Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

NEW HISTORY-BASED DELAY PREDICTORS FOR SERVICE SYSTEMS

Mamadou Thiongane
Wyean Chan
Pierre L'Ecuyer

Department of Computer Science and Operations Research
Université de Montréal
2920, chemin de la Tour
Montréal Québec, H3C 3J7, CANADA

ABSTRACT

We are interested in predicting the wait time of customers upon their arrival in some service system such as a call center or emergency service. We propose two new predictors that are very simple to implement and can be used in multiskill settings. They are based on the wait times of previous customers of the same class. The first one estimates the delay of a new customer by extrapolating the wait history (so far) of customers currently in queue, plus the last one that started service, and taking a weighted average. The second one takes a weighted average of the delays of the past customers of the same class that have found the same queue length when they arrived. In our simulation experiments, these new predictors are very competitive with the optimal ones for a simple queue, and for multiskill centers they perform better than other predictors of comparable simplicity.

1 INTRODUCTION

Delay estimation and its announcement to customers that join a queue in a service system can be used to improve their experience and the overall efficiency of the system. For example, in Canada and the USA, some emergency rooms display the average wait time of recent patients over the internet or on an electronic dashboard. This information may help reduce peak congestion by encouraging incoming patients to visit a less busy emergency facility. In telephone call centers, the waiting queue is generally invisible to the caller, unlike physical queues in department stores and supermarkets. Forecast delay announcements can then provide very useful information to callers. Upon hearing the estimated delay, a customer may choose to hang up, or stay in the queue, or opt to be called back later (Armony, Shimkin, and Whitt 2009). In this paper, we focus solely on delay prediction, without considering the impact of its announcement on customer behavior.

Most previous work on delay prediction has been made for systems with a single *first-come, first-served* (FCFS) waiting queue with a single class of customers, for which simple analytical formulas can be derived for various performance measures. Ibrahim and Whitt (2009a) distinguish two categories of predictors: those that depend mainly on the length of the queue and system parameters, named *queue-length* (QL) predictors, and those that rely primarily on past delay information, named *delay-history* (DH) predictors. One very simple and popular DH predictor is the *last-to-enter-service* (LES) predictor, which just returns the wait time experienced by the last customer to have started service. The main difference between QL and DH predictors is that QL predictors are generally derived from queueing theory, whereas DH predictors are usually parameter-free heuristics based on historical observations.

QL predictors are known to be optimal for a simple system such as a single M/M/s queue in steady-state, but they do not apply to multiskill systems and would be hard to extend to systems with mutiple queues.

Very few predictors have been proposed for queueing systems with multiple types of customers and multiple queues that can share some servers, as in multiskill call centers, in which each server is an agent that can handle a subset of the call types, and each call type has its own queue. These types of predictors generally apply machine learning algorithms over the observed data. More precisely, they use optimisation methods to learn the parameters of the predictors. Senderovich et al. (2015) proposed predictors for a multiskill call center with multiple call types but a single type (or group) of agents that can handle all call types. Thiongane, Chan, and L'Ecuyer (2015) studied two classes of delay predictors that can be used for more general multiskill call centers or service systems. One uses *regression splines* (RS) and the other uses an *artificial neural network* (ANN). These predictors use the delay information of the LES customer and the size of the queues. Ang et al. (2016) study the Lasso method (Tibshirani 1999) for the wait time prediction in emergency departments based on certain state variables of the system (such as the queue length) and functions of them. These predictors perform well empirically in simulations, but one drawback is that they have a large number of parameters that must be "learned" beforehand. This training phase of the model requires a large amount of data and computational time. They are also more complex to implement in practice.

In this paper, we focus on simple DH-type predictors that are easy to implement and have very few parameters. We propose two new predictors. The first one extends the LES predictor by also considering the wait time experienced thus far by the customers of the same type that are *still in the queue*. The final wait times of those customers are still unknown, but the predictor extrapolates the wait times realized so far. We call this predictor the *extrapolated LES* (E-LES). The second predictor estimates the wait time of the new customer by a moving average of the realized wait times of the customers of the same type who found the same queue length when they arrived. We call it the *average conditional LES* (AvgC-LES). These new predictors are attractive largely because of their simplicity. They have very few parameters, do not need an optimisation phase, and are easy to implement in practice. The second one has a single parameter: the window size for the moving average. The first one has none in its basic form, whereas some of its variants have one parameter, which serves to exclude some of the customers at the back of the queue, whose realized wait time so far does not provide sufficient information. We study these predictors in the context of call centers, but they can also be used for other service systems.

