This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Sizing Finite-Population Vehicle Pools

Tommy Carpenter, *Student Member, IEEE*, Srinivasan Keshav, *Member, IEEE*, and Johnny Wong

***Abstract*—We refer to a *vehicle pool* as a number of vehicles at a single location used for the same purpose. We focus on the problem of sizing vehicle pools for a finite set of *subscribers* who can use the pool. Our goal is to minimize the number of vehicles in the pool while still meeting nearly all subscriber requests. Formally, we propose three analytical techniques to size a vehicle pool for a *finite population* of subscribers, according to the pools' *busy period demand* to guarantee all requests are served with probability $1 - \epsilon$, i.e., a *quality-of-service* (QOS) guarantee. Moreover, we propose an additional heuristic sizing method, which requires no prior data about pool demand. Although this method does not provide probabilistic bounds on QOS, we show in practice that it still achieves a high QOS. We evaluate our sizing methodologies using seven years of data from a local car share, using three performance metrics: *availability* (percentage of requests served), *utilization* (the percentage of time that vehicles in the pool are used) and *member-to-vehicle ratio* (the size of the pool relative to the size of its user population). We show that our methods perform well with respect to these metrics.**

***Index Terms*—Car shares, electric vehicles (EVs), optimization, queueing theory, transportation planning.**

I. Introduction

CONSIDER the following three problems. First, suppose that a battery electric vehicle (BEV) dealer wishes to offer its customers an internal combustion vehicle (ICV) to use for long trips. How many ICVs will the dealership need for these customers? Second, suppose a car share wishes to size their fleet, such that customers are provided with a high quality of service (QOS) and high vehicle utilization (underutilized vehicles reduce profits). What is their optimal fleet size? Finally, suppose a private (not publicly accessible) parking lot is to be sized; how many spaces should it have?

These sizing problems have several factors in common:

1) The pool of resources (cars or parking spaces) has a *finite population* of *subscribers*, e.g., those allowed to use the pool. That is, there is a limited number of people from which resource requests can be generated.
2) The pool should be sized, so that the probability that any subscriber request is met exceeds $1 - \epsilon$.
3) Demand for the pool vehicles is nonstationary, i.e., varying with time. Vehicles are most commonly used during the day, and moreover, there may be holidays during which the demand for vehicles is much higher than the rest of the year (first two problems) or not at all (third problem).

As we discuss in Section II, limited work has been dedicated to this important problem. While busy period sizing has been studied extensively in the context of infinite-population systems, we are not aware of any prior work which incorporates busy period sizing into finite-population systems. Thus, this is the first work to size a finite-population vehicle pool, with respect to its busy period demand.

Our work is primarily motivated by the first problem. To give an idea of the importance of this problem, several studies find that "range anxiety"—BEV owners' fear of depleting their battery before reaching their destination—remains a critical barrier to BEV adoption [1]–[6]. While these studies show that current BEV specifications suffice for the majority of drivers' needs, because BEVs have limited battery range, they are not capable of completing long trips, which worries potential BEV owners. Three solutions have been proposed to address this problem: battery switching, widespread fast electric vehicle (EV) charging outlets, and larger BEV batteries. *Battery switching stations*, i.e., stations where depleted batteries are replaced in minutes with charged batteries, may pose a long-term solution to the limited range of BEVs but will be initially sparsely deployed because of their large initial cost of $\approx 500\,000$ [7]. Publicly available fast-charging outlets would allow owners to charge their BEVs quickly during trips. Unfortunately, although overnight charging presents the least burden on the electrical grid, most driving is done during the day [1], [8]–[11]. Finally, the most expensive component of a BEV is its battery. EV batteries currently cost \$400–\$500/kWh [1], [12], but each kilowatthour adds only ≈ 5 km of range [13], [14]. Increasing the size of EV batteries is, therefore, an expensive solution to this problem.

Recently, BMW [15] has offered a simpler solution: they suggest that dealerships reserve a portion of their unsold ICVs for BEV customers. When in need, BEV owners can use an ICV from the pool for their trip. However, the problem of *sizing* such pools remains.

Our three main contributions are the following:

1) We give three analytical techniques to size finite-population pools subject to a given QOS (see Section III).
2) We give one method to heuristically size a finite-population pool that requires no data (see Section IV), and show that it still provides a very high QOS.
3) We numerically evaluate all four sizing methods using data from a large Waterloo-based car share (see Section V). We find that our sizing methods allow for high vehicle availability and reasonable (with respect to current practice) vehicle utilization and pool sizes.

Manuscript received April 30, 2013; revised July 30, 2013; accepted November 26, 2013. The Associate Editor for this paper was H. Dong.

The authors are with the Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: tcarpent@uwaterloo.ca; keshav@uwaterloo.ca; jwwong@uwaterloo.ca).

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.2013.2293918

1524-9050 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

II. Problem Formulation and Related Work

The problem of sizing a vehicle pool is simple to formulate (but difficult to solve). The pool should be the minimal size, such that the probability that a subscriber submits a request that is rejected, i.e., there are no pool vehicles available, is less than an advertised QOS ϵ. This sizing problem is hard to solve due to nonstationary demand from the finite population. Busy period sizing, i.e., sizing a system according to the period of highest demand, has been studied extensively in the case of infinite-population systems, such as telephone networks [16] and the electrical grid [17]. However, we are the first work to size finite-population systems with nonstationary demand.

We briefly make a note about advance *reservations*. Many car shares allow their subscribers/customers to reserve a vehicle for a predetermined period of time. Our sizing methods can be used whether subscribers make reservations for pool vehicles or simply show up to the pool. In the former case, the pool should be sized, such that the probability that a reservation cannot be made or held is less than ϵ.

There are few research efforts dedicated to mathematically sizing parking lots or car shares based on their expected usage. Parking lot sizes are restricted by real estate availability, budgets, and municipal sizing laws based on the occupancy of the accompanying building(s). It may also be hard to judge the demand for a parking lot before it is built. Moreover, car shares are sized according to company economics, and private companies often do not wish to disclose their business model.

We are aware of only two relevant papers in this space. Hampshire and Gaites [18] present the closest related work; the authors focus on the sizing and profitability of a peer-to-peer vehicle sharing service. However, their sizing algorithm makes two assumptions that do not apply to our problem. First, the authors assume that the arrival of subscribers to car shares is stationary. It is not, particularly near major holidays, i.e., some car shares even have higher rates during peak periods [19]. Thus, much of our work deals with sizing according to busy periods. Second, the authors assume that the population of subscribers is infinite. We instead focus on smaller pools that cannot be approximated with an infinite-population model.

Arbatskaya *et al.* [20] discuss sizing a parking lot game-theoretically. They study the utility and disutility of (not) finding a parking space, and modeling the welfare to drivers spending time competing with others for remaining parking spaces. While the authors show several interesting analytical results, their sizing algorithm is not applicable to our problem. The authors size lots to maximize game-theoretic objectives, using factors such as drivers' disutility in arriving early to increase their chance of finding a space and the competition (game) between multiple drivers. We instead focus on sizing pools, such that a randomly arriving subscriber is served according to a QOS. Moreover, the authors find that, game-theoretically, the size of the lot should be equal to the number of drivers who use the lot, which is infeasible in practice.

Some of our sizing methods requires knowledge about the pools' expected utilization, i.e., how often and when subscribers will make requests. Several studies have attempted to measure drivers' mobility patterns to determine how often and how far drivers commonly drive their own vehicles [21]–[23] or car share vehicles [24]–[27]. These mobility analyses help estimate pool demand.

Reports, detailing the utilization, the member-to-vehicle (M2V) ratios, and the QOS of existing car shares [19], [26], [28]–[32], are used as baselines for our evaluation results (see Section V-B).

While we do not discuss the problem of pool pricing in this paper, several authors have studied the economics of parking lots [33]–[38] and car share subscriptions [18], [19], [28], [39].

