Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds.

GHOST SIMULATION MODEL FOR DISCRETE EVENT SYSTEMS, AN APPLICATION TO A LOCAL BUS SERVICE

Felisa Vázquez-Abad

Department of Computer Science
Hunter College of the City University New York
New York, NY 10065, USA

ABSTRACT

In this paper we present a simulation model for large networks that increases the efficiency compared to a discrete event simulation model. These networks have two different time scales: a fast one and a slow one. The main idea is to replace some of the faster point processes by a "fluid" (called the ghost processes) thus accelerating the execution of the simulation. Using local modularity for the code, there is no need to keep a list of events. Clocks are not necessarily synchronized. When a local clock advances due to a slower event, retrospective calculations recover the fine detail lost in the fluid model. Mathematically, the model is a special case of the Filtered Monte Carlo method. Efficiency improvement results not only from the speed of execution, but also from variance reduction. We provide proofs of unbiasedness. Throughout the paper we use a case scenario of an airport car park.

1 MOTIVATION

The motivation for the present work goes back to 2005, when the Melbourne airport (in Melbourne, Australia) needed some advice to buy a new fleet of buses for the airport car park. At the time, the author was at the University of Melbourne and she participated in finding the solution. Figure 1 depicts the route of the buses: From the checkpoint, they first pick up all passengers in the Arrivals terminal (a) that wish to go to the parking lots. Next they follow the loop leaving passengers and picking up new passengers that wish to go to the Departures terminal (there may be multiple ones). At the Departures terminal (d) all passengers still in the bus unload.

The airport needed to determine which vendor to use for the type of bus (they differ in bus capacity, loading times and operation costs) and the number of buses to buy. They required that a quality of service (QoS) criterion be satisfied, which they call the "*95/10 rule*": at least 95% of the passengers should wait less than 10 minutes for a bus.

The mathematical description of the process is very complex. The clients of the bus service are either passengers arriving in (a) and going to a car park, or passengers arriving in their cars (probably in groups) wishing to go to (d). The arrival processes are correlated: each spot in a car park can be modeled as an on-off process. When cars arrive at the long term parking at the airport they choose the car parks depending on the current car park occupancy, and the holding time corresponds to the number of days that the car owners spend outside Melbourne during each trip. At the end of the trip they will arrive at (a) and request service to go to the car park where they left the car. Trip durations are of the order of days and weeks. This is the slowest time scale for this system. Passenger arrivals (whether at (a) or at the car parks) are the fastest time scale processes. Keeping track of all these correlations presented practical problems and was too complex for analysis and even for simulations.

Although never explicitly stated, there is an implicit optimization problem at hand: the number of buses b should be the minimum required to operate under the 95/10 rule, and the type of bus affects not only

978-1-4799-2076-1/13/$31.00 ©2013 IEEE

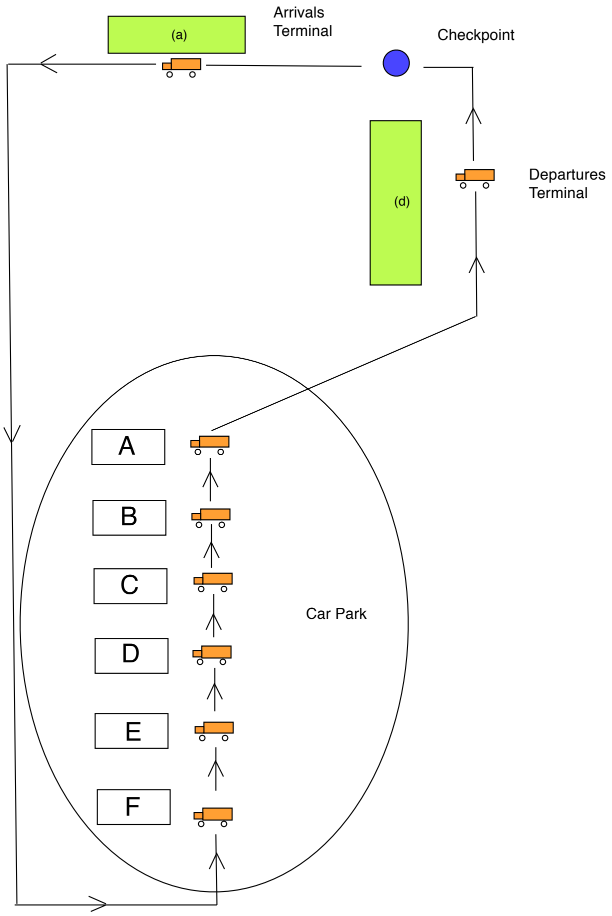

Figure 1: The bus routes in the Melbourne airport.

the dynamics through the capacity and loading/unloading times, but also the operational costs. One must add to the problem the assumption that an "optimal" schedule is followed with the b buses. These may have constraints, such as the regulatory number of continuous hours that every driver can be allowed to do. A margin of extra of buses must also be considered, on account of bus maintenance and regular repairs. Because of the complexity, we decided to use simulation for optimization. Each simulation fixes the type of fleet (there were only three) and evaluates the performance of the system as a function of b, seeking to satisfy the constraint. Due to the high variability of the various processes involved, it is impossible to use deterministic allocation methods for every hour of the day, using input that is noisy.

