RVs) in Section V-A, and ii) a steady-state approach based on the Engset model in Section V-B. We then discuss how to implement these methods efficiently (Section V-C) and provide a brief comparison (Section V-D).

A. Transient-State Analysis with Difference of RVs

In the transient-state approach, we divide the daily cycle (e.g., 6:00am - 12:00am) into short epochs according to the train schedule $\{t_r : i = 1, 2, \dots, R\}$, as illustrated in Figure 2.

1) *RVs for Bike Return/Demand:* Consider the time interval $(t_{r-1}, t_r]$ between the arrivals of two consecutive trains, $r - 1$ and r , for $1 \leq r \leq R$. For any station n , we define the following random variable (RV):

$$U_n(t_r) = \text{number of bikes returned to station } n \text{ during } (t_{r-1}, t_r].$$

According to Corollary 1, $U_n(t_r)$ is a Poisson process with rate $\mu_n(t_r)$. Therefore, its PDF is:

$$P_{U_n(t_r)}(x) = \frac{(\mu_n(t_r))^x e^{-\mu_n(t_r)}}{x!}, \quad (17)$$

where $0 \leq x \leq \hat{S}_n(t_r)$ with $\hat{S}_n(t_r)$ computed according to Corollary 3. Thus, $\hat{S}_n(t_r)$ gives the (expected) maximum

number of customers who may return bikes to station n during the time interval $(t_{r-1}, t_r]$.

Similarly, for customers arriving by train and renting bikes at station n , we define another RV:

$D_n(t_r)$ = number of bikes demanded at station n at time t_r .

According to Corollary 2, $D_n(t_r)$ is a delayed Poisson process with rate $\lambda_n(t_r)$. Thus, its PDF is:

$$P_{D_n(t_r)}(x) = \frac{(\lambda_n(t_r))^x e^{-\lambda_n(t_r)}}{x!}, \quad (18)$$

where $0 \leq x \leq \check{S}_n(t_r)$ with $\check{S}_n(t_r)$ computed according to Corollary 4. Thus, $\check{S}_n(t_r)$ gives the (expected) maximum number of customers that may borrow bikes from station n upon arrival of train r . We assume that $U_n(t_r)$ and $D_n(t_r)$ are independent.

2) *Transient-State Blocking Probability*: We define the difference between $U_n(t_r)$ and $D_n(t_r)$ as:

$$f_n(t_r) = U_n(t_r) - D_n(t_r).$$

Consider the situation where $f_n(t_r) \leq 0$. Then, according to the difference distribution of two discrete RVs, its PDF is:

$$P_{f_n(t_r)}(x) = \sum_{k=0}^{\check{S}_n(t_r)} P_{U_n(t_r)D_n(t_r)}(k, k-x) \quad (19a)$$

$$= \sum_{k=0}^{\check{S}_n(t_r)} P_{U_n(t_r)}(k) P_{D_n(t_r)}(k-x), \quad (19b)$$

where $P_{U_n(t_r)D_n(t_r)}(k, k-x)$ is the joint probability of $U_n(t_r)$ and $D_n(t_r)$, and $-\check{S}_n(t_r) \leq x \leq 0$. Recall that x denotes the value of random variable $f_n(t_r)$ and thus it is negative when there is a greater demand for bikes than those returned.

Hence, upon the arrival of train r at station n , given that $B_n(t_{r-1})$ bikes were available at the station beforehand, the probability that a random customer from other stations finds an empty bike pool (defined as event \emptyset) is given by:

$$P(\emptyset | B_n(t_{r-1})) = \begin{cases} 0, & \text{if } B_n(t_{r-1}) \geq \check{S}_n(t_r), \\ \sum_{x=-\check{S}_n(t_r)}^{-B_n(t_{r-1})} P_{f_n(t_r)}(x), & \text{otherwise.} \end{cases} \quad (20)$$

3) *Pool Size Minimization*: The relationship between $B_n(t_{r-1})$ and $B_n(0)$ can be expressed as:

$$B_n(t_{r-1}) = \begin{cases} 0, & \text{if } B_n(0) + C_n(t_{r-1}) \leq 0, \\ B_n(0) + C_n(t_{r-1}), & \text{otherwise,} \end{cases} \quad (21)$$

where $C_n(t_{r-1})$ is the expected difference between bike returns and demands at station n at time t_{r-1} and computed according to Corollary 5.