III. Statistically Sizing a Pool Subject to a QOS

We provide three analytical sizing techniques subject to a QOS, based on information about the subscribers' mobility patterns. We give a binomial-based sizing method in Section III-A, which only requires knowing the average number of times per year the average subscriber will arrive. We give a queueing-theory-based algorithm in Section III-B, which requires knowing, on average, how often subscribers need vehicles and how long they need them for. Finally, we give a different queueing-theory-based method in Section III-C, which requires a data set tracking the arrival times of subscribers. Throughout this section, we use the term *loss probability* to denote the probability that a random subscriber arrives to an empty pool, given the pool has size m vehicles, and denote it by $p(b|m)$. Here, we assume that the goal is to size the pool, such that $p(b|m) < \epsilon$.

A. Binomial-Based Sizing

Our first analytical technique is based on the binomial distribution, which can be used to conservatively size a pool with limited data about the subscribers. Consider a pool on a given day. Assume that all subscribers in need of a vehicle travel to the pool at hour h and will return their vehicle before h the following day. These two assumptions mean that the same vehicle cannot be reused multiple times a day and no vehicle is used for multiple days in a row. We want to find m, such that $p(b|m) < \epsilon$. Let

- $p(a)$ be the probability that a subscriber arrives on a given day. In this conservative sizing method, we assume that all subscribers are independent.
- S is the total number of subscribers.
- m is the number of vehicles in the pool.
- ϵ is the QOS.

The probability that exactly k subscribers arrive is given by

$$p(k) = \binom{S}{k} p(a)^k \left(1 - p(a)\right)^{S-k}.$$

If exactly k subscribers arrive, $p(b|m,k)$ is given by

$$p(b|m,k) = \begin{cases} 0, & \text{if } k \le m \\ (k-m)/k, & \text{otherwise.} \end{cases}$$

Thus, given S subscribers, the probability that a random subscriber finds an empty pool is given by marginalizing out k, i.e.,

$$p(b|m) = \sum_{k=0}^{S} p(b|m,k)p(k). \quad (1)$$

Equation (1) gives $p(b|m)$ for a fixed m; an algorithm is given in Section III-D to optimize m.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

B. Sizing via the ELM

Pools can be sized using the *Engset Loss Model* (ELM), which is known in Kendall's notation as the $G_I/G_I/m/m/S$ queue. We treat arriving subscribers as jobs arriving to a queueing system. We further treat vehicles in the pool as parallel servers that serve incoming jobs. Each job (subscriber) that arrives receives dedicated service from one server (takes a vehicle) until the job is processed (the subscriber brings back the vehicle). We assume that subscribers will not wait in an empty pool; thus, there is no buffer in the system. Finally, there are only a finite number of sources from which jobs can be generated, i.e., the pool subscribers, giving us a finite-population model. Thus, we have a $G/G/m/m/S$ system, i.e., a variant of the ELM.

1) Model Requirements: As done in Section III-A, we wish to size the pool with m vehicles, such that $p(b|m) < \epsilon$. We derive $p(b|m)$ for the ELM, as follows. When a subscriber is borrowing a vehicle, we say they are in the *service state* and remain in this state, on average, for their *mean service time* (MST), i.e., the average duration they need a vehicle for. After finishing service, the subscriber enters the *thinking state* and waits on average their *mean think time* (MTT), i.e., their average duration between completing service and their next request. The ELM requires that all subscribers' think times be independent identically distributed (i.i.d.) according to an arbitrary distribution G_{tnk} with mean $1/\lambda_B$, and that all subscribers' service times be i.i.d. according to an arbitrary distribution G_{ser} with mean $1/\mu$. That is, all subscribers have the same MTT and MST and, moreover, that these are invariant over time. In Section III-B6, we show results for $p(b|m)$, which is a function of these two means.

Our sizing problem violates these two requirements. First, we have heterogeneous subscribers with different MTTs and MSTs. Moreover, we expect the pool to have *busy periods*, i.e., MTTs are not invariant to time. We deal with these violations in Section III-B2 and B4, respectively.

2) Modeling the Average Subscriber: Our key idea is to model a heterogeneous population of subscribers by modeling the *average subscriber*. For this sizing method, we assume that the following data can be collected from each subscriber $s = 1, \ldots, S$:

- s's MTT $1/\lambda_s$. Their *mean think rate* (MTR) is given by λ_s.
- s's MST $1/\mu_s$. Their *mean service rate* is given by μ_s.
- $p(s, B)$ is the probability that s arrives during a busy period.

These data can be collected from subscribers using customer surveys of the form "how many times per year will you need a vehicle?"; "how long are you likely to need it for?"; and "will you arrive during any of [these] busy times?"

3) Assumptions: We make and justify four assumptions.

- We assume that a weighted average of all subscribers' MTTs (see Section III-B5) is a good estimate of $1/\lambda_B$. In the ELM model, only the mean of G_{tnk} (not the distribution of think times) is needed to compute $p(b|m)$; hence, our assumption that subscribers are identically distributed according to G_{tnk} simply means that this weighted average is a good estimate of $1/\lambda_B$. Moreover, subscribers' think times are likely independent because each has their own mobility patterns and vehicle needs; hence, think times are independently drawn from G_{tnk}. Thus, think times are i.i.d. according to G_{tnk}.
- We assume that a weighted average of all subscribers' MSTs is a good estimate of $1/\mu$. We further hypothesize that the duration for which subscribers need their vehicles and the times they return their vehicles are independent. Thus, service times are i.i.d. according to G_{ser}.
- The derivations of $1/\lambda_B$ and $1/\mu$ in the following section assume that subscribers arrive at the pool at most once during each busy period. Since the busy period is a number of hours each day, or a day/weekend each year, it is unlikely that a subscriber will rent multiple times within one period.
- We assume that $p(b|m)$ is the same for all subscribers and is given by steady-state results for the ELM given later in this section. This assumption follows from our methodology of modeling the average subscriber.

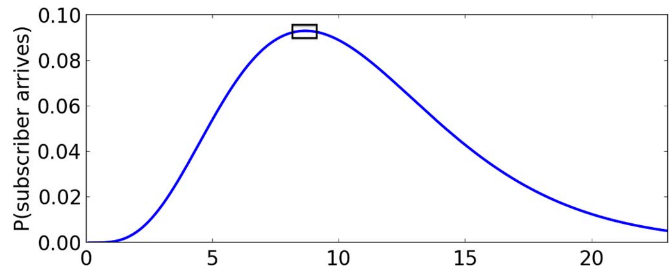

Fig. 1. Example arrival-time distribution over one day.

4) Nonstationarity and Busy Period Sizing: We now deal with the issue that subscribers' think times may decrease near busy periods; hence, think times are not stationary. A sizing technique known as *busy period modeling* allows us to remove this temporal correlation. The idea is to assume that the MTT during the busiest period, i.e., $1/\lambda_B$, *always holds* [40], [41]. The benefit of this assumption is as follows. Suppose the probability that subscriber s arrives to a pool at a particular time on a day is shown in Fig. 1. If many subscribers followed the same distribution, clearly [8 A.M., 9 A.M.] would be a busy period for the pool compared to the rest of the day. However, within this hour, as shown in the boxed region, the distribution is approximately uniform, e.g., subscribers who arrive during this busy period are equally probable to arrive at any time. That is, within the busy period, arrival times are approximately independent; hence, by assuming that it is always the busy period, arrival times are independent, and we have a stationary queueing system.

There are two busy-period-sizing methodologies. First, one can size the system according to the historical single busiest period of a defined length. The electrical grid, for example, may be sized according to the highest electrical demand during one hour ever recorded. A data set is needed to use this approach; see Section III-C. Alternatively, one can size the system according to the average demand during recurring busy periods. For example, a teletraffic network may be sized according to the call rate during all 5–6 P.M. periods. Here, we estimate the MTT during the average busy period, not the historical single busiest period.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Within this sizing methodology, there are two further options. We can size the pool according to a daily busiest period of K hours or a yearly busiest period of K contiguous days. For both options, the probability that subscriber s arrives during any busy period B, $p(s, B)$, is derived in the following section.