Inspired in the real problem, we now look at a simplified model where some interesting results can be shown. It is on the basis of this simplified model that we ran simulations and estimated the required fleet characteristics. Our first mathematical challenge was to express the constraint in an unequivocal manner. We did this assuming that the 95/10 rule refers to a steady state operation of the airport, rather than to an actual daily frequency. In order to accelerate the simulation, we used conditioning arguments and defined our data classes so that retrospective conditional expectations can be retrieved for aggregate groups of passengers. Vázquez-Abad and Zubieta (2005) introduced this approach for the simulation of an urban subway system. In that study, the trains were assumed to have infinite capacity, loading is simultaneous (and instantaneous), and people could be modeled as a fluid all along. In contrast, we consider here buses that have limited capacity and passengers must be generated to load and unload the buses one at a time, so we had to adapt the method.

2 STATIONARY MODEL

To obtain a model for the arrival processes of passengers at both the parking lots and the arrivals terminal (a), we assume that the dynamics are stationary. Realistically, business travel and leisure travel have cyclic demands (every week), but in some very busy airports the aggregated demands from numerous trips result in a steady pattern. As an approximation, we derive the following results for the infinite parking capacity model.

Model Assumptions. Cars arrive at the various parking lots according to independent homogeneous Poisson processes with rates $(\lambda_p; p = 1,\dots,P)$. The aggregate arrival process is therefore a Poisson process $N(\cdot)$ with rate $\lambda = \sum_{p=1}^{P} \lambda_p$. Each car will occupy its spot in the parking for the whole duration of the trip of the corresponding car owner(s). This duration is called the *holding time*. We assume that consecutive holding times $\{Y_i, i \geq 1\}$ are iid with a general distribution with finite mean and variance (this distribution can be stratified by travel type, for instance, but the final results will not change if we assume that business and leisure arrivals are independent). At the end of the trip the car owners will arrive at the terminal (a) to pick up their cars.

Lemma 1 As $t \to \infty$ the arrival process at terminal (a) is a Poisson process with rate λ.

Proof. Call $N_a(t)$ the arrival process at terminal (a). Then we have:

$$N_a(t) = \sum_{i=1}^{N(t)} \mathbf{1}_{\{A_i+Y_i \leq t\}},$$

where $(A_i; i \geq 1)$ are the consecutive times of arrivals of cars to the parking (regardless of which lot they choose). Fix t and call $\xi_i = \mathbf{1}_{\{A_i+Y_i \leq t\}}$. Conditioning on the event $\{N(t) = n\}$, the n arrival epochs have the joint distribution of n iid uniform random variables on $(0,t)$, so that $N_a(t)$ has a binomial distribution with parameters (n, β_t), where:

$$\beta_t = \mathbb{P}(\xi_i = 1 \,|\, N(t) = n) = \mathbb{P}(Y_i \leq t - A_i \,|\, N(t) = n) = \frac{1}{t}\int_0^t G(t-s)\,ds = \frac{1}{t}\int_0^t G(s)\,ds,$$

where G is the distribution function of Y. Thus,

$$\mathbb{P}(N_a(t) = m) = e^{-\lambda t} \sum_{n=0}^{\infty} \frac{(\lambda t)^n}{n!} \binom{n}{m} \beta_t^m (1-\beta_t)^{n-m} = e^{-\lambda \beta_t t} \frac{(\lambda \beta_t t)^m}{m!},$$

which shows that $N_a(t)$ has a Poisson distribution with parameter $\lambda \beta_t$ for each time t. Independence of increments follows from the independence of increments of the process $N(\cdot)$ and of the holding times. Thus the departure process is an inhomogeneous Poisson process with rate function $\lambda(s) = \lambda G(s)$. Notice that this is the familiar result for the output of the $M/G/\infty$ queue (Ross 2009). As $t \to \infty$, we use the fact that $\mu \equiv \mathbb{E}(Y) = \int_0^\infty [1 - G(s)]\,ds$ to obtain $\beta_t = 1 - \frac{1}{t}\int_0^t [1-G(s)]\,ds \to 1$, which shows the result. □

A similar technique to the above proof is used in Taylor and Karlin (1998) to derive the stationary occupancy number for the $M/G/\infty$ system. This method can be applied to our model to establish that the aggregate (long term) occupancy numbers of the different car parks are independent Poisson random variables with respective parameters $\lambda_p \mu, p = 1;\dots,P$.

2.1 Notation

There are P different parking lots. We designate the arrivals terminal as station $p = 0$, and the departure terminal as station $p = P + 1$.

$N(t)$: arrival process at the arrivals terminal (a), ∼PoissonP(λ).

$N_p(t)$: arrival process at car park $p \in \{1,\dots,P\}$, $\sim$PoissonP(λ_p) for passengers arriving at (a).
$\bar{N}(t) = N_a(t) + \sum_{p=1}^{P} N_p(t)$: total arrival process for clients.
W_i : waiting time for the bus (in queue) of client i in minutes.
$\pi_p = \mathbb{P}(\text{passenger } i \text{ has destination } p) = \frac{\lambda_p}{\lambda}, \quad p = 1,\dots,P.$
$\mathscr{C}$: Total bus capacity
δ : time required to load/unload each passenger (empirically about 9.75 seconds)
b : total number of buses in the fleet (control variable).
u : headway control variable.
$V_j(p)$: arrival time of bus j at loading dock in station p.
$D_j(p)$: departure time of bus j from station p.
$O_j(p)$: number of people that unload bus j at station p: the *outgoing* passengers.
$T(p)$: travel time between stations $p-1$ and p, for $p = 1,\dots,P$.

2.2 Optimization Formulation

Passengers travel in groups, and the group size distribution is known via empirical data. Although all our results can be developed for a model with random group sizes, the main ideas are easier to develop assuming individual instead of group arrivals, which simplifies the notation, and this is what we do here.