Thus, $B_n(t_{r-1})$ in Equation (20) can be replaced with the above expression, and our target is to find the minimum initial bike pool size such that:

$$P(\emptyset | B_n(t_{r-1})) \leq \epsilon, \quad (22)$$

where ϵ is the desired blocking probability. Hence, our optimization problem, for any station n , can be formalized as:

$$\text{minimize} \quad B_n(0) \quad (23a)$$

$$\text{given} \quad (17) \sim (22) : \forall r \in \{1, 2, \dots, R\}. \quad (23b)$$

In the above optimization problem, the decision variable is $B_n(0)$, i.e., the initial bike pool size at station n , $1 \leq n \leq N$. By solving this problem, we find the minimum number of bikes at any station that can satisfy the constraints given by equations (17) \sim (22).

B. Steady-State Analysis with the Engset Model

Our second approach is based on a steady-state analysis of the busiest period (with the largest bike demand). This gives an upper bound for the required number of bikes.

1) *Busiest Period*: To find the busiest period at any station n , $1 \leq n \leq N$, we also divide the timeline into short epochs according to the train schedule $\{t_r : i = 1, 2, \dots, R\}$. For each time epoch $(t_{r-1}, t_r]$, $1 \leq r \leq R$, the bike returns and demands at station n are given by $U_n(t_r)$ in Equation (17) and $D_n(t_r)$ in Equation (18), respectively, which have expected values of $\mu_n(t_r)$ and $\lambda_n(t_r)$ defined in Corollary 1 and 2, respectively.

We define the expected *satisfaction degree* at station n at time t_r (denoted by $\zeta_n(t_r)$) as the difference between $\mu_n(t_r)$ and $\lambda_n(t_r)$:

$$\zeta_n(t_r) = \mu_n(t_r) - \lambda_n(t_r).$$

The expected busiest period at station n is the one with the smallest satisfaction degree:

$$Q_m = \{(t_{m-1}, t_m] : \zeta_n(t_m) = \min\{\zeta_n(t_r) : r = 1, 2, \dots, R\}\}.$$

Note that due to different arrival and return time distributions (recall the example in Figure 3), the busiest periods at different stations might be different.

2) *Steady-State Blocking Probability*: To perform steady-state analysis during the busiest period, similar to that in [16], we apply the Engset model to estimate the blocking probability. The Engset model is a variant of the Erlang (loss) model which assumes a finite population of potential customers in a customer-server queue system.

For any station n , we set up the Engset model as follows.

- There are c servers (i.e., bikes) and M potential customers:

$$c = B_n(0) + S_n \text{ and } M = S_n + \sum_{1 \leq k \leq N, k \neq n} S_k H_{k,n}. \quad (24)$$

- The mean arrival rate of service requests is α , and the mean service rate of the servers is β , where according to our estimations in Equation (1) and Equation (2):

$$\alpha = \lambda_n(t_m) \text{ and } \beta = \mu_n(t_m), \quad (25)$$

where m is the busiest period index defined in Section V-B1.

- The service requests among the customers are independent, and the service time is independent of the thinking

time (the time between the end of service and next service request of the customer).

For steady-state analysis, we define the following RV:

$Y_n(t)$ = number of bikes rented from station n side at time t .

The stochastic process $\{Y_n(t), t \geq 0\}$ is a continuous-time Markov chain with finite state space $E = \{0, 1, \dots, c\}$. According to the equilibrium distribution of the Engset loss model [22], the long-term fraction of lost service requests, i.e., the blocking probability in our context, is:

$$P(c) = \frac{\binom{M-1}{c} \rho^c (1-\rho)^{M-1-c}}{\sum_{i=0}^c \binom{M-1}{i} \rho^i (1-\rho)^{M-1-i}}, \quad (26)$$

where

$$\rho = \frac{1/\beta}{1/\alpha + 1/\beta}.$$

3) *Pool Size Minimization*: Our target is to find the minimum $B_n(0)$ for each station n such that the following condition can be satisfied:

$$P(c) \leq \epsilon. \quad (27)$$

Thus, the optimization problem can be formalized as:

$$\text{minimize} \quad B_n(0) \quad (28a)$$

$$\text{given} \quad (24) \sim (27). \quad (28b)$$

The decision variable in the above optimization problem is $B_n(0)$, i.e., the initial bike pool size at station n , $1 \leq n \leq N$. By solving this problem, we find the minimum number of bikes at any station that can satisfy the constraints given by equations (24) \sim (27).