We performed simulation experiments to compare the accuracy of different predictors on different call center models. In those experiments, we found that AvgC-LES was usually more accurate than E-LES, which was in turn more accurate than LES. For a single queue, for which QL is known to be optimal, AvgC-LES came very close to QL. For the multiskill call center examples, AvgC-LES was slightly less accurate than RS and ANN. However, it is much simpler.

The remainder of the paper is structured as follows. Section 2 presents a literature review of delay predictors for service systems, with a particular focus on call centers. Section 3 introduces our new delay predictors and describes other predictors considered in our numerical comparisons, reported in Section 4. A conclusion is given in Section 5.

2 LITERATURE REVIEW

As mentioned in the introduction, most work on delay prediction has been done for single-queue systems, in which customers are answered in FCFS order, so that future arrivals do not affect the wait time of present customers. Theoretical results on delay predictors are limited almost exclusively to these simple systems. In multi-queue systems, customers may have different priorities and FCFS may not hold in general.

Consider a new customer who enters a queue with C customers already waiting in that queue, and let W be the random variable that represents her wait time. A natural predictor of W is its expectation conditional on C. This is the QL predictor. For a GI/M/s queue with general inter-arrival time distribution, exponential service times with rate μ, s servers, and no abandonment (infinite customer patience), this conditional expectation is (Whitt 1999):

$$\mathbb{E}[W \mid C] = \frac{C+1}{s\mu}.$$

In a GI/M/s+M queue, each customer has an exponential patience time with mean ν^{-1}, after which she abandons if still waiting in queue. The expected virtual delay conditional on C is then

$$\mathbb{E}[W \mid C] = \sum_{c=0}^{C} \frac{1}{s\mu + c\nu}. \tag{1}$$

The virtual delay is the wait time that a customer must wait before she is answered. If the customer abandons, then her virtual delay is the wait time she would have needed to wait before receiving service. If we select only the customers who did not abandon, their average waiting is slightly less. Jouini, Dallery, and Aksin (2011) give the conditional expected wait time of a *served* customer who finds C customers already in queue upon arrival:

$$\mathbb{E}[W \mid C \text{ and served}] = \sum_{c=1}^{C+1} \frac{1}{s\mu + c\nu}. \tag{2}$$

Equation (2) is preferable to (1) if we are only interested in the prediction errors of served customers, which is the case in this study.

QL formulas are much harder to develop for multi-queue systems, because servers may be restricted to serve a subset of customer types, and they could attribute different priorities to different customer types. For the special case where every server can serve all customer types with the same priority order, Senderovich et al. (2015) propose QL formulas that give upper and lower bounds on the expected delay time.

Ibrahim and Whitt (2009a) and Ibrahim and Whitt (2011) study several DH predictors in a single-queue environment. When a new customer enters the queue, these predictors return the delay experienced by the *last-to-enter-service* (LES) customer, *head-of-line* (HOL) customer, *last-to-complete-service* (LCS) customer, or most *recent arrival to complete service* (RCS). LES and HOL predictors are generally better than LCS and RCS. The latter predictors are based on the delay times of customers who have already completed service, so they use older information than LES and HOL. Ibrahim and Whitt (2009b), Armony, Shimkin, and Whitt (2009) and Ibrahim, Armony, and Bassamboo (2016) study extensively the LES predictor for a single queue, and they show that LES is asymptotically accurate (and optimal) as the numbers of servers and customers grow to infinity. Ibrahim, Armony, and Bassamboo (2016) also propose different adjustments to the LES predictor. For example, the delay estimation could be adjusted proportionally based on the current queue length and the queue length observed by the LES customer when it arrived.

A simple generalization of LES, often used in practice (Dong, Yom Tov, and Yom Tov 2016), takes the average wait time of the last N customers who entered service, or of the customers who entered service in the last T units of time. Ibrahim, Armony, and Bassamboo (2016) have observed that this type of averaging usually does not improve the accuracy (and may degrade it) compared with the original LES, because the average version uses older information. We also observed this in our numerical experiments.

Another class of predictors proposed recently use machine learning algorithms (e.g., artificial neural networks, regressions, or decision trees) to train the delay predictors; see for example Senderovich et al. (2014), Senderovich et al. (2015), Thiongane et al. (2015), and Ang et al. (2016). These algorithms are generally more accurate than DH predictors, but they are also more complex to implement in practice, and they require a large amount of data and training time. For these reasons, there is still interest for simpler methods.