5) Deriving Sizing Parameters: Here, we use the following notation:

- A_B denotes the arrival rate of pool subscribers during B, i.e., one of the pools' busy periods.
- $c_s = 1/\lambda_s + (1 - p(b|m))1/\mu_s$ denotes subscriber s's *mean cycle time*, i.e., the mean time between s's requests, since, on average, they think for time $1/\lambda_s$ and then receive service for time $1/\mu_s$, at which point they begin thinking again. With probability $p(b|m)$, s is blocked and does not receive service and immediately goes back to thinking.
- $\omega_s = 1/c_s$ represent s's *cycle rate*, i.e., the "rate" at which they arrive, which represents how active a subscriber s is.
- n_B denotes the number of subscribers that arrive to the pool (including those blocked) during busy periods.
- K is the length of busy periods given as input in Section III-B4.
- τ denotes the percentage of time that the pool is in a busy period (based on K), which is given as input.

Consider the probability that subscriber s arrives during any busy period, i.e., $p(s, B)$. We first examine the case where there is a recurrent daily busy period. *Which* period of the day is busiest can be computed through customer surveys. We assume that the pool implementer breaks up a day into a set of arbitrary-length chunks, which we denote by $\mathcal{K}$. We define $\mathbf{t}_s$ to be the probability that a given arrival by s occurs during each chunk of the day. This can be derived from a subscriber survey of the form "on a day you need a vehicle, how likely is it that you arrive during [chunk 1], . . . , [chunk$|\mathcal{K}|$]." We stress that $\mathbf{t}_s$ does *not* give the probability that subscriber s arrives during each period of each day, but rather the probability that a *given* arrival occurs during each period. We estimate the busiest period, by finding the weighted average probability that a subscriber makes a request during each period, and then find the maximum of these probabilities over all periods, i.e.,

$$B = \arg\max_{k \in \mathcal{K}} \left(\frac{\sum_{i=1}^{S} (\omega_i \mathbf{t}_i[k])}{\sum_{j=1}^{S} \omega_j} \right).$$

To estimate $p(s, B)$, i.e., the probability that s arrives during any busy period, we assume that the probability that s arrives on any given day is uniform and given by ω_s, but take into account the nonuniformity over time of day, i.e.,

$$p(s, B) = \omega_s * \mathbf{t}_s[B]. \tag{2}$$

If we instead wish to size the pool according to the busiest contiguous block of days over the year, we assume that the pool implementer knows the period they wish to size for, e.g., "Mother's Day weekend." Without a data set, too much information would be needed from subscribers to estimate *which* day of the year would be the busiest; hence, we assume that the pool implementor can simply ask "out of the X times you expect to need a vehicle per year, how many do you expect to be during [busy block of days]?"[1] Then, $p(s, B)$ is given simply by the answer divided by X.

Next, we derive n_B, i.e., the number of users expected to arrive to the pool during any given busy period. We simply weight an arrival by each subscriber s by $p(s, B)$, i.e.,

$$n_B = \sum_{i=1}^{S} p(s, B)1. \tag{3}$$

We first derive $1/\mu$ as it is used to compute $1/\lambda_B$. Because service times are unaffected by busy periods, to derive $1/\mu$, we simply calculate a weighted average of subscribers' MSTs, i.e.,

$$\frac{1}{\mu} = \frac{\sum_{i=1}^{S} \omega_i \frac{1}{\mu_i}}{\sum_{j=1}^{S} \omega_j}. \tag{4}$$

We use the following rationale to derive $1/\lambda_B$: *the effective arrival rate of subscribers to the pool during busy periods is equal to the number of subscribers that arrive during the busy period divided by the length of the busy period.* In our terminology, this is written as

$$A_B = \frac{n_B}{K}. \tag{5}$$

Suppose we know the MTT of all users during the busy period, i.e., $1/\lambda_B$. Then, A_B is also given by

$$A_B = S\left(\frac{1}{c_B}\right) = S\left(\frac{1}{\frac{1}{\lambda_B} + (1 - p(b|m))\frac{1}{\mu}}\right). \tag{6}$$

That is, if $1/c_B$ represents the *average cycle time of each individual subscriber during busy periods*, then we expect S/c_B to arrive each time unit. By combining (5) and (6), we get

$$\frac{S}{\frac{1}{\lambda_B} + (1 - p(b|m))\frac{1}{\mu}} = \frac{n_B}{K} \rightarrow \frac{1}{\lambda_B} = \frac{SK}{n_B} - (1 - p(b|m))\frac{1}{\mu}. \tag{7}$$

Note that we compute $1/\lambda_B$ by assuming a value for $p(b|m)$ and for a fixed m. In the following section, we give results for $p(b|m)$ for a fixed m by assuming $1/\lambda_B$. We then give an iterative system to solve for both parameters in Section III-B7.

6) Blocking Probability: Using the ELM, the probability that all m vehicles are being used is given by

$$p(b|m) = \frac{\binom{S}{m}\rho^m}{\sum_{i=0}^{m}\binom{S}{i}\rho^i}, \qquad \rho = \frac{1/\mu}{1/\lambda_B} \tag{8}$$

$$= \frac{\binom{S}{m}\psi^m(1-\psi)^{S-m}}{\sum_{i=0}^{m}\binom{S}{i}\psi^i(1-\psi)^{S-i}}, \qquad \psi = \frac{\rho}{1+\rho}. \tag{9}$$

Equation (8) is the *Engset distribution* [16], [42], [43]. It is equivalent to (9), which is a *truncated binomial distribution*

[1]This can be easily generalized to the case where the pool implementor has knowledge that one of a few busy days or weekends will be the busiest, but needs to survey customers to determine which is actually the busiest.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[16], [43]. The variable ψ is known as "offered traffic" and is a measure of how busy the queueing system is (closer to one is busier). Unfortunately, both (8) and (9) become numerically unstable as S and m grow because $\binom{S}{m}$ becomes too large to compute with. Instead, we calculate $p(b|m)$ using the numerically stable approximation [16]

$$p(b|\rho, S, m) = \begin{cases} \frac{\rho\cdot(S-m+1)\cdot p(b|m-1)}{m+\rho\cdot(S-m+1)\cdot p(b|m-1)}, & \text{if } m > 0 \\ 1, & \text{if } m = 0. \end{cases} \quad (10)$$

7) Blocking Probability Iteration Algorithm: Note that (7) finds $1/\lambda_B$, assuming a value for $p(b|m)$, and (10) finds $p(b|m)$, assuming a value for $1/\lambda_B$. We now give an iteration algorithm to compute $p(b|m)$ for a fixed number of servers m and a desired accuracy σ:

Algorithm 1: Compute $p(b|m)$

1: Inputs: S, m, σ
2: Compute $1/\mu$ using (4)
3: set $k = 0$, $p(b|m)^{(k)} = 0.5$
4: **do**
5: Compute $(1/\lambda_B)^{(k)}$ using $p(b|m)^{(k)}$ and (7)
6: Compute $p(b|m)^{(k+1)}$ using $1/\lambda_B^{(k)}$, $1/\mu$, and (10)
7: $k++$
8: **while** $(|p(b|m)^{(k+1)} - p(b|m)^{(k)}| > \sigma)$

We stop when $p(b|m)$ changes between two iterations by less than σ, i.e., our desired accuracy. This iteration gives $p(b|m)$ for a fixed m; an algorithm is given in Section III-D to optimize m.

C. Data-Set-Based Sizing

If a data set gives 1) the times at which all requests are made and 2) the duration of all requests, it can be used to obtain more accurate sizing than the one given in the previous section. Tracking which subscribers make the requests is not needed. This is because we cannot compute ρ directly whether the data set keeps track of which subscribers arrive at which times.[2] Instead, we estimate their MTR during busy periods, as done in the previous section. However, a data set allows us to directly measure A_B, as well as $1/\mu$, leading to more accurate sizing since fewer approximations are made.