C. Solution Methodology, Complexity and Implementation

To solve the optimization problems formulated in Equations (23) and (28), and find an optimal pool size $B_n(0)$ at each station, we use the following two-phase algorithm, inspired by that used by Carpenter et al. [16].

- Bracketing: Given an initial small value of $B_n(0)$, compute the corresponding blocking probability (Equations (22) or (27)); if the desired threshold is not satisfied, then double the value of $B_n(0)$. Repeat this process until the desired blocking probability is reached, which results in a pool size range of $[B_L, B_H]$ where $B_L = B_H/2$.
- Binary Searching: Use binary search within the range of $[B_L, B_H]$ to find out the smallest value B_M that can satisfy the desired blocking probability threshold.

Equations (17) and (18) have factorial terms and therefore can be expensive to compute, especially with a large number of arrivals. Note that the Poisson probability decreases dramatically as the values of two RVs, U_n and D_n , increase. For example, for a Poisson process with mean value $\lambda = 15$, the probability of exceeding 32 is extremely small, i.e., $P(x > 32) \approx 0$. Thus, for efficiency, we can ignore values larger than a pre-defined threshold. In addition, computing the blocking probability in Equation (26) also becomes intractable when $\binom{M-1}{c}$ is large. Nevertheless, approximations can be

applied, such as the numerically stable approximation given by Reference [23].

For bike pool sizing at station n , the problem defined in Equation (23) has a complexity of $\mathcal{O}(RT_1T_2 \log m_0^*)$, where:

- R is the number of trains stopping at the station, which is usually less than 100 in practice;
- T_1 is the time it takes to compute i) the mean values of the bike return/demand processes μ_n/λ_n in Equations (1)/(2) and ii) the maximum bike return/demand counts \hat{S}_n/\check{S}_n in Equations (3)/(4) in each train's interarrival time, which amounts to simple summation operations in short time intervals;
- T_2 is the time it takes to compute the probability $P(\emptyset|m)$ in Equation (20), which can be done efficiently by exploiting the aforementioned optimizations;
- $\log m_0^*$ reflects the complexity of bracketing and binary searching described above, where m_0^* is the obtained optimal value, which depends on the number of customers travelling to station n .

For bike pool sizing based on steady-state analysis, the problem defined in Equation (28) can be solved in $\mathcal{O}(T_1'T_2' \log m_0^*)$ time, where:

- T_1' is the time to find the busiest period Q_m , which requires computing μ_n and λ_n , $1 \leq n \leq N$;
- T_2' is the time to compute the probability $P(c)$ in Equation (26), which again can be done efficiently by exploiting the aforementioned approximation.

After solving the bike pool sizing problem, we obtain initial sizes for each bike station, i.e., the values of $B_n(0)$, $n = 1, 2, \dots, N$. These values indicate the number of bikes that should be deployed at each station at the beginning of a day.

D. Model Comparison and Discussion

The advantage of the steady state method is that it is faster to compute and requires less information—only the bike return and demand rates. Thus, the steady state method can be directly used by current bikeshare systems with historical bike return/demand data. On the other hand, as we will show in Section VI, the steady state method, which is based on the busiest period, is prone to over-provisioning compared to the transient state method.

While the transient state method outputs smaller pool sizes, it requires more information: train schedules and travel times, separated customer departure/return distributions, etc. However, another benefit of the transient state method is that it computes multiple pool sizes, one for each arrival of a train, and therefore can be used to guide dynamic bike dispatching [20]. Thus, instead of ensuring that there are $B_n(0)$ bikes at station n at the beginning of the day, we can examine the expected number of bikes required throughout the day and reposition bikes as needed.

So far, we have only considered one railway line. Nevertheless, our solutions can be easily adapted to multiple lines. The main complication is that some customers may transfer from one train line to another enroute to their final destination. Fortunately, these transfers do not require the use of bikes and therefore the structure of our model is not affected. Only the

travel time matrix Δ may change due to the additional transfer times.

Also, we assumed a linear bi-directional railway (with trains from opposite directions assumed to arrive at the same time, as per Section III-B). However, our techniques also apply to circular railways. We can still assume that customers arrive on trains from two directions, as long as we do not “double-count” the arrivals (we assume that all customers travelling from station $i + 1$ to station i do so directly rather than going all the way round through station $i - 1$).

VI. EVALUATION

In this section, we use Monte Carlo simulations to test our bike pool sizing methods.