3 DELAY PREDICTORS

In this section, we define the specific delay predictors considered in this paper. Since we are interested in predictors that are likely to be implemented in practice, we consider primarily DH predictors having

few parameters to optimize. For comparison, in our simulation studies we also include QL predictors for single-queue systems, and machine learning algorithms for multiple queues and multiskill instances. Note that even though they may perform better (when they are applicable), these predictors have other limitations, as we said earlier. The DH estimators always use only the delays of customers of the same class (same queue) as the one for which we are making the prediction. Thus, for each method considered, there is a different predictor for each customer class j, even if we do not always index it by j explicitly.

3.1 Last-to-Enter-Service (LES)

This simple predictor returns the wait time experienced by the last-to-enter-service customer of the same class who had to wait (Ibrahim and Whitt 2009a). If the LES customer did not wait in queue (its delay time was zero), this predictor will return the wait time of the most recent customer who had a positive delay.

3.2 Average LES (Avg-LES)

This *averaging* version of the LES predictor is often used in practice (Dong, Yom Tov, and Yom Tov 2016). For class j, it returns the average delay experienced by the N_j most recent customers who entered service after waiting a positive time, for a fixed integer $N_j > 0$, or by the (variable number of) customers who entered service in the last T_j units of time. A larger N_j or longer time window T_j increases the smoothness and may thus reduce the variance of the predictor, but this larger lag most often results in less accurate predictions because it uses older (less relevant) information. In particular, the predictions are more likely to be based on the waits of customers who saw a very different queue ahead of them when they arrived. The N_j or the T_j can be taken as all equal, but it could also make sense to take larger N_j or smaller T_j for more frequent classes of customers. In our experiments, we found that the best choice of N_j was usually $N_j = 1$.

3.3 Weighted Average LES (WAvg-LES)

Avg-LES can be generalized to a *weighted average* of the past wait times. For each queue (customer class) j, we select a sequence of non-negative weights $\phi_{j,1}, \phi_{j,2}, \ldots$, usually non-increasing and converging to 0, and such that $\sum_{i=1}^{\infty} \phi_{j,i} = 1$. Then we predict the wait time of an arriving customer by

$$D_j = \sum_{i=1}^{\infty} \phi_{j,i} W_{j,i},$$

where $W_{j,i}$ is the wait time of the ith-last customer of class j that started service (the LES for $i = 1$, the previous one for $i = 2$, etc.).

By taking $\phi_{j,i} = 1/N_j$ for $i = 1, \ldots, N_j$ and $\phi_{j,i} = 0$ for $i > N_j$, we recover Avg-LES. If we take $\phi_{j,i} = \alpha_j(1-\alpha_j)^{i-1}$ instead, for some smoothing factor $\alpha_j \in (0,1]$, we obtain an *exponential smoothing average* (ESAvg-LES) instead of an ordinary average. For $\alpha_j = 1$, we recover LES. For $\alpha_j < 1$, the implementation must be approximate, because in practice we only have a finite number of past delays. In our implementation, for each class j, we initialize a predictor S_j to -1, and we update S_j as follows. Each time a new customer of this class starts service after a wait time W, we set S_j to W if $S_j = -1$, otherwise we update it to

$$S_j := \alpha_j W + (1-\alpha_j) S_j.$$

When a customer of class j enters the queue, its wait time is predicted by the current S_j. If $S_j = -1$, we return the value of the LES predictor. This predictor has many parameters (the weights) in its general form, but this large number of parameters can be easily reduced by putting constraints on the weights. According to our experiments, the best choice of α_j is usually close or equal to 1.

3.4 Proportional Queue LES (P-LES)

Let Q_{LES} denote the number of customers in queue when the LES customer arrived, Q the number of customers in queue ahead of the new arrival, and x the LES delay. To account for the change in queue length, Ibrahim, Armony, and Bassamboo (2016) consider (as a heuristic) a predictor that multiplies x by the ratio Q/Q_{LES}. Here, we modify this predictor by adding the end of service to free a server, as in the QL formulas. This gives

$$D = x\frac{Q+1}{Q_{\text{LES}}+1}.$$

This modification also resolves the case in which the LES customer entered an empty queue ($Q_{\text{LES}} = 0$).

3.5 Extrapolated LES (E-LES)

Here we propose a DH predictor that relies on delay information from the customers who are *currently waiting in queue*. The final delay times of these customers are still unknown, but we extrapolate the elapsed (partial) delays to predict them. This is the main distinction between E-LES and the previous DH predictors (LES, Avg-LES, WAvg-LES and P-LES), which rely only on past complete delay times. E-LES uses partial but fresher information. It works as follows.