Suppose the granularity of the data set is H hours, i.e., suppose it is possible to measure the arrival rate during a period of H from the data set. Let $K = qH$, where q is an integer that represents the length of the segment the pool is to be sized for (the historical single busiest period of duration K, which must be some multiple of H), and A_i represents the arrival rate during each segment of length K. Suppose the data set is divided into n segments. We use a sliding window approach to calculate the arrival rate during the busiest segment seen in the data set, i.e.,

$$A_B = \max\left(A_0 = \frac{\sum_{i=0}^{q-1} H_i}{K}, \ldots, A_j = \frac{\sum_{i=j}^{q+j-1} H_i}{K}, \ldots\right). \quad (11)$$

It is straightforward to compute $1/\mu$ from the data set; we simply average the duration of all requests. Then, we compute $1/\lambda_B$ from (6); we do not need (5) or (7) because A_B is known. Equation (6) is iterate with (10) to obtain a pairing of $p(b|m)$, $1/\lambda_B$ for a given m.

We note that, if the goal is to obtain the most conservative sizing by finding the highest MTR observed during a period (or conversely, the lowest MTT observed), the data set should be examined at the finest granularity possible. That is, finding the MTR using chunks of a larger size will never be higher than the MTR found using chunks of a smaller size. This can be proven as follows. Let S be an arbitrary period of time. Suppose we divide S into c equally sized smaller periods. Denote the MTR during period i as λ_i. We have

$$\lambda_S = \frac{\sum_{i=1}^{c} \lambda_i}{c} \leq \frac{c(\max_{i=1,\ldots,c} \lambda_i)}{c} \leq \max_{i=1,\ldots,c} \lambda_i \quad \square.$$

As with the previous section, this iteration gives $p(b|m)$ for a fixed m; an algorithm is given in Section III-D to optimize m.

D. Optimizing the Number of Cars

Section III-A–C give three different methods of computing $p(b|m)$ for a fixed number of vehicles m, depending on what data are available. We use Program 1 to size pools, given these three sizing methodologies. Objective (12) is to minimize the number of vehicles (m) in the pool. Constraint (13) ensures that m is large enough to meet the QOS, but constraint (14) ensures m is less than the maximum number of vehicles that can be stored in the pool. Thus, this program is infeasible if $m_{\max}$ is too small given the number of subscribers and their usage patterns.

Program 1: Pool Sizing Integer Program

Inputs:

$m_{\max}$: the maximum size of the pool

ϵ : desired QOS

Decision Variables:

$m \in \mathbb{N}$: the number of vehicles to store in the pool

Objective:

$$\min\ m \quad (12)$$

Subject To:

$$p(b|m) < \epsilon \quad (13)$$

$$m \leq m_{\max}. \quad (14)$$

[2] If it does not, we cannot calculate an MTT or an MTR because it is unknown which arrivals correspond to which departures. Even if it does, the same user does not arrive and depart multiple times within one busy period (they may arrive but then return the vehicle after the busy period); thus, we cannot calculate the average subscribers' cycle time *within* busy periods.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Program 1, despite being an integer program, is solvable in $\mathcal{O}(TS \log m^\star)$, where T is the time it takes to compute $p(b|m)$ according to which of the three methods is being used, and $m^\star$ is the optimal solution, using Algorithm 2 as follows. First, during the *bracketing phase*, we keep doubling the pool size m_H, until $p(b|m_H) \leq \epsilon$. We also maintain the previous m value before this threshold is reached, which is denoted by m_L. Thus, we find a smaller interval $(m_L, m_H]$ than $[1, S]$, in which $m^\star$ lies. This process is logarithmic in $m^\star$ because we grow m_H at an exponential rate; hence, during the bracketing phase, in the worse case, we compute $p(b|m)$ $\log m^*$ times. When this procedure terminates, $m_L < m^\star \leq m_H$. We then use binary search on the interval $(m_L, m_H]$ to find $m^\star$.[3] Binary search is logarithmic in the width of the interval, which is at most $m^\star - 1$. The worst case is when $m_H = m^\star - 1$ and is doubled, giving us $m_H - m_L = (2m^\star - 2) - (m^\star - 1) = m^\star - 1$. Therefore, we compute $p(b|m)$ another $\log m^*$ times during the search phase. Thus, Algorithm 2 is in $\mathcal{O}(TS \log m^\star)$. Note that, if $m^\star$ is close to S, this algorithm has slightly worse runtime than standard binary search on the interval $[1, S]$, which has a worst case runtime of $TS \log S$. However, $m^\star$ will usually be much smaller than S.

Algorithm 2: Solution to Program 1

$m_L, m_H \leftarrow 1$
while $p(b|m_H) > \epsilon$ // Phase 1: Bracketing
  $m_L = m_H$
  $m_H = 2 * m_H$
while 1 // Phase 2: Binary Search
  $m_M \leftarrow (m_L + m_H/2)$
  if $p(b|m_M) \leq \epsilon$ **then** $m_H \leftarrow m_M$ **else** $m_L \leftarrow m_M$
  if $m_H - m_L \leq 1$
    if $m_H \leq k$ **then return** m_H **else return** "Infeasible"

IV. Dynamic Resizing

If a strict QOS does not need to be statistically guaranteed, a pool can be resized over time as actual demand is observed. Here, we present a dynamic resizing algorithm that does not rely on historical data and does not make assumptions about subscriber behavior. This method differs from the three previous methods in three ways: 1) it does not statistically guarantee a QOS; 2) m is not static, i.e., it changes over time as subscribers arrive to the pool; and 3) it does not use Program 1 to set m. This method is motivated by the algorithm used for transmission control protocol (TCP) congestion control: *additive increase multiplicative decrease* [44]–[46]. In TCP congestion control, a sender begins with a small window size, i.e., the number of packets they could have sent and are waiting to receive an acknowledgment for at any time. The sender then increases their window size by $\alpha \in \mathbb{N}$ repeatedly ("additive increase") until it is too large and a packet loss is observed due to network congestion. At this point, their window size is multiplied by a fraction $\beta \in (0, 1)$ ("multiplicative decrease"), at which point the process repeats.[4] This results in a sawtooth curve that converges to the optimal size if α and β obey[5] [45], [46]

$$0 < \alpha < \text{pool size}, \ \alpha \in \mathbb{N} \tag{15}$$
$$0 < \beta < 1. \tag{16}$$

Here, we take the opposite approach and propose a *linear-decrease–multiplicative-increase* resizing algorithm. Suppose the initial size of a pool is $m = TCP_{\text{in}}$, where $m = TCP_{\text{in}}$ is set using a reasonable guess. If no subscribers are blocked after y subscriber arrivals, the pool is resized to $m = m - \alpha$, where α obeys (15), which simply states that the pool size is not reduced to zero. This process repeats until a subscriber arrives to an empty pool, which is analogous to a packet loss. At this point, the pool size becomes m/β, where β obeys (16). Note that y controls the convergence rate at the risk of a larger number of unserviced subscribers. This value should be made small if we wish to reduce the pool size as fast as possible at the risk of a larger number of unserviced subscribers, or large if we wish to limit the number of future unserviced subscribers.

V. Evaluation

To evaluate our sizing methodologies, we use reservation data from a local car share (see Section V-A). We use the three primary performance metrics used to evaluate car shares (see Section V-B) to evaluate our sizing methods. We present our methodology in Section V-C and our results in Section V-D.

A. Data Set

We use data from a local car share, i.e., Community Carshare [47], to evaluate our sizing methodologies. The data set is composed of all reservations made with all share vehicles since the reservation system was put into place in early 2005. The data set is composed of seven years of reservations and spans from March 2005 to October 2013. There are 51 223 reservations in the data set, each detailing the start time and end time of the reservation, as well as the distance driven in kilometers. Fig. 2 shows the number of "active" members over time; for each time t on the x-axis, the y-axis shows the number of unique members who made at least one reservation within the past year of t, which we use as a proxy for the number of active members at any given time.