A. Experimental Setup

We use the following simulation parameters based on real-world data (Montreal Metro train schedules [24] and customer departure/return distributions from the Montreal BIXI open dataset [25]):

- Number of stations (N): Based on the average number of stations per railway line in the Montreal Metro system, we set $N = 18$ with station IDs $1 \sim 18$.
- Central stations: To examine the impact of customer traffic among stations, we test two scenarios: 1) a symmetric scenario in which all 18 stations can be home or remote stations for any customer, and 2) an asymmetric scenario in which stations with IDs $6 \sim 11$ are central stations and only these can serve as remote stations (details in Section VI-B).
- Number of customers at each station ($S_n, 1 \leq n \leq N$): We randomly generate a number between 100 and 200.
- Population partition matrix (H): We randomly partition the customer population between each station and the central stations using a uniform distribution.
- Daily cycle ($[t_0, t_e]$): By referring to the opening/closing times of the Montreal Metro system, we define the daily cycle as [6:00am, 12:00am].
- Time resolution: We set the time resolution as one minute, so that 18×60 samples are generated in total during one simulation.
- Number of trains (R): For each direction, we set the number of trains passing through each station to 90.
- Train inter-arrival time ($X_r, 1 \leq r \leq R$): Based on the train inter-arrival time of the Montreal Metro system, we randomly choose a number between 3 and 11 minutes.
- Train travel time matrix (Δ): We randomly choose a number between 5 and 10 minutes as the travel time between two adjacent stations.
- Customer departure/return distributions (V/Θ): We first extract a number of empirical departure/return distributions from the Montreal BIXI open data, and then randomly choose one of them for the population at each station.
- Blocking probability threshold (ϵ): We set the threshold to 5%, and thus the expected bike availability is 95%.

Here, we name the expected bike availability “Quality of Service (QoS)”.

Our evaluation methodology is as follows. First, we generate random parameter values, as described above, and use them to build the traffic (S, H, X and Δ) and customer (V and Θ) matrices. Next, we compute pool sizes for each station using each of our two techniques. Finally, using these pool sizes, we simulate customer and train arrivals and compute the actual blocking probability ϵ' . Based on the actual blocking probability, we define $aba := 1 - \epsilon'$ as the actual bike availability.

Additionally, we compare our techniques to a simple baseline referred to as *naïve provisioning*: we allocate two bikes for each customer, one at the home side and one at the remote side. This gives a blocking probability of zero given our assumption about daily routines and local/remote stations. We report the values of the following two metrics:

- Bikes per customer (abbreviated *bpc*) of the system:

$$bpc := 1 + \frac{\sum_{n=1}^N B_n(0)}{\sum_{n=1}^N S_n}, \quad (29)$$

where $B_n(0)$ is the minimized pool size with our method at station n and S_n is the number of customers with local station n . Note that the “1” in the equation represents that each customer has at least one bike at the home side.

- Bike saving ratio (abbreviated *bsr*) at the remote side:

$$bsr := 1 - \frac{\sum_{n=1}^N B_n(0)}{\sum_{n=1}^N S_n}, \quad (30)$$

and the “1” in this equation represents that each customer has one bike at the remote side (in naïve provisioning).

Note that the bike saving ratio defined in Equation (30) is a relative reduction in the number of bikes, with a value region of $[0, 1]$. The absolute reduction of bikes of the system, however, is computed by $1 - \frac{bpc}{2}$ with a value region of $[0, 0.5]$, as each customer at least has one bike (i.e., $bpc_{min} = 1$).

B. Performance Analysis

To numerically evaluate the performance of our pool-sizing algorithms, we study two types of scenarios. We believe these scenarios to represent extreme cases, and therefore our results can be interpreted as lower and upper bounds on bike pool sizes in real cities. We note that it is certainly possible to study our algorithm in a variety of other scenarios. Our focus here is to demonstrate the approach taken in our work, rather than the specific numerical values so obtained. The two types of scenarios are:

- Symmetric scenarios: Here, customers from any local station randomly choose a remote station, resulting in nearly symmetric customer traffic at each station; i.e., the number of local customers at any one station is similar to the number of incoming customers from other (remote) stations.
- Asymmetric scenarios: customers from any local station randomly choose one of a small number of central stations (stations with ID $6 \sim 11$ in our simulations) as their

destinations, which results in asymmetric customer traffic at each central station. At central stations, the number of incoming customers is about 3 times higher than the number of local customers.