Suppose a new customer enters a queue with C customers ahead, numbered from 1 to C, with customer 1 at the head of the queue. The LES customer, who was just in front of customer 1, has the number 0. For any customer $c \in \{1,\dots,C\}$, let $Q(c)$ be the number of customers *already in queue* when customer c arrived, $A(c)$ be the number of customers *currently* ahead of c, and $W(c)$ be the wait time experienced by customer c up to now. Thus, customer c found $Q(c)$ customers in the queue upon arrival, and has advanced by $Q(c)-A(c)$ positions in the queue during the elapsed time $W(c)$ since its arrival. Since $Q(c)+1$ customers must exit the system (after being served or have abandoned) before customer c can begin service, it seems natural to predict the wait time $E(c)$ of customer c by the linear extrapolation

$$E(c) = W(c)\frac{Q(c)+1}{Q(c)-A(c)}. \tag{3}$$

For $c = 0$, we set $E(0)$ equal to the real wait time of the LES customer, because her true delay is already known.

The predicted delay D of the new customer is the average of the extrapolated delays of the C customers in queue and the true delay of the LES customer:

$$D = \frac{1}{C+1}\sum_{c=0}^{C} E(c). \tag{4}$$

Formula (4) based on (3) provides a natural weighted average that puts more weights on more recent delays, so one might hope that it captures the changes in system dynamics earlier than other DH predictors. Note that because the C customers share the same queue, their wait times are generally correlated, so the $E(c)$ are not independent. One weakness of predictor (4) is that customers near the end of the queue have experienced only short wait times so far and typically have a small value of $Q(c)-A(c)$, hence they are likely to provide less delay information and their extrapolated delays $E(c)$ are usually noisy. To reduce this noise, we can add multiplicative weights that decrease with c, as in WAvg-LES. Those weights may depend on C, c, $Q(c)$ and $A(c)$.

We have implemented one version of this that selects a threshold parameter τ and includes in (4) only the customers who advanced by at least τ positions in the queue since their arrival. That is, we define $\mathscr{C} = \{c \le C : Q(c)-A(c) \ge \tau\} \cup \{0\}$ and

$$D = \frac{1}{|\mathscr{C}|}\sum_{c\in\mathscr{C}} E(c), \tag{5}$$

where $|\mathscr{C}|$ is the size of set $\mathscr{C}$. The weights are $1/|\mathscr{C}|$ for $c \in \mathscr{C}$ and 0 otherwise. In our implementation, we select a fixed constant $\delta > 0$ and we set dynamically the threshold τ as a proportion δ of the current queue length C, that is $\tau = \lceil \delta C \rceil$. A larger δ will return predictions closer to those of LES. In particular, if we set $\delta = \infty$, then $\mathscr{C} = \{0\}$ and we obtain LES. In our simulation experiments, the best δ we found empirically (from a few selections) never exceeded 0.4. We report and use this best value for each example.

Another heuristic to select the weights is to always include a fixed proportion $\beta \in [0,1]$ of customers from the head of the queue. We replace C by $C' = \lceil \beta C \rceil$ in (4), which gives the predictor

$$D = \frac{1}{C'+1} \sum_{c=0}^{C'} E(c). \tag{6}$$

Choosing $\beta = 0$ excludes all customers in queue and then E-LES becomes the same as LES. We shall use (5) and not (6) for our simulation experiments.

3.6 Average LES Conditional on Queue Length (AvgC-LES)

This method is inspired by the QL predictor for a single queue with exponential service times, which predicts the delay as the expected wait time conditional on the queue length when the customer arrives. Instead of using a mathematical formula based on exponential service times as in QL, this proposed predictor uses the wait times of past customers of the same type who found the same queue length when they arrived.

More precisely, for each queue j, we select a maximal queue size K_j to be considered and, for each queue size $k \in \{1,\ldots,K_j\}$, we select an integer $N_{j,k} > 0$ as in Avg-LES. We memorize the wait times of the last $N_{j,k}$ customers of class j who found a queue of size k at their arrival. For a new arrival of type j that finds a queue j of size k, the wait time is predicted by the average of those $N_{j,k}$ previous wait times. If k is unbounded or if certain values of k are rare, then we can regroup the values in a smaller number of subsets and maintain an average for each subset. If fewer than $N_{j,k}$ waits have been recorded so far, we take the average of those recorded. If none has been recorded, we take the LES.