It follows from Lemma 1 that for our model $\lambda_a = \lambda$, so the total arrival rate for $\bar{N}(\cdot)$ is 2λ. Under the assumption that the system is stable (every queue at the stations empties infinitely often w.p.1) the QoS constraint can be stated as:

$$\mathscr{G} = \lim_{H\to\infty} \mathbb{E}\left(\frac{1}{H} \sum_{i=1}^{\bar{N}(H)} \mathbf{1}_{\{W_i > 10\}} \right) \le 0.05\,(2\lambda) = 0.1\,\lambda. \tag{1}$$

In order to state the objective function we first introduce the control variables. We work with a given fleet type that determines $\mathscr{C}$, δ and other operational costs. The stationary model assumes continuous operation of the buses and no attempt is made in the present paper to assign individual drivers to trips. Rather, we look for the optimal bus operation in steady state. Each "solution" to this problem gives rise to specific timetables to be chosen by the management company, which is outside the scope of this paper. If the round trip time RTT of the buses were deterministic, then the *headway* (defined as the mean time between consecutive buses) would be RTT$/b$. However, the round trip time is a random variable that will depend mostly on the unloading times and the loading times of the passengers waiting at the stations. Under fluctuations of the RTT a phenomenon called "bunching" eventually occurs, where an unusually delayed bus is closely followed by an almost empty one. Several measures exist to prevent bus bunching. Because we study a relatively short loop for the routes, the most efficient way to control bunching is to impose a delay that targets a desired headway. At the checkpoint depicted in Figure 1 just before the Arrivals terminal, if bus number j arrives at time V_j and bus $j-1$ left from (a) at time D_{j-1}, then the departure time of bus j is set as $D_j = \max(V_j, D_{j-1} + u)$, so that there are at least u minutes between consecutive buses at the checkpoint. This control rule could be implemented at every station if desired. Instead, after consultation with the Melbourne airport, we use this control rule at the checkpoint only, and at any other station $p \neq 0$ we set:

$$V_j(p) = \max(D_j(p-1) + T(p) + \delta O_j(p), D_{j-1}(p)), \tag{2}$$

so that buses do not overtake each other: if a bus that arrives at a station finds another bus currently loading passengers, then (after unloading) it waits for that bus to finish loading and leave, and then takes its place in the loading area. The time $T(p)$ is a deterministic travel time between stations. (Random travel times pose no difficulties for our simulation model, but it complicates notation and analysis).

Considering our stationary model for the demand of service, given (u,b) the operational costs are defined in terms of the long term average cost per unit time, $J(u,b)$. We assume that this quantity is directly measured from the dynamics of the process and can be estimated accurately with sample averages. The problem becomes

$$\min_{u,b} J(u,b), \quad \text{subject to: } \mathscr{G}(u,b) \leq 0.1\lambda,$$

where $\mathscr{G}$ is defined in (1).

The optimization algorithm that we use is simple. It works under the assumption that $J(u,\cdot)$ is monotonically increasing: more buses imply more costs in the long run. Under this assumption, we first use Fibonacci search to find u_0 such that $\mathscr{G}(u_0,\infty) \leq 0.1\lambda$, and set $b_0 = \infty$. Let $\bar{R}$ denote the maximum round trip time (assumes loading and unloading of full bus). Because the function $\mathscr{G}$ is monotonically increasing in u, the solution to the constrained problem must correspond to an active constraint, so we use stochastic approximation for target tracking, for $n \geq 1$:

$$\begin{aligned} b_n &= \min(b_{n-1}, \bar{R}/u_n) \\ u_{n+1} &= u_n - \varepsilon_n(\mathscr{G}(u_n,b_n) - 0.1\,\lambda) \end{aligned}$$

The fleet size b_n is always initialized to overestimate the actual need for buses. Call u^*, b^* the final value of this iterative procedure. Then decrease b from the current value b^* while $\mathscr{G}(u^*,b) \leq 0.1\,\lambda$. Because the quantity $\mathscr{G}$ is not available in closed form we use simulations to drive the optimization algorithm. Analysis of the optimization procedure and proof of convergence is outside the scope of this work and will be reported elsewhere.

2.3 Infinite Capacity Bus

Fix a control (u,b). The stationary round trip time per bus is a random variable that depends on the total loading and unloading time spent at each station. In this section we analyze the loading time and draw stability results from it.

When a bus arrives at a station it unloads passengers (if any) and then loads passengers waiting at the queue. During this time, more passengers may arrive, and they load as soon as the first group is in the bus. Again, during the loading time of these new passengers, new arrivals may occur, and so on.

Let T_0 be the elapsed time between the departure of the previous bus at station p and the moment when passengers start loading the current bus. The Markov chain $\{X_n; n = 0,1,\ldots\}$ representing consecutive groups of passengers that load the bus is described by:

$$\begin{aligned} X_n &\sim \text{Poisson}(\lambda T_n), \\ T_{n+1} &= \delta X_n. \end{aligned}$$

State $\{0\}$ is the only absorbing state for this chain and the total loading time L is defined by:

$$L = \sum_{n=1}^{\tau} T_n; \quad \text{where } \tau = \min(n : X_n = 0). \tag{3}$$