1) *Overall & Detailed Results:* We generated 100 symmetric and asymmetric scenarios, respectively, using the parameters in Section VI-A. Table II shows the average values of aba , bpc and bsr for naive provisioning and our techniques under symmetric and asymmetric scenarios. Our Method 1 corresponds to transient-state analysis and Method 2 corresponds to steady-state analysis. We conclude that our methods reduce the required number of bikes per customer, and therefore improve bike utilization and reduce costs. In particular, Method 1 reduces the number of bikes per customer from 2 to $1.25 \sim 1.62$, for a savings of 39 to 75 percent at the transit station. Method 2 requires slightly more bikes per customer but still outperforms naive provisioning.

Symmetric Scenarios: Next, we show all the 100 symmetric scenarios and examined the bpc , bsr and aba of each one. Results are shown in Figure 5 (a), (b) and (c), respectively. We conclude that our techniques consistently outperform the baseline (naive provisioning) in all tested metrics and provide a blocking probability near zero percent (recall that $\epsilon = 0.05$).

Asymmetric Scenarios: Similarly, Figure 6 shows the bpc , bsr and aba for the 100 asymmetric scenarios. Compared to symmetric scenarios, the improvement in bpc and bsr with our techniques is lower. This is because in asymmetric scenarios, the number of incoming customers is much higher than that of local customers at some stations, leading to a much higher bike demand (as well as return) during peak hours. Thus, some stations must have very large bike pools to meet the specified blocking probability threshold. Note that for Method 1, although the aba values from some scenarios are below the QoS requirement line (0.95), most of them can meet the requirement and the overall average value of the actual bike availability is 0.971.

2) *Robustness Testing:* Since the estimated customer departure/return distributions (i.e., V/Θ) in our models might not precisely match the true bike rentals and returns, we now study how our methods are affected by changes in V and Θ . To do this, we modify our simulations as follows. If the number of arrivals during some time interval is calculated to be n , we instead set it to a random value in the interval $[n(1-\sigma), n(1+\sigma)]$, where σ is a *deviation index* randomly set to a number between zero and one. Figure 7 plots the actual blocking probability ϵ' as a function of the deviation index σ for both our methods and for both symmetric and asymmetric scenarios. Each datapoint corresponds to the average value of ϵ' over 100 simulation runs. The horizontal dotted line corresponds to the blocking probability threshold, $\epsilon = 0.05$. We conclude that:

- Method 2 is robust to perturbations in V and Θ and maintained the blocking probability below 0.05 even when σ was as high as 0.4. This is because Method 2 naturally calculates large pool sizes based on the busiest period.
- Method 1 is robust for symmetric scenarios (nearly as robust as Method 2), but less so for asymmetric scenarios,

where $\sigma > 0.1$ caused it to exceed the desired blocking probability. Intuitively, Method 1 naturally produces tighter pool sizes and therefore is more sensitive to deviations in arrival distributions.

VII. DISCUSSION

A. Why Commuters?

We now justify our choice to study commuters as the users of the bikeshare system. Although it is true that many people use a bikeshare system for irregular travel, its use for daily commuting has become more frequent. For example, in New York, “*biking has become part of New York’s commuting infrastructure...*” and “*about one in five bike trips is by a commuter*” [26]. Similarly, according to bicycle commuting data for the US, “*the number of bike commuters is on the rise*” and “*from 2000 to 2013, bicycle commuting rates in large BFCs² increased 105%*” [27]. Thus, with the population of bicycle commuters getting larger, it is interesting to investigate their bicycle usage patterns and build a better bike sharing system for them. Moreover, a bike sharing system helping commuters with their first/last mile travel would stimulate more people to use public transportation, which aligns well with the environmental goals of most governments.

A second reason to study commuters is because their behaviour (e.g., their departure/arrival distributions with respect to the public transportation system) are more predictable and amenable to mathematical modeling, making it possible to provide a mathematically well-grounded sizing solution. In contrast, it is challenging, if not impossible, to model the transit behaviour of irregular users of a bikeshare system.

B. Why Static Dispatching Every Night?

In the paper, we assume that bikes are repositioned once every 24 hours at night or early morning. Although repositioning bikes throughout the day is certainly feasible, given that bikeshares operate in dense city centers, many cities chose to reduce the time taken to reposition bikes, as well as avoid adding to traffic congestion, by repositioning bikes only late at night. For example, as reported by the Institute for Transportation and Development Policy (ITDP) [28], “*Many systems, however, try to do most of the redistribution at night, when there is less traffic and it is more efficient*”. Similarly, recent work assumes that “*the number of bikes at each station is known in advance and will not be changed during the rebalancing operation (when system is closed or during midnight)*” [29].