B. Performance Metrics

We evaluate our methods using the same three metrics that car shares use to evaluate their business.

- *M2V* is the number of subscribers for each vehicle in the pool.

[3] $p(b|m)$ is monotonically decreasing with increasing m; hence, we can create an array $[p(b|m_L), \ldots, p(b|m_H)]$, which is naturally sorted in decreasing order. We can then use binary search to find the minimum index, such that $p(b|m) < \epsilon$. We do not need to compute all the values of this array; however, we only need to work with the boundaries and midpoint at each step.

[4] In TCP Reno congestion control, $\alpha = 1$, $\beta = 0.5$.

[5] This is true if all senders use the same values for α and β [45], [46].

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

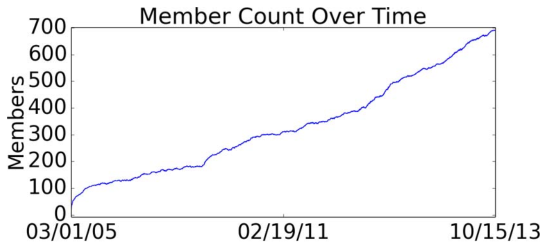

Fig. 2. Active members versus time.

- *Utilization* is the percentage of each day that the average pool vehicle is occupied by a subscriber.
- *Availability* is the percentage of served requests $(1 - \epsilon)$.

The goal is to maximize all three simultaneously; however, while a higher M2V ratio leads to a higher utilization, it also leads to lower availability. For comparison, we report available performance metrics for rental agencies and car shares. Existing car shares and rental companies have traditionally both aimed to maximize utilization at the expense of availability, because profit is directly proportional to utilization. For the BEV pool application, however, availability may be essential to profit, i.e., advertising high availability would help sell more BEVs.

1) Utilization: Utilization data for car shares are limited. City Carshare [19] reports that the world's oldest and most successful car share, i.e., Mobility Switzerland, has now achieved a utilization of almost 40%. Autoshare and Flexcar report utilizations of 20%–30% [31]. Rental vehicle operations have much higher utilizations than car shares. Most rental companies do not require a membership to rent a vehicle; hence, their pool of potential renters on any given day is potentially "infinite," as compared to a car share with known subscribers. Consequently, the total number of requests is greater, as is the probability of a request during off peak hours (simply due to the larger number of total requests); hence, rental agencies can perform scheduling to maximize utilization. Rental vehicle utilizations of 65% or higher are considered standard [48]–[52]. In 2012, Hertz, which is one of the worlds' largest rental vehicle companies, reported a utilization of 67% [52]. The article "How To Run A Car Share" supports these numbers, stating that a successful car share has a utilization of 20%–40%, whereas a successful rental agency has a utilization of double that [53].

2) M2V Ratios: Shaheen *et al.* report that, in U.S., car shares had M2V ratios of 37 : 1 in 2006 [30] and 49 : 1 in 2008 [29]. Interestingly, Canadian shares have lower M2V ratios; as of 2008, Canadian car shares had an average of 24 : 1 [29]. Zipcar reports the 2012 M2V ratio of nearly 50 : 1 [54]. The M2V ratio metric does not apply to rental agencies because membership is not required to rent a vehicle.

3) Availability: Both vehicle sharing and vehicle rental availability data are limited. In an early study in 1998, Shaheen *et al.* [26] reported that the world's most successful car shares aimed for an M2V of 15–20 : 1, which led to an availability of 95%, i.e., a value of $\epsilon = 0.05$. Brook [32] also states that "a M2V ratio of 20 : 1 provides the best balance between utilization and availability for a car share," but does not give an availability target. We could not find any vehicle rental availability data.

TABLE I
NOTATION REFERENCE TABLE

Name	*Description*
$p(\cdot)$	denotes "probability of $(\cdot)$"
S	the number of pool subscribers
s	indexes a subscriber
$p(a)$	daily Binomial probability each subscriber will make a request (§III-A)
$p(b\|m)$	blocking probability given a pool size of m
m	the size of the pool
$1/\lambda_s$	subscriber s's mean think time (MTT)
$1/\lambda_u$	subscriber u's mean service time (MST)
$p(s, B)$	probability s arrives during a busy period
$1/\lambda_B$	MTT of all subscribers during busy periods
$1/\mu$	MST of all subscribers
B	the pool's busiest period(s)
A_B	the arrival rate of subscribers to the pool during B
c_s	s's mean cycle time
ω_s	$1/c_s$; s's cycle rate
n_B	number of subscribers that arrive during busy periods
K	the length of the pools busy periods (§III-B5)
$\mathcal{K}/\|\mathcal{K}\|$	the set/number of pools potential busy periods (§III-B5)
τ	the percentage of time the pool is in a busy period
ρ	$(1/\mu)/(1/\lambda_u)$, a measure of subscribers' activeness, the ratio of time they are in vs. out of the system
σ	desired iteration accuracy of Algorithm 1
TCP_{in}	initial guess of pool size for dynamic resizing (§IV)
α, β	dynamic decrease and increase parameters (§IV)
y	# of served requests before decrease triggered (§IV)

C. Evaluation Methodology

Here, we describe our data set sampling methodology and the parameters computed for each method. For our discussion, we abbreviate each of our sizing methodologies, i.e., we refer to the binomial based method (see Section III-A) as BIN, the ELM based on subscriber surveys (see Section III-B) as ELM_S, the ELM that uses the data set (see Section III-C) as ELM_D, and the dynamic resizing method (see Section IV) as TCP (since it is based on TCP congestion control).

1) Bootstrapping: We use *random bootstrapping with replacement* [55] and *cross validation* to evaluate our sizing methodologies as follows. We perform I iterations. On each iteration, half of the data set is selected at random. This half, which is known as the training set, is used to compute parameters needed for the sizing methodologies (with the exception of TCP; see Section V-C3). The other half, which is known as the test set, is used to evaluate the performance metrics (M2V, utilization, and availability) for each sizing methodology. After each iteration, the entire data set is sampled again in the same fashion (hence with replacement). Thus, we obtain a total of I samples of the three performance metrics. Bootstrapping treats each performance metric as a sampling statistic. We therefore obtain a total of I values for each of these sampling statistics and, thus, are able to compute confidence intervals.

2) Input Configuration: The following parameters in Table I needed for various sizing methodologies are given as input: ϵ, K, $|\mathcal{K}|$, TCP_{in}, α, β, and y. Moreover, for ELM_S, the number of daily chunks and the length of each chunk are given.

3) Training Parameters: During each sampling iteration, the following training parameters are computed from the training set. For all methods, we compute S, i.e., the number of subscribers, and for all methods except TCP, we compute m s.t. $p(b|m) < \epsilon$ (for TCP, m is not a constant). For BIN,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

we compute $p(a)$ as the mean probability that each subscriber needs a vehicle on a given day. Using the terminology defined in Section III-B5, the probability that subscriber i needs a vehicle on a given day is ω_i, which represents "i arrives once every c_i days," or, alternatively, "the probability i arrives on a given day is ω_i"; hence, we compute $p(a) = \sum_{i=1}^{P} \omega_i / P$. For ELM$_S$, each subscriber's MST and MTT are computed from their reservations. Then, n_B, $1/\mu$, and $1/\lambda_B$ are computed based on the input parameters (discussed in the prior section and the methodology in Section III-B). For ELM$_D$, we first compute A_B, using a sliding window according to K, and then $1/\mu$ and $1/\lambda_B$. There is no training for TCP.

4) Performance Metric Computation: We now describe how we compute the performance metrics. During each iteration, the test set is used to compute a stream of arrivals and departures referred to as an arrival departure stream (ADS). We form two lists containing the sorted times of all arrivals and departures (departure refers to a member returning their vehicle) seen in the test set. These two lists are then merge-sorted into one stream. When there is a tie, i.e., an arrival and departure both occurring at the same date and time, we assume that the arrival comes first. This results in one stream of arrivals and departures.