Call $\mathfrak{F}_n$ the σ-algebra generated by $\{X_n\}$, and notice that $\{X_n\}$ is a non-negative supermartingale only when $\delta\lambda < 1$, because in this case $\mathbb{E}(X_{n+1} \,|\, \mathfrak{F}_n) = \lambda(\delta X_n) < X_n$. This is the condition for stability that ensures $\mathbb{P}(\tau < \infty) = 1$. The expected loading time for this model is calculated using $\mathbb{E}(T_n \,|\, T_{n-1}) = \delta\lambda T_{n-1}$:

$$\mathbb{E}(L \,|\, T_0) = T_0 + \mathbb{E}\left(\sum_{n=1}^{\infty} \mathbb{E}(T_n \,|\, T_{n-1})\right) = T_0 \sum_{n=0}^{\infty} (\delta\lambda)^n = \frac{T_0}{1-\delta\lambda}.$$

Let $T_0 = V_j(p) - D_{j-1}(p)$ be the elapsed time between departure of bus $j-1$ and the start of loading of bus j to station p. The model corresponds to a $M/D/1$ server with vacations of length T_0, whose distribution depends on u and b. Even for this simple model with infinite bus capacity, the solution is intractable analytically. It may be possible to provide approximations to the stationary waiting times using this model, but the QoS of interest is a tail probability, which is much harder to calculate exactly.

2.4 Finite Capacity Bus

Fix a control (u,b) and a total bus capacity $\mathscr{C}$. Consider a station $p \in \{0,\dots,P\}$ where a bus arrives. Let X_n be the number of passengers loading during time T_n as before, and define T_0 as the elapsed time since the last bus departure and the time when people start loading, W_0 as the initial number of people waiting in the station and $C_0 \le \mathscr{C}$ as the initial available capacity (after unloading at p). Now the dynamics are:

$$\begin{aligned} Z_n &\sim \text{Poisson}(\lambda T_n) \\ X_n &= \min(W_n + Z_n, C_n) \\ W_{n+1} &= \max(0, C_n - X_n) \\ T_{n+1} &= \delta X_n \\ C_{n+1} &= C_n - X_n \end{aligned} \tag{4}$$

With these new definitions, the loading time satisfies (3). The three dimensional Markov chain $\{(X_n, W_n, C_n)\}$ has again an absorbing state when $X_n = 0$, which is when no more passengers load. Define $X = \sum_{n=0}^{\tau} X_n$ as the total number of passengers that load the bus ($L = \delta X$), and define:

$$v_{i,k}(t) = \mathbb{E}(X \,|\, T_0 = t, C_0 = i, W_0 = k).$$

The boundary conditions are:

$$\begin{aligned} v_{i,k}(0) &= 0; \quad \forall\, i,k \\ v_{0,k}(t) &= 0; \quad \forall\, t,k \\ v_{i,k}(t) &= i; \quad \forall\, t, \text{ when } k \ge i, \end{aligned}$$

because in the first case there is no capacity left in the bus, and in the second case all i board the bus to fill capacity (leaving the $i-k$ remaining people on the station waiting for the next bus). Let $k < i$, then $k+m$ people will board if $Z = m < i-k$ arrive during the time interval of length t, otherwise i people will board and leave the rest of the arrivals at the station. This yields the multi-dimensional recursion:

$$v_{i,k}(t) = k + (i-k)\bar{F}_{i-k}(t) + \sum_{m=0}^{i-k-1} (m + v_{i-k-m,0}(\delta(k+m)))\, e^{-\lambda t} \frac{(\lambda t)^m}{m!},$$

where $\bar{F}_{i-k}(t) = \mathbb{P}(Z \ge i-k) = e^{-\lambda t} \sum_{m=i-k}^{\infty} \frac{(\lambda t)^m}{m!}$. Because of the model with deterministic service times, time can be discretized in multiples of δ. To solve the recursions, we start with the case $k = 0, t = \delta$, increasing i and t and calculate a table numerically.

In order to find the stability condition, we now require that each queue in the stations empties infinitely often with probability one. The incoming passenger rate is λ_p. During the time that a bus is loading, the service is deterministic, using δ units of time per person. Let Q_n denote the queue size at the start of loading periods, and call $q(t)$ the level of the queue at time $t < T$, where T is the time when the following bus starts loading. Then given the initial bus capacity C:

$$\mathbb{E}(q(t) \,|\, Q_n) = Q_n + \lambda_p t - \min\left(C, \frac{t}{\delta}\right),$$

Suppose that $\lambda_p \delta < 1$, as before. Then the local stability condition at each station $\mathbb{E}(Q_{n+1} \,|\, Q_n) < Q_n$ now reads $\lambda_p \mathbb{E}(T) \le \mathbb{E}(C)$, or:

$$\lambda_p \le \min\left(\frac{1}{\delta}, \frac{\mathbb{E}(C)}{\mathbb{E}(T)}\right)$$

that is, the expected number of arrivals between buses $\mathbb{E}(T)$ should be smaller than the expected available capacity $\mathbb{E}(C)$. Because both T and C depend on (u,b), a closed form solution is not available. We use an approximation to dimension our simulations as follows.