Nevertheless, note that our first method (i.e., the transient-state analysis using the differences of r.v.s) is compatible with dynamic re-balancing strategies. This is because it computes multiple pool sizes, one for each arrival of a train, and therefore can be used to guide dynamic bike dispatching. Thus, using this approach, instead of ensuring that there are $B_n(0)$ bikes at station n at the beginning of the day, we can examine the expected number of bikes required throughout the day and reposition bikes as needed.

²Bike Friendly Communities

TABLE II
SIMULATION RESULTS AVERAGED OVER 100 RUNS; $\epsilon = 0.05$

	Symmetric Scenarios			Asymmetric Scenarios		
	Naïve provisioning	Method 1	Method 2	Naïve provisioning	Method 1	Method 2
Bikes per customer (bpc)	2	1.25	1.44	2	1.62	1.75
Bike savings ratio (bsr)	0	0.753	0.562	0	0.385	0.252
Actual bike availability (aba)	1.0	1.0	1.0	1.0	0.971	0.997

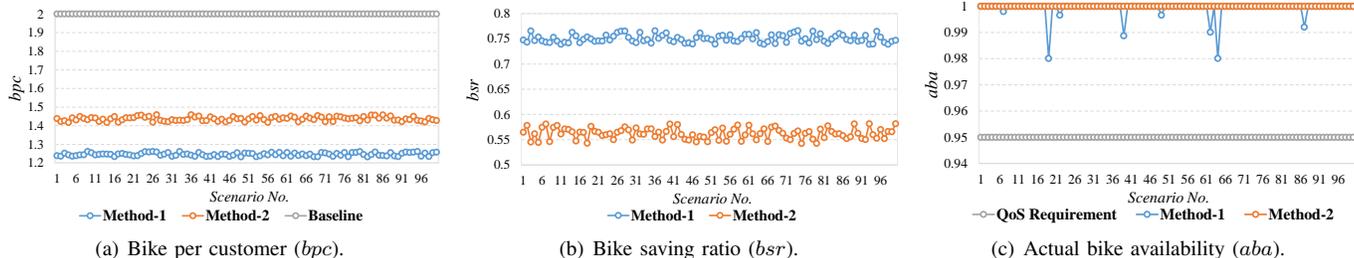


Fig. 5. Detailed results from five simulation runs using symmetric scenarios; QoS Requirement = 0.95.

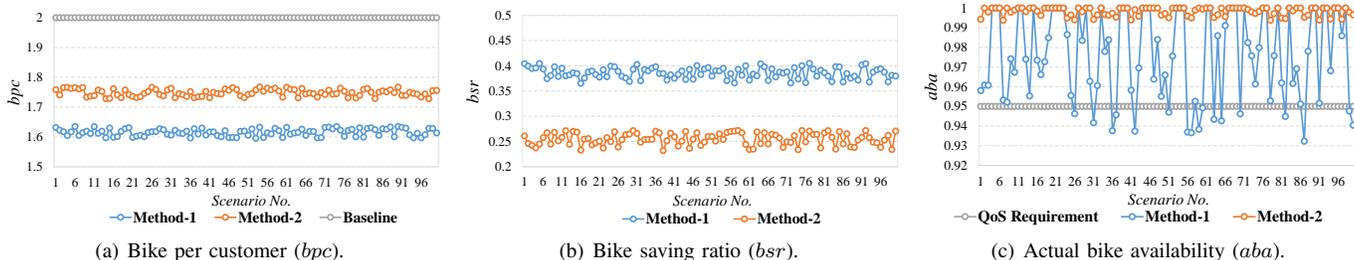


Fig. 6. Detailed results from five simulation runs using asymmetric scenarios; QoS Requirement = 0.95.

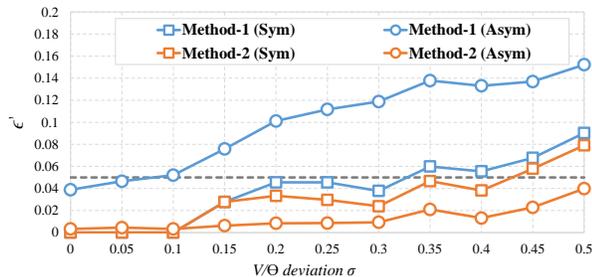


Fig. 7. Robustness testing: V/Θ deviation vs. actual blocking probability.