For BIN, ELM$_S$, and ELM$_D$, the pool size m is computed during the training phase. During each iteration, we replay the ADS for these three methodologies using Algorithm 3. Availability is computed as $1 - blocked$/size(ADS). To compute utilization, the out array is used to compute a Riemann sum; the value of each element in the array (the number of vehicles out at that time) is multiplied by the length of time between each array element, and these values are summed. This gives the total number of "utilized vehicle-hours." This number is then divided by $m*$ (duration of test set), which represents the maximum number of vehicle hours possible, assuming that every vehicle was always out, giving us the percentage of time that each vehicle is used. Finally, we annotate each rental reservation in the data set with the number of active members at that time. This allows us to compute the average number of pool subscribers M. This can be seen as horizontally sampling half the values in Fig. 2. We then compute the M2V as M/m.

Algorithm 3: Replay ADS

```
// out array tracks the number of vehicles out of the pool
blocked_count, out[0] = 0
for i in range [0, len(ADS))
  if ADS[i] == "Arr" and out[i − 1] < m // not blocked
        out[i] = out[i − 1] + 1
  else if ADS[i] == "Arr" // blocked
        out[i] = m, blocked_count + +
  else out[i] = max(0, out[i − 1] − 1) // departure
```

For TCP, we use the same algorithm, with a few modifications. First, whenever there is a blocked subscriber, the pool size is increased by dividing by $\beta < 1$. Moreover, the number of nonblocked arrivals since the last block is tracked, and once this counter reaches y, m decreases by α, and this counter is reset. Since the pool size for TCP varies (m is not constant like the other three methods), utilization and M2V are computed differently. We compute a Riemann sum of the out array, as for the other methods, and further keep track of the changing TCP pool size to compute a similar Riemann sum. Finally, we divide the former by the latter to compute utilization. The utilization computations are illustrated graphically in Fig. 3. We compute the M2V for TCP by dividing M by the average of the changing TCP pool size.

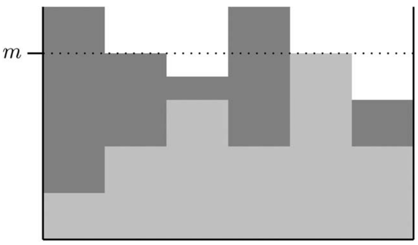

Fig. 3. Light gray area la represents the number of cars out over time. The m line shows the pool size for any of the first three methods. Utilization is computed as la/(area of m rectangle). The dark gray area da represents the changing TCP pool size. TCP utilization is computed as la/da.

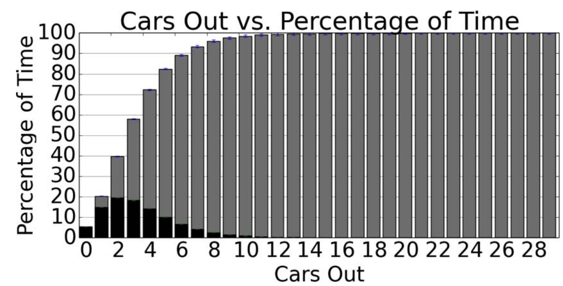

Fig. 4. Black distribution shows the probability of a specific number of vehicles being out of the pool. The gray distribution shows the cumulative distribution function.

5) Overall Versus Busy Period Utilization: In the following section, we present results for the three performance metrics (M2V, utilization, and availability). We compute utilization and availability across the entire data set, rather than within busy periods. We take this approach for two reasons. First, we conclude from the literature discussed in Section V-B that, while car shares are nonstationary and thus experience higher demand during busy periods, overall utilization and availability are still the key performance metrics that determine profitability. Busy period utilization and availability affect the overall QOS, but our sizing methods consider this: by sizing for busy periods, our advertised QOS is always met. Second, we experiment with different analytical methods that size according to different busy periods. Overall utilization and availability metrics are comparable across methods, but metrics reported within the respective busy periods would not be easily comparable because the methods use busy periods with different lengths and they occur at different times.

D. Results

The results of our numerical evaluation are shown in Figs. 4 and 5. In Fig. 5, the notation $D(\cdot)$ indicates that $\cdot$ is the length

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

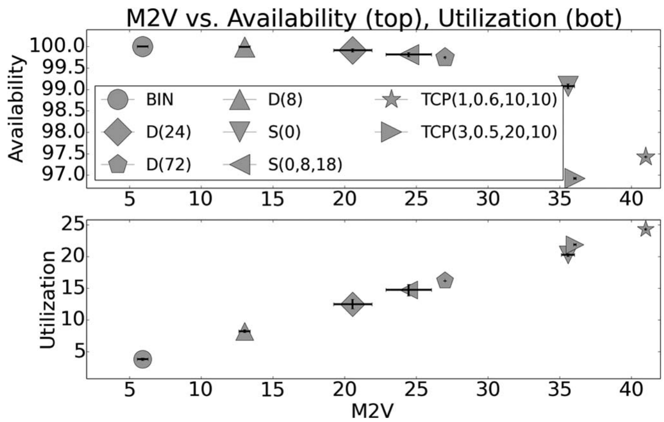

Fig. 5. Comparison of achieved performance metrics.

of the historical single busiest period the pool is sized for. $S(\cdot)$ indicates that $\cdot$ is the set of daily chunks used; each number gives the start time of a new chunk. For example, $S(0, 8, 18)$ indicates three daily chunks: [12 A.M.–7:59 A.M.], [8 A.M.–5:59 P.M.], and [6 P.M.–11:59 P.M.]. The TCP notation is in the form $(\alpha, \beta, y, TCP_{\rm in})$. The QOS $\epsilon = 0.1$, corresponding to a busy period availability target of 90%. All confidence intervals are normal 99% intervals over $I = 100$ iterations.

We make three related observations from the results, followed by a comparison of each sizing method in Section V-E. First, we see that the car share used for our case study has rare but very high peak periods. As shown in Fig. 4, the average number of vehicles out at any time is approximately three, but the peak of nearly 30 is ten times larger. Because the busy period demand is much greater than the average demand, and because these periods occur rarely, the busy-period-sizing methods lead to a very conservative sizing. Sizing to meet 90% of all busy period arrivals leads to a 99% overall availability, i.e., a high QOS, but this comes at the cost of lower overall utilization. Unfortunately, we cannot solve both problems using a static sizing method, i.e., either we allow for a low QOS during busy periods or we suffer from lower overall utilization. The dynamic sizing method, i.e., TCP, solves this problem. TCP outperforms all other methods because it can increase the pool size during periods of high demand and decrease the pool size during periods of low demand. This allows for a high QOS, during busy periods, and a relatively high overall utilization. Moreover, it provides high availability in practice, although it does not guarantee any QOS. Finally, we see that the best performing non-TCP method is $S(0)$, i.e., it has the smallest pool size, the highest M2V, and the highest utilization. This supports our previous discussion; $\text{ELM}_S(0)$ performs better than the other methods because it sizes for the average *day*[6] instead of sizing for busy periods. We further note that $S(0)$ and TCP achieve a utilization in the range of other successful car shares, as previously discussed.

We briefly detail the reason for variability in the performance metrics. As discussed in Section V-C1, we sample the M2V, utilization, and availability distributions I times to compute confidence intervals. In each iteration, the subscribers seen in the test set differ, which causes m, as determined by the sizing methods, to vary. Moreover, M varies with each test set. These two factors contribute to the variance in M2V, which are shown as horizontal error bars. During each iteration, the ADS also differs, which leads to different computations of availability and utilization because the Riemann sums change. The computations of utilization and availability also vary as the pool size varies; hence, the vertical error bars on utilization and availability reflect all of the aforementioned variations.