At station $p=0$ (arrivals terminal), the worst case scenario is that all $\mathscr{C}$ seats are occupied. Of these, the number of people that have destination $p \in \{1,\ldots,P\}$ has a Binomial distribution with parameters $(\mathscr{C},\pi_p)$. Also on the worst case scenario, upon unloading at $p-1$ the bus fills again so that the available initial capacity at p satisfies $\mathbb{E}(C) \le \mathscr{C}\pi_p$. The time T between consecutive cycles is such that $\mathbb{E}(T) = \mathbb{E}(\text{RTT} + \tilde{u})/b$, where $\tilde{u}$ is the residual waiting time at the checkpoint to ensure that buses depart at least u time units apart. Notice that $\mathbb{P}(\tilde{u}=0) > 0$ and that $\mathbb{P}(\tilde{u} \le u) = 1$. Again, we find an upper bound with the worst case scenario which is when the loading times are maximal (namely, $\delta\mathscr{C}$). Call $K = \sum_{p=1}^{P+1} T(p)$ the travel time. Because there is loading and unloading,

$$\frac{K}{b} \le T \le \frac{K + 2(P+1)\delta\mathscr{C} + u}{b}, \quad \text{with prob. 1}$$

where we have defined K as the total travel time, so that a sufficient stability condition is:

$$\lambda_p \le \min\left(\frac{1}{\delta}, \frac{b\mathscr{C}\pi_p}{K}\right).$$

3 SIMULATION MODEL

3.1 Local Model at Stations

Consider the dynamics at a local node, namely a station p. The station keeps the (local) time $D_{j-1}(p)$. In the code, this "clock" is a local clock at the station. Unloading is performed upon arrival of bus j. Loading is programmed following (4), that is, by "groups" (indexed by n) as will be described later.

Assuming that the arrivals at station $p=0$ are all independent Poisson processes with rates $\{\lambda_p, p = 1,\ldots,P\}$, then given N_1 initial passengers in bus j the number that debark at station $p=1$ has a Binomial distribution $\text{Bin}(N_1,\pi_1)$. If $N_2 \le N_1$ is the number of passengers left in the bus, then the number debarking at station $p=2$ has a Binomial distribution with parameters $(N_2,\tilde{\pi}_2)$, where $\tilde{\pi}_2 = \pi_2/(1-\pi_1)$, and in general, if there are N_p passengers in the bus that boarded at (a) when the bus arrives at station p, then the number of outgoing passengers that unload $O_j(p) \sim \text{Bin}(N_p,\tilde{\pi}_p)$, with $\tilde{\pi}_p = \pi_p/(\sum_{\ell=p}^{P}\pi_\ell)$.

Stations represent the queue as a linked list of "passenger groups". Each group $g_n = (P_n, I_n)$ has an integer P_n denoting the number of people in the group, and a time interval I_n that records the interval where the P_n people arrived at station p. The first group in queue g_1 has the earliest arrival interval (people in this group have been waiting longer) and will be the first to load.

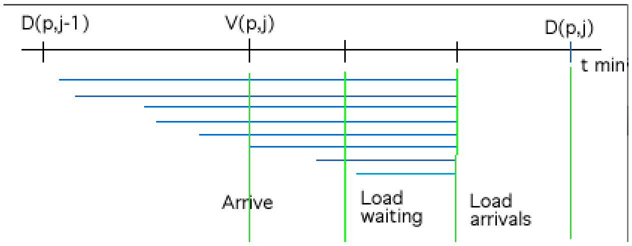

Figure 2: Time line at station p: during the time that customers waiting board the bus, there may be other arrivals which will be loaded next, and during this time, there may be other arrivals, until either the bus is full to capacity or no more people are left in the queue.

The algorithm for loading initializes the elapsed time $\Delta T_0 = V_j(p) - D_{j-1}(p)$ as the time from the previous bus departure until loading starts, where $V_j(p)$ and satisfies (2). The ghost simulation model generates only this slower time scale of bus dynamics. Individual passenger arrivals are not generated, but rather only one random variable is produced: $Z_0 \sim \text{Poisson}(\lambda_p \Delta T_0)$. This represents the aggregate number of arrivals during this time interval. These new arrivals are placed as a group at the end of the list, say in position N, with $P_N = Z_0$, and $I_N = (t, t + \Delta T_0)$, where t is the local station clock, initialized at $t = D_{j-1}(p)$. Then t is updated to $t + \Delta T_0$ and loading can start.

Refer to Figure 2: bus j arrives at station p at time $V_j(p)$ and it commences unloading. After unloading, the people waiting in the station start loading. But during the elapsed time there may be more arrivals (their individual waiting times until loading are represented by lines). These must be loaded afterwards. In our code loading is done recursively by groups: start at $n = 1$ at the front of the queue. While there is available capacity in the bus ($C_n > 0$) and there are people waiting, if $P_n \leq C_n$, then all P_n load (they are removed from position 1 in the list). If $P_n > C_n$, then C_n people load and the remaining ones are put back in position 1 in the list. Every time a group loads (totally or partially) we update capacity, clocks and and generate new arrivals:

$$
\begin{aligned}
\Delta T_n &= \delta \times \min(P_n, C_n), \\
Z_n &\sim \text{Poisson}(\lambda_p \Delta T_n), \\
C_{n+1} &= \max(0, C_n - P_n).
\end{aligned} \tag{5}
$$

These new arrivals are placed at the end of the list, say position $N+1$, with $P_{N+1} = Z_n$, and $I_{N+1} = (t, t + \Delta T_n)$, where t is the local clock variable. Then t is updated to $t + \Delta T_n$.

3.2 Conditional Monte Carlo Statistics

In this section we describe how the ghost model can accelerate the simulation, and provide the basis for variance reduction. The gain in simulation time is mostly due to advancing the clock at the station between bus departures and arrivals. In our original ghost model (Vázquez-Abad and Zubieta 2005) passengers were treated as a fluid, only considering expected values. Because of limited bus capacity in this problem we have added the recursive group generation while loading takes place. It slows down the algorithm, but not very much because typically there are very few iterations in this recursion. Generation of passengers is a retrospective simulation and arrivals are aggregated in lumps.