C. Optimal Dock Sizing

Our model computes the size of the dock at each station to be at least as large as $\max_t B_n(t)$ (recall Section IIIA). However, it is possible to pose the *dock sizing* problem as the decision on how large to make each bike station dock, given the pool size and anticipated trends in bikeshare usage. Jointly solving the dock sizing problem along with bike-pool sizing, however, is much more challenging, especially in our scenario of multimodal transit. Moreover, Free-Floating Bike Sharing (FFBS) and dock-free bikes have become popular (such as Mobike and Ofo in most cities in China) [18], [30]. This alleviates the need for precise dock sizing in new generation bike sharing systems. Therefore, we do not consider optimal dock sizing in this paper.

VIII. CONCLUSIONS

We addressed the new problem of bikeshare pool sizing in the context of bike-and-ride multimodal transit. To solve

this problem, we determined the smallest number of bikes that should be available at each public transit station at the beginning of every day to guarantee a desired blocking probability threshold. We presented two solutions: one based on transient-state analysis of bike arrivals and departures throughout the day, and the other based on the Engset model of steady-state analysis during the busiest period. Monte Carlo simulation results showed that our techniques can reduce the required pool size at the public transit station by 39 to 75 percent compared to a baseline solution. Since the pool sizes suggested by the steady-state method are based on the busiest period, they were larger and therefore less sensitive to errors in estimating customer flows.

An interesting direction for future work is to compare regular bikeshares and electric bikeshares in the context of multimodal transit. Electric bicycles are more expensive but can be used to travel longer distances, thereby increasing the cycling reachable regions for each public transport station.

ACKNOWLEDGEMENTS

This work was supported by Cisco Systems Canada and the Canadian Natural Sciences and Engineering Research Council under a Cooperative Research and Development grant, the Canadian Natural Sciences and Engineering Research Council under grant No.195819339, the NSERC Canada Research Chair, and the National 973 Basic Research Program under grant No.2014CB347800.