[6]Under the parameter of 0, $S(0)$ states that there is only one daily chunk starting at midnight that lasts 24 h. Thus, the probability that each subscriber arrives during any daily busy period is the same as their probability of arriving on any day. Specifically, $\mathbf{t}_s[B]$ in (2) is 1 (the whole day is the busy period); hence, each subscriber s's busy hour weight $p(s, B) = \omega_s$, i.e., their probability of arrival on any day. Thus, the pool is sized according to the average day.

E. Method Comparison: Advantages and Disadvantages

BIN offers the highest availability, but at the expense of an increased pool size and decreased utilization. It should be used if the pool demands very high availability and pool size (hence cost) is not a concern. It still determines a pool size much lower than the number of members; thus, it is still better than the simplest approach of sizing with one vehicle per member.

ELM_S is the most mathematically complex method, but it is also the most versatile. Configured with a set of daily time periods, it can statistically size the pool for busy periods of any length, according to a desired QOS, and only requires a simple customer survey. Moreover, it can be "transformed" into a nonbusy-period-sizing method when configured with a parameter of 0 to size for the average day, which we have shown to work well for pools with rare high-demand periods.

ELM_D, although it is the most "informed" method (it uses a mobility data set for training), does not outperform ELM_S. This suggests that a simple survey suffices for the sizing problem.

TCP requires no data and no training, yet outperforms all other methods. However, it may be problematic to implement in practice, i.e., constantly adding and removing vehicles from a pool, particularly in large numbers, may not be economically feasible. However, this method may work well in a *multipool* environment (left as future work), where vehicles can be moved between several pools owned by the same entity. As demand in various pools change, vehicles can be moved between pools to provide the system-wide high availability, utilization, and M2V, without having to constantly buy or sell vehicles.

VI. Future Work

We suggest three avenues to address our work's limitations.

- Let $F_i : \mathbb{R} \to \mathbb{R}$ be a function that denotes subscriber i's disutility derived from a specific value of $p(b|m)$, i.e., the QOS of the pool. Subscribers' disutility of driving to an empty pool may increase nonlinearly, perhaps exponentially, as $p(b|m)$ increases. Consider modifying Program 1

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

by removing constraint (13) and modifying objective (12) as

$$\arg\min_{m} \left(\sum_{i=1}^{S} F_i \left(p(b|m) \right) \right). \tag{17}$$

This formulation minimizes subscriber *risk*, instead of forcing an explicit QOS ϵ. Exploring this alternative formulation may be a fruitful avenue for future work.

- In this paper, we have focused on sizing a single pool. However, the problem becomes more complex when multiple pools are owned by the same entity, allowing for vehicles to be moved between the pools according to demand. New sizing methodologies may be needed for the multipool case to model the sharing of vehicles between pools.
- We have not taken pricing into account. The price of the pool depends on the pool size m; thus, it may be as important as ϵ. Future work would be to size a pool using a multiobjective optimization program.

VII. Conclusion

Prior work has not studied sizing finite-population vehicle pools with nonstationary demand. Vehicle demand varies with time of day and time of year, and moreover, many pool applications have a finite population; hence, there is a need for such sizing methodologies. We propose four such methods. Three of them size to meet a QOS ϵ. Our fourth method, although it does guarantee a QOS, requires no data or training, performs well, and still offers a high QOS. We further show that each of our methods has advantages and disadvantages, with respect to various performance metrics.

Regarding our main application of interest, we note that there are several advantages for potential BEV buyers and BEV dealerships of offering this pooling service.

- Integrating this service into dealerships significantly reduces subscribers' *transactional cost*. Dealerships could collect all subscribers' information at the time of purchase, so that subscribers need not fill out paperwork each time they obtain an ICV. Moreover, because the subscription is offered by the dealership, subscribers would not have to compare the price of several rental vehicle agencies to determine the cheapest option.
- Dealerships can internally compute the cost of this service and amortize this cost into the price of their BEVs. The service can then be sold as "free" to potential customers, which would appeal from a marketing perspective.
- Dealerships already maintain a pool of vehicles for customers awaiting repairs for their vehicles; thus, our approach does not impose any radical changes to current practices.

Acknowledgment

The authors would like to thank Community Carshare [47] for sharing their data for our evaluation and P. Azimzadeh, L. Patel, and A. Schmieder for various contributions.

References

[1] A. Boulanger, A. C. Chu, S. Maxx, and D. L. Waltz, "Vehicle electrification: Status and issues," *Proc. IEEE*, vol. 99, no. 6, pp. 1116–1138, Jun. 2011.

[2] A. Everett, M. Burgess, M. Harris, S. Mansbridge, E. Lewis, C. Walsh, and S. Carrol, "Initial findings from the ultra low carbon vehicle demonstrator program," Technology Strategy Board, Swindon, U.K., Technology Strategy Board Report, Sep. 2011.

[3] M. Nilsson, "Electric vehicles: The phenomenon of range anxiety," ELVIRE, Gif-sur-Yvette, France, ELVIRE Report, Jun. 2011.

[4] "Gaining traction: A customer view of electric vehicle mass adoption in the U.S. automotive market," New York, NY, USA, Deloitte Report, Jun. 2010.

[5] "Gauging interest for plug-in hybrid and electric vehicles in select markets," London, U.K., Ernst & Young Report, Jun. 2010.

[6] S. Carroll, "The smart move trial: Description and initial results," cenex, Leicestershire, U.K., Centre of Excellence for Low Carbon and Fuel Cell Technologies Report, Mar. 2010.

[7] J. Yarow, The Cost of a Better Place Battery Swapping Station: $500 000, Apr. 2009, accessed Jan. 11th, 2011. [Online]. Available: http://tinyurl.com/crgqzx

[8] K. Clement, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 371–380, Jan. 2010.

[9] S. Hadley and A. Tsvetkova, "Potential impacts of plug-in hybrid electric vehicles on regional power generation," *Elect. J.*, vol. 22, no. 10, pp. 56–68, Nov. 2009.

[10] S. Blumsack, C. Samaras, and P. Hines, "Long-term electric system investments to support plug-in hybrid electric vehicles," in *Proc. IEEE Power Energy Soc. Gen. Meet.*, Jul. 2008, pp. 1–6.

[11] S. Hadley, "Evaluating the impact of plug-in hybrid electric vehicles on regional electricity supplies," in *Proc. iREP Symp.—Bulk Power Syst. Dyn. Control*, Aug. 2007, pp. 1–12.

[12] R. Hensley, S. Knupfer, and D. Pinner, Green Tech Media: EV batteries plummet in price: Down to a kWh, Green Tech Media, Boston, MA, USA. [Online]. Available: http://www.greentechmedia.com/articles/read/ev-batteries-dropping-rapidly-in-price/

[13] *The Nissan Leaf*, Nissan, Dallas, TX, USA, Jan. 2011, accessed June 28th, 2011.

[14] *Chevrolet Volt Specifications*, Chevrolet, Detroit, MI, USA, Jan. 2011, accessed July 5th, 2011.

[15] A. Ingram, BMW electric car owners get gas-powered loaner for longer trips, HighGear Media, Menlo Park, CA, USA, accessed April 20th, 2013. [Online]. Available: http://www.greencarreports.com/news/1082784_bmw-electric-car-owners-get-gas-powered-loaner-for-longer-trips?utm_source=GreenCarReports&utm_medium=twitter

[16] V. B. Iversen, *Teletraffic Engineering and Network Planning (2009 version)*. Kgs. Lyngby, Denmark: Techn. Univ. Denmark, 2009.

[17] T. Carpenter, S. Singla, P. Azimzadeh, and S. Keshav, "The impact of electricity pricing schemes on storage adoption in Ontario," in *Proc. e-Energy*, 2012, pp. 1–10.

[18] R. C. Hampshire and C. Gaites, "Peer-to-peer carsharing: Market analysis and potential growth," in *Proc. TRB Annu. Meet.*, 2010, pp. 119–126.