The function $J(u,b)$ can be directly estimated from the ghost model without bias. This is because the operating costs depend on average bus utilization, which depends on the bus dynamics only. This is the slower time scale that is accurately simulated in the ghost model.

The challenge is the estimation of the function $\mathscr{G}(u,b)$ which is a quantile of the passenger wait times. There are various models that can be considered for the individual waiting time. The first is the actual wait for each individual, from arrival to loading the bus; or from arriving to the moment when the bus leaves the station. The personnel at the Melbourne airport mentioned that the waiting time was mostly related to the time when the bus arrives at the station to load the passengers, and that loading times are not counted. Apparently the psychological reaction to wait is more important when the vehicle is not yet in sight. We followed this model in our program. For other models of the waiting time appropriate corrections may be required.

The main results in this section show that, although the model for the simulation aggregates passengers into groups, the final statistics is *consistent* for $\mathscr{G}(u,b)$.

Theorem 1 Let $g_n = (P_n, I_n)$ be a group loading such that $P_n \sim \text{Poisson}(\lambda_p |I_n|)$ and assume that $P_n \leq C_n$. Let W_n^* be the number of people in group g_n that wait more than 10 minutes for the bus. Conditioning on $I_n = (t_1, t_2)$,

$$
\mathbb{E}(W_n^* \,|\, I_n) = \begin{cases} P_n \left(\frac{t - 10 - t_1}{t_2 - t_1} \right) & \text{if } t_1 < t - 10 < t_2, \\ P_n & \text{if } t_2 < t - 10. \end{cases} \tag{6}
$$

Proof. Figure 3 shows a scheme of the relative placement of times. The group arrived during an interval in the past. If $t_2 < t-10$ then all waited more than 10 minutes. If $t_1 > t-10$ then none wait more than 10 minutes.

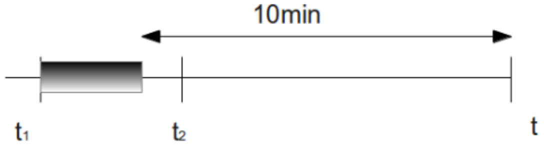

Figure 3: Timeline for interval I_n and $t-10$.

Suppose now that $t_1 < t-10 < t_2$. Given P_n arrivals in time interval (t_1,t_2), the individual arrival epochs $(T_i; i=1\ldots,P_n)$ have a joint distribution of the order statistics of iid uniform random variables on (t_1,t_2). That is, $T_i \overset{d}{=} t_1 + (t_2-t_1)\times\text{Beta}(i,P_n-i+1)$. Call $b_i = (T_i-t_1)/(t_2-t_1) \sim\text{Beta}(i,P_n-i+1)$ and $x = (t-10-t_1)/(t_2-t_1)$. The number of people that wait more than 10 minutes is:

$$\mathbb{E}(W_n^* \,|\, I_n) = \mathbb{E}\left(\sum_{i=1}^{P_n} \mathbf{1}_{\{T_i \le t-10\}} \,|\, P_n\right) = \sum_{i=1}^{P_n} \mathbb{P}(T_i \le t-10) = \sum_{i=1}^{P_n} \mathbb{P}(b_i \le x) = \sum_{i=1}^{P_n} I_x(i,P_n-i+1)$$

where $I_x(\alpha,\beta)$ is the incomplete Beta function. In the case of integer arguments, it simplifies to:

$$\mathbb{E}(W_n^* \,|\, I_n) = \sum_{i=1}^{P_n}\left(\sum_{j=i}^{P_n}\binom{P_n}{j}x^j(1-x)^{P_n-j}\right) = \sum_{i=1}^{P_n}\mathbb{P}(X_x > i) = \mathbb{E}(X_x) = xP_n,$$

for $X_x \sim \text{Bin}(P_n,x)$. Putting results together, this proves the claim. □

In our code, it is when people load the bus that we record the statistics to estimate $\mathscr{G}(u,b)$. If $P_n < C_n$ then we use Theorem 1 to update statistics. When $P_n > C_n$ only C_n passengers will load the bus, and thus we add to the tally the number $\min(\mathbb{E}(W_n^*),C_n))$. The remaining $P_n - C_n$ passengers are re-inserted in the linked list of the queue in first position. Suppose that they keep the original interval I_n. This will be the first group to load the following bus. If the time between buses is large enough and the loading time is now $t' >> t$ then at loading time it will be certain that all of them waited more than 10 minutes ($t_2 < t'-10$) and the statistics (6) will not be biased. However, in the case that the next bus is ready for loading immediately (which may happen with non zero probability) or very soon, (6) may provide a biased estimate. The reason for this is that the people that are left waiting are always *the last ones* that arrived in (t_1,t_2), because of the FCFS policy loading the bus in order of arrivals. The correction that we make approximates the arrival interval of the remaining customers by setting:

$$t_1' = t_1 + \frac{C_n}{P_n+1}(t_2-t_1),$$

which is the expected arrival time of passenger C_n. Using this approximation, we update $I_n = (t_1',t_2)$ when the $P_n - C_n$ passengers are reinserted in the queue. Although there is a source of bias for these (rare) groups of passengers due to the non-lienarity of (6) in t_1, the effect is very small.