REFERENCES

- [1] J. Gorzelany, "The world's worst traffic jams, ever," <http://www.forbes.com/sites/jimgorzelany/2015/10/15/the-worlds-worst-traffic-jams-ever/#7c86b7476c6c>, Oct. 2015.
- [2] Y. Bambrick, "The integration of bikes and transit: Multi-modal goes the distance," <http://cutaactu.ca/en/node/592>, May. 2016.
- [3] J. Bachand-Marleau, J. Larsen, and A. El-Geneidy, "Much-anticipated marriage of cycling and transit: How will it work?" *Transportation Research Record: Journal of the Transportation Research Board*, no. 2247, pp. 109–117, 2011.
- [4] J. Pucher and R. Buehler, "Integrating bicycling and public transport in north america," *Journal of Public Transportation*, vol. 12, no. 3, p. 5, 2009.
- [5] P. Rietveld, "The accessibility of railway stations: the role of the bicycle in the netherlands," *Transportation Research Part D: Transport and Environment*, vol. 5, no. 1, pp. 71–75, 2000.
- [6] K. Krizek and E. Stonebraker, "Bicycling and transit: A marriage unrealized," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2144, pp. 161–167, 2010.
- [7] P. DeMaio, "Bike-sharing: History, impacts, models of provision, and future," *Journal of Public Transportation*, vol. 12, no. 4, p. 3, 2009.
- [8] NICHES, "New seamless mobility services: Public bicycles," http://www.nybikeshare.org/file_download/NICHES_public_bikes.pdf.
- [9] Wikipedi, "List of bicycle-sharing systems," https://en.wikipedia.org/wiki/List_of_bicycle-sharing_systems, Dec. 2016.
- [10] N. Zart, "Californian cities start electric bike-share system," <https://cleantechnica.com/2017/12/15/californian-cities-start-electric-bike-share-system/>, Dec. 2017.
- [11] J. C. García-Palomares, J. Gutiérrez, and M. Latorre, "Optimizing the location of stations in bike-sharing programs: a gis approach," *Applied Geography*, vol. 35, no. 1, pp. 235–246, 2012.
- [12] S. Ji, C. R. Cherry, L. D. Han, and D. A. Jordan, "Electric bike sharing: simulation of user demand and system availability," *Journal of Cleaner Production*, vol. 85, pp. 250–257, 2014.
- [13] J.-R. Lin and T.-H. Yang, "Strategic design of public bicycle sharing systems with service level constraints," *Transportation research part E: logistics and transportation review*, vol. 47, no. 2, pp. 284–294, 2011.
- [14] J.-R. Lin, T.-H. Yang, and Y.-C. Chang, "A hub location inventory model for bicycle sharing system design: Formulation and solution," *Computers & Industrial Engineering*, vol. 65, no. 1, pp. 77–86, 2013.
- [15] J. Chen, X. Chen, H. Jiang, S. Zhu, X. Li, and Z. Li, "Determining the optimal layout design for public bicycle system within the attractive scope of a metro station," *Mathematical Problems in Engineering*, vol. 2015, 2015.
- [16] T. Carpenter, S. Keshav, and J. Wong, "Sizing finite-population vehicle pools," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1134–1144, 2014.
- [17] L. Cagliero, T. Cerquitelli, S. Chiusano, P. Garza, and X. Xiao, "Predicting critical conditions in bicycle sharing systems," *Computing*, vol. 99, no. 1, pp. 39–57, 2017.
- [18] A. Pal and Y. Zhang, "Free-floating bike sharing: Solving real-life large-scale static rebalancing problems," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 92–116, 2017.
- [19] F. Cruz, A. Subramanian, B. P. Bruck, and M. Iori, "A heuristic algorithm for a single vehicle static bike sharing rebalancing problem," *Computers & Operations Research*, vol. 79, pp. 19–33, 2017.
- [20] C. Contardo, C. Morency, and L.-M. Rousseau, *Balancing a dynamic public bike-sharing system*. Cirrelt, 2012, vol. 4.
- [21] A. Gauthier, "The bike-share planning guide," Institute for Transportation and Development Policy (ITDP), New York, NY, Tech. Rep., 2013.
- [22] H. C. Tijms, *A first course in stochastic models*. John Wiley and Sons, 2003.
- [23] V. B. Iversen, "Teletraffic engineering and network planning," 2015.
- [24] Wikipedi, "Montreal metro," https://en.wikipedia.org/wiki/Montreal_Metro, Dec. 2016.
- [25] Manulife, "Bixi: Open data," <http://montreal.bixi.com/en/open-data>, Dec. 2016.
- [26] The New York Times, "More new yorkers opting for life in the bike lane," www.nytimes.com/2017/07/30/nyregion/new-yorkers-bike-lanes-commuting.html, Jul. 2017.
- [27] League of American Bicyclists, "Bicycle commuting data," www.bikeleague.org/commutingdata, Accessed: Sept. 2017.
- [28] Institute for Transportation and Development Policy, "The bike-share planning guide," https://www.itdp.org/wp-content/uploads/2014/07/ITDP_Bike_Share_Planning_Guide.pdf, Accessed: Sept. 2017.
- [29] J. Liu, L. Sun, W. Chen, and H. Xiong, "Rebalancing bike sharing systems: A multi-source data smart optimization," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1005–1014.
- [30] S. Pham, "China's 'dockless' bike sharing could be coming to a street near you," <http://money.cnn.com/2017/07/05/technology/startups/china-bike-sharing-global/index.html>, Jul. 2017.



Guoming Tang (S'12-M'17) received the Bachelor and Master degrees from the National University of Defense Technology, China, in 2010 and 2012, respectively, and the PhD degree in Computer Science from the University of Victoria, Canada, in 2017. He is currently an Assistant Professor at the National University of Defense Technology, China. His research interests include sustainable/green computing, machine learning and optimization.



Srinivasan Keshav (M'99) received a Ph.D. in Computer Science from the University of California, Berkeley in 1991. He was subsequently a member of technical staff at Bell Labs and, from 1996 to 1999, an Associate Professor at Cornell. In 1999 he left academia to co-found Ensim Corporation and GreenBorder Technologies Inc. He has been at the University of Waterloo since 2003, holding a Canada Research Chair and subsequently the Cisco Chair in Smart Grid.



Lukasz Golab (M'11) is an Associate Professor at the University of Waterloo and a Canada Research Chair. He holds a BSc in Computer Science from the University of Toronto and a PhD in Computer Science from the University of Waterloo, winning the Alumni Gold Medal for top PhD graduate. He has published over 80 articles on data analytics and data management systems.



Kui Wu (S'98-M'02-SM'07) received the BSc and the MSc degrees in computer science from the Wuhan University, China, in 1990 and 1993, respectively, and the PhD degree in computing science from the University of Alberta, Canada, in 2002. He joined the Department of Computer Science, University of Victoria, Canada, in 2002, where he is currently a Full Professor. His research interests include smart grid, mobile and wireless networks, and network performance evaluation. He is a senior member of the IEEE.