[19] E. Sullivan and L. Magid, *Bringing Car-Sharing to Your Community*. San Francisco, CA, USA: City Carshare, 2007.

[20] M. Arbatskaya, K. Mukhopadhaya, and E. Rasmusen, "The parking lot problem," Indiana University, Bloomington, IN, USA, Techn. Rep., Jul. 2004.

[21] N. Pearre, W. Kempton, R. L. Guensler, and V. V. Elango, "Electric vehicles: How much range is required for a days driving?" *Transp. Res. C*, vol. 19, no. 6, pp. 1171–1184, Dec. 2011.

[22] J. Gonder, T. Markel, A. Simpson, and M. Thornton, "Using GPS travel data to assess the real world driving energy use of plug-in hybrid electric vehicles (PHEVs)," presented at the Transportation Research Board (TRB) 86th Annu. Meet., Washington, DC, USA, Jan. 2007 Paper NREL/CP-540-40858.

[23] R. L. Guensler and B. M. Williams, "The role of instrumented vehicle data in transportation decision making," in *Proc. 4th Int. Conf. Decision Making Urban Civil Eng.*, Nov. 2002, pp. 1–6.

[24] R. Cervero, A. Golub, and B. Nee, "City CarShare: Longer-term travel demand and car ownership impacts," *J. Transp. Res. Board*, vol. 1992, pp. 70–80, 2007.

[25] T. Stillwater, P. L. Mokhtarian, and S. A. Shaheen, "Carsharing and the built environment," *J. Transp. Res. Board*, vol. 1, no. 2110, pp. 27–34, 2009.

[26] S. Shaheen, D. Sperling, and C. Wagner, "Carsharing in Europe and North America: Past, present and future," *Transp. Quart.*, vol. 52, no. 3, pp. 35–52, 1998.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[27] K. M. N. Habib, C. Morency, M. T. Islam, and V. Grasset, "Modelling users' behaviour of a carsharing program: Application of a joint hazard and zero inflated dynamic ordered probability model," *Transp. Res. A*, vol. 46, no. 2, 2012.
[28] S. A. Shaheen, A. P. Cohen, and J. D. Roberts, "Carsharing in North America: Market growth, current developments and future potential," *J. Transp. Res. Board*, no. 1986, pp. 116–124, 2006.
[29] S. A. Shaheen, A. P. Cohen, and M. S. Chung, "North American carsharing: 10-year retrospective," *J. Transp. Res. Board*, vol. 1, no. 2110, pp. 25–44, 2009.
[30] S. A. Shaheen, A. Schwartz, and K. Wipyewski, "Policy considerations for carsharing and station cars," *J. Transp. Res. Board*, no. 1887, pp. 128–136, 2004.
[31] A. Lubinsky and C. Brown, "Is car sharing a threat to the auto rental business?" ARN, Torrance, CA, USA, Dec. 2006.
[32] D. Brook, "Carsharing—Start up issues and new operational models," presented at the Transportation Research Board 83rd Annu. Meet., Washington, DC, USA, Jan. 2004.
[33] S. P. Anderson and A. de Palma, "The economics of pricing parking," *J. Urban Econom.*, vol. 55, no. 1, pp. 1–20, Jan. 2004.
[34] J. O. Jansson, "Road pricing and parking policy," *Res. Transp. Econom.*, vol. 29, no. 1, pp. 346–353, 2010.
[35] A. Glazer and E. Niskanen, "Parking fees and congestion," *Regional Sci. Urban Econom.*, vol. 22, no. 1, pp. 123–132, 1992.
[36] E. Verhoff, P. Nijkamp, and P. Rietveld, "The economics of regulatory parking policies," *Transp. Res. A*, vol. 29, no. 2, pp. 141–156, Mar. 1995.
[37] R. C. Larson and K. Sasanuma, "Congestion pricing: A parking queue model," *Ind. Syst. Eng.*, vol. 4, no. 1, pp. 1–17, Sep. 2010.
[38] D. Gillen, "Parking policy, parking location decisions and the distribution of congestion," *Transportation*, vol. 7, no. 1, pp. 69–85, Mar. 1978.
[39] A. Millard-Ball, G. Murray, C. Jessica Ter Schure, and J. Fox, "Carsharing: Where and how it succeeds," Transportation Research Board, Washington, DC, USA, TCRP Rep. 108, 2005.
[40] J. Riordan, "Telephone traffic time averages," *Bell Syst. Tech. J.*, vol. 30, no. 4, pp. 1129–1144, Oct. 1951.
[41] J. Flood, *Telecommunications Switching, Traffic and Networks*. Upper Saddle River, NJ, USA: Pearson Education, 2001.
[42] L. Kleinrock, *Queueing Systems Volume 1*. Hoboken, NJ, USA: Wiley, 1975.
[43] H. Tijms, *A First Course in Stochastic Models*. Hoboken, NJ, USA: Wiley, 2003.
[44] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 5th ed. Reading, MA, USA: Addison-Wesley, 2009.
[45] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Comput. Netw. ISDN Syst.*, vol. 17, no. 1, pp. 1–14, Jun. 1989.
[46] Y. R. Yang and S. S. Lam, "General AIMD congestion control," in *Proc. ICNP*, 2000, pp. 187–198.
[47] Community Carshare. [Online]. Available: http://communitycarshare.ca/
[48] Automotive Insights no. 01_2011, Munich, Germany, Roland Berger Strategy Consultants, Jan. 2011. [Online]. Available: http://www.rolandberger.com/media/pdf/Roland_Berger_Automotive_Insights_1_2011_20110908.pdf
[49] *How to Analyze a Car Rental Company*, Investor Campus, Sandton, South Africa.
[50] J. Tenant, How to improve utilization and still satisfy the customer, ARN, Torrance, CA, USA, accessed April 10th, 2013. [Online]. Available: http://www.autorentalnews.com/article/print/story/2010/01/how-to-improve-utilization-and-still-satisfy-the-customer.aspx
[51] C. Brown, Car rental: A tough year for pricing? Business Fleet Mag., Torrance, CA, USA, accessed April 10th, 2013. [Online]. Available: http://www.businessfleet.com/blog/auto-focus/story/2012/02/car-rental-a-tough-year-for-pricing.aspx
[52] C. Rullo, Hertz: Increasing utilization feels good, accessed April 10th, 2013. [Online]. Available: http://seekingalpha.com/article/1327391-hertz-increasing-utilization-feels-good
[53] C. Brown, How to run a successful car-sharing operation, ARN, Torrance, CA, USA, accessed April 12th, 2013. [Online]. Available: http://www.autorentalnews.com/article/story/2009/09/how-to-run-a-successful-carsharing-operation/page/2.aspx
[54] A. Young, Avis buys Zipcar to compete better against Hertz and enterprise, IBTimes, New York, NY, USA, accessed April 10th, 2013. [Online]. Available: http://www.ibtimes.com/avis-buys-zipcar-compete-better-against-hertz-enterprise-has-car-sharing-finally-come-age-995762
[55] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, no. 1, pp. 1–26, 1979.

Tommy Carpenter (S'12) received the Bachelor's degree in computer science from Central Connecticut State University, New Britain, CT, USA, in 2010. He is currently working toward the Ph.D. degree in computer science at University of Waterloo, Waterloo, ON, Canada, where he is a member of the Information Systems and Science For Energy (ISS4E) research group.

Srinivasan Keshav (M'99) received the Bachelor's degree in computer science and engineering from Indian Institute of Technology Delhi, New Delhi, India, in 1986 and the Ph.D. degree in computer science from University of California at Berkeley, Berkeley, CA, USA, in 1991.

He is a Professor, ACM Fellow, Canada Research Chair, and Cisco Systems Research Chair in Smart Grid with the Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada.

Johnny Wong received the Ph.D. degree in computer science from University of California, Los Angeles, CA, USA, in 1975.

Since 1975, he has been a Professor with the Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada, where he was the Director from 2003 to 2006.