Simulation efficiency is defined in terms of both precision and speed of execution, as explained by Glynn and Whitt (1992). Because our statistics are based on conditioning, the local model is an example of conditional Monte Carlo. While variance reduction by conditioning can be ensured for simple random variables (Ross 2012), the case of a Markov process is not straightforward. Given N groups loading, our statistics are of the form

$$\mathbb{E}\left(\sum_{n=1}^{N} W_n^*\right) = \mathbb{E}\left(\sum_{n=1}^{N}\mathbb{E}(W_n^* \,|\, \mathfrak{F}_n)\right),$$

where $\mathfrak{F}_n = \sigma(g_n, C_n, \Delta T_n, Z_n)$, from (5). Although each term does satisfy $\mathbb{V}\text{ar}(W_n^*) \geq \mathbb{V}\text{ar}(W_n^* \,|\, \mathfrak{F}_n)$, the contribution of covariances in the sum may prevent variance reduction for the estimation of $\mathscr{G}(u,b)$. However, it is possible to show here that the correlation between groups that board different buses is very small.

3.3 Increasing Efficiency

The computational effort required to perform the local dynamics is indeed smaller than generating every arriving passenger and keeping a detailed list of events for unloading and loading. In order to further speed up the computation, we have implemented a simplified version that approximates $\mathbb{E}(W_n^* \,|\, \mathfrak{F}_n)$. In this implementation, instead of keeping an interval I_n, a single aggregate number is recorded, namely $\tau_n = (t_1 + t_2)/2$ which is the average expected arrival time of the passengers in group g_n. This way, the linked list representing the queue is simpler, each group having attributes $g_n' = (P_n, \tau_n)$. When loading at time t we update the statistics using:

$$\mathbb{E}(W_n^* \,|\, g_n) \approx \tilde{W}_n^* \stackrel{\text{def}}{=} P_n \mathbf{1}_{\{\tau_n < t-10\}}, \tag{7}$$

and so we increase the statistics for the estimation of $\mathscr{G}(u,b)$ adding $\min(\mathbb{E}(W_n^*, C_n))$. If $P_n > C_n$ then we put back the remaining customers in the front of the queue, this time with the same value of τ. This means that we count *all* of the passengers that load only when their average wait is more than 10 minutes.

Theorem 1 Consider the time t when passenger group $g_n = (P_n, I_n)$ boards the bus, with $I_n = (t_1, t_2)$, and let $\tau_n = (t_1 + t_2)/2$. Let $y = t - 10$ and $\tilde{W}_n^*$ be defined as in (7). Assuming that $P_n \leq C_n$, then $\mathbb{E}(W_n^* \,|\, \mathfrak{F}_n) = \mathbb{E}(\tilde{W}_n^* \,|\, \mathfrak{F}_n)$.

Proof. If $y \leq t_1$ then $y < \tau_n$ and all P_n wait more than 10 minutes, so $W_n^* = \tilde{W}_n^* = P_n$ in this case. If $y \geq t_2$ then also $y > \tau_n$ and again $W_n^* = \tilde{W}_n^* = 0$. Given $y \in I_n$, there is no subinterval of I_n that is more likely to contain the point y, because t is determined by the bus dynamics and previous customer groups loading. Therefore, y has uniform distribution on I_n. This yields

$$\mathbb{E}(\tilde{W}_n^* \,|\, P_n, y \in I_n) = P_n \mathbb{E}(\mathbf{1}_{\{y > \tau_n\}}) = P_n/2.$$

On the other hand, using (6),

$$\mathbb{E}(W_n^* \,|\, \mathfrak{F}_n, y \in I_n) = \frac{P_n}{(t_2 - t_1)} \int_{t_1}^{t_2} \frac{u - t_1}{(t_2 - t_1)} \, dx = \frac{P_n}{2},$$

proving the claim. □

As with the previous model, when $P_n \leq C_n$ the statistics is unbiased, but when $P_n > C_n$ only C_n board and the rest join the queue awaiting for the next bus to load. If this is the case then we add to the tally $C_n \mathbf{1}_{\{\tau < t-10\}}$ and we must put back in first place the remaining customers. Because we do not have now the interval I_n but only the midpoint τ_n, we can't use the same correction to define g_n'. Let us assess the source of bias from these (rare) groups that fill the buses with $g_n' = (P_n - C_n, \tau_n)$. If $\tau_n < t - 10$ then we use Lemma 2 to establish unbiasdness: here all P_n should be added to the tally. Those that have to wait for next bus will also count. Consider now the case $\tau_n > t - 10$. None of the C_n boarding will be counted for the statistics, however the remaining customers will wait additional ΔT minutes for the next bus loading time, and this may make them wait more than 10 minutes. Let $I_n' = (t_1, t_2)$ be the actual time when the remaining $P_n - C_n$ customers arrived (unknown), and $t' = t + \Delta t$ the time when next loading starts. Use again $y = t - 10$. If $y + \Delta t \in (0, t_1)$ then all points in I_n' have a wait of less than 10 minutes at time t', and no passenger should count. However when $y + \Delta t \in (\tau_n, t_1)$ the algorithm will add $(P_n - C_n)$ to the count because $\tau_n < t + \Delta t - 10$. The bias is here $P_n - C_n$. If $\Delta t > t_2 - y$ then all $P_n - C_n$ passengers should count for the wait, as the algorithm does, with no bias. Finally, if $y+\Delta t \in I'_n$ then using the same argument as in the proof of Lemma 2, the expected number that wait more than 10 minutes is $(P_n - C_n)/2$. In summary, only when $\tau_n > y$ and $P_n > C_n$, the expected bias in the count is:

$$(P_n - C_n)\,\mathbb{P}(y+\Delta t \le t_1) + \frac{(P_n - C_n)}{2}\,\mathbb{P}(y+\Delta t \in (t_1,t_2)).$$

We cannot evaluate these probabilities exactly using only τ_n. The choice between using the simplified model $g'_n = (P_n, \tau_n)$ or the full model $g_n = (P_n, I_n)$ depends on the trade off between speed and accuracy.

3.4 Ghost Model Structure

The local model described can be coded within a discrete event simulation where the event list contains all bus movements; namely arrivals and departures from stations. However, managing list searches can always slow down the execution of a simulation. Instead, we maximize efficiency exploiting the modularity of the system. Because buses do not overtake each other, we can program the whole system looking at each bus at a time: the process seen by a bus requires no anticipative knowledge of buses that start their trip later. We need only keep the appropriate clocks at the stations and implement the unloading and loading functions at the right times. Table 1 summarizes the classes for our code, written in JAVA.

Table 1: Classes for the ghost model.

Class	Variables	Methods
GhostSystem (main)	all global variables	checkpoint, Stats
Bus	C_n, N_p, local clock	Load/unload groups, add_riders
Station	t (local clock), $D_j(p)$, LinkedList (queue)	PassArrivals, List_mgmt
PassengerGroup	$g_n = (P_n, \tau_n)$	–

The main program executes an outer loop for buses around an inner loop of stations. Logically, each bus executes its round trip from the Arrivals terminal (a) to the Departures terminal (d). The bus receives an initial departure time from the Arrivals terminal and the GhostSystem uses this time to compute all time dependent trip calculations. Each station p is aware of the time count for bus j and it keeps the consecutive departure times $D_j(p)$. GhostSystem uses (2) when a bus arrives at a station: the bus provides the clock $D_j(p-1)+T(p)$ and $O_j(p)$, and the station p provides the information $D_{j-1}(p)$. This keeps station clocks updating in a forward direction even though system wide time shifts depending on the active bus.

Interestingly, the simulation is performed constantly going forward and backwards in time. Indeed, not only do we perform retrospective simulation every time that buses load passengers (to generate arrivals), but also when the outer loop of buses is executed, say for bus j's trip, with high probability bus $j+1$ will have a starting time which is before the time that bus j has finished its round trip.

In addition to the local dynamics as explained before, we keep a "parking bay" (a queue) for unused buses. When b is very large, there may be unused buses that should be called for only when the time u is elapsed from a departure and no other bus has finished its round trip.

4 CONCLUDING REMARKS

This paper presents the extension of a methodology that works well when fast Poisson processes can be aggregated for the simulation. Although it is well known that conditioning on time intervals, the expected number of arrivals is proportional to the rate, this approximation would result in significant bias when loading "fluid" passengers into buses. To remedy the situation we resort to the generation of aggregated groups of passengers. Using conditioning arguments, it is possible to simplify the required bookkeeping and calculations to a minimum, thus speeding up the simulation.

For the problem of quantile estimation, we have identified the source of bias and provided a methodology can that be used to treat other performances. The trade off between accuracy (bias) and speed of execution will highly depend on the problem at hand. Naturally very large networks with very different time scales may benefit much more from the ghost model than networks with similar time scales.

The program is "linear", looping each bus at a time, and very easy to code. Perhaps the most challenging part of the code is taking account of the various clocks at the bus depot, when bus rides must be started. Although our current program depends highly on the specific structure of the problem where buses do not overpass each other, several modifications can be made to fit other problems.

REFERENCES

Glynn, P. W., and W. Whitt. 1992. "The Asymptotic Efficiency of Simulation Estimators". *Operations Research* 40:505–520.

Ross, S. M. 2009. *Introduction to Probability Models*. 10th ed. Academic Press.

Ross, S. M. 2012. *Simulation*. 5th ed. Academic Press.

Taylor, H., and S. Karlin. 1998. *An Introduction to Stochastic Modeling*. 3rd ed. Academic Press.

Vázquez-Abad, F., and L. Zubieta. 2005. "Ghost Simulation Model for the Optimisation of an Urban Subway System". *DEDS Journal* 15 (3): 207–235.

AUTHOR BIOGRAPHY

FELISA VÁZQUEZ-ABAD (felisav@hunter.cuny.edu) is Professor of Computer Science at Hunter College of the City University New York. She is Executive Director of the CUNY Institute for Computer Simulation, Stochastic Modeling and Optimization that she helped to create in 2013. She obtained a B.Sc. in Physics in 1983 and a M.Sc. in Statistics and Operations Research from the Universidad Nacional Autónoma de Mxico. In 1989 she obtained a Ph.D. in Applied Mathematics from Brown University. She spent four years doing postdoctoral research at Brown University and later at the INRS-Telecommunications in Montreal, Canada. She became a professor at the University of Montreal, Canada in 1993, where she remained until 2004 when she became a professor at the University of Melbourne, Australia, until 2009. In 2000, she was a recipient of the Jacob Wolfowitz award for advances in the mathematical and management sciences. Her interests focus on the optimization of complex systems under uncertainty, primarily to build efficient self- regulated learning systems. She has applied novel techniques for simulation and optimization in telecommunications, transportation, finance and insurance and she is interested by real life problems. She co-authored a US patent for an optical network switch and was research consultant to the Melbourne Airport. She has participated in Grant Selection Committees and has been Associate Editor for IEEE Transactions on Automatic Control, Management Science, and Operations Research Letters, and Area Editor of the ACM Transactions on Computer Modeling and Simulation. She has been web editor of the INFORMS College on Simulation since 2000.