For this predictor, contrary to Avg-LES, a larger $N_{j,k}$ usually performs significantly better than $N_{j,k} = 1$. The key difference with Avg-LES is that here the average is only over customers that see the same queue length when they arrive. We observed that for long simulations with a single queue, the accuracy of this predictor is very close to that of QL. This can be explained by the fact that AvgC-LES has collected enough data to compute good expected conditional wait times just like QL. In a many-server heavy-traffic efficiency-driven regime (Whitt 2004), AvgC-LES with $N_{j,k} = 1$ becomes the LES predictor conditional on the queue length, whose predictions are close to those of QL.

One could also consider weighted-average versions of AvgC-LES, which replace the ordinary average of the $N_{j,k}$ previous wait times for class j and queue size k by a weighted average as in WAvg-LES. In particular, one can use exponentially decreasing weights with small smoothing factors (e.g., 0.1 or less), so that each new observation makes a relatively small contribution to the average. One advantage of exponential smoothing in the long run is that there is no need to store all the individual wait time observations. In our experiments, exponential smoothing performed similarly but never better than the ordinary average in terms of prediction error, so we do not report detailed results for it.

4 SIMULATION RESULTS

In this section, we report the results of simulation experiments that compare the accuracy of new and existing predictors on three queueing models. We start with the classic M/M/s+M model, for which an analytic formula is available for the expected delay given the current state of the system. The aim is to verify that our predictions are not too far from these exact expectations in this simple case. The second example is an N-model, with two classes of customers and two groups of servers, in which the first group serves only customers from the first class and the second group serves both classes. The third one is a model of a multiskill center based on real data from the call center of a utility provider in Quebec, Canada.

The model has six classes of customers (call types), eight agent groups, and is non-stationary. Our models are simulated using the call center simulator *ContactCenters* (Buist and L'Ecuyer 2005, Buist 2009) in which we have implemented the delay predictors.

For the predictors that require parameters, we explored a few choices and selected those that gave the best results. The best N_j for Avg-LES are usually small (less than 10 and often equal to 1) and the best $N_{j,k}$ for AvgC-LES are usually large (100 or more). In agreement with this, we found in our experiments that for exponential smoothing, the best smoothing factor α_j is usually larger than 0.9 for ESAvg-LES and smaller than 0.1 for the exponentially weighted version of AvgC-LES. Since the results were also very similar to those for ordinary averaging, we will not report them in the tables.

4.1 Measuring Prediction Errors

The accuracy of a predictor can be measured by its *mean squared error* (MSE). For a given class of customers, let D be the predicted delay time of a "random" customer upon arrival, and let W be the realized waiting time. We consider only the customers who experience positive wait times ($W > 0$) and who wait until they receive service (those who abandon are not considered). The MSE is defined as

$$\mathrm{MSE} = \mathbb{E}\left[(W-D)^2\right].$$

Since we cannot compute the MSE exactly, we estimate it by its empirical (and consistent) counterpart, the *average squared error* (ASE). Let N be the number of served customers who had to wait in queue. We denote their predicted and realized delays by $D_1,\ldots,D_N$, and $W_1,\ldots,W_N$, respectively. The ASE is defined as

$$\mathrm{ASE} = \frac{1}{N}\sum_{n=1}^{N}(W_n - D_n)^2.$$

In our numerical experiments, we report a normalized version of the ASE, called the *root relative average squared error* (RRASE), which is the square root of the ASE divided by the average wait time of the N customers:

$$\mathrm{RRASE} = \frac{\sqrt{\mathrm{ASE}}}{\sum_{n=1}^{N} W_n/N} \times 100.$$

4.2 A Single Queue of Type M/M/s+M

We consider an M/M/s+M single-queue model with time varying arrival rate. The day is divided into 20 periods of one hour. The arrival process is Poisson with constant rate λ_p in period p, for $p = 1,\ldots,20$. We take $\lambda_p = 25$ for odd p and $\lambda_p = 20$ for even p. The service times are exponential with mean 1 and the patience times are exponential with mean 2. There are $s = 20$ servers for the entire day. We simulate the model for 100 independent days to estimate the accuracy of the predictors. We find that the average queue length over the day is 7.7 customers, the delay probability is 91.9%, the abandonment probability is 15.8%, and the average waiting time is 0.33 hour.

For this model, the QL predictor (2) gives the exact conditional expectation and minimizes the MSE, so it is optimal for our criterion, under the assumption of exponential service and patience times with known and constant rates (i.e., if μ, ν and $s = 20$ are known and do not vary with time). We compare the performance of other predictors with QL to see how close they are from optimal.

Table 1 reports the RRASEs for various predictors. We used $N_j = 2$ for Avg-LES, $N_{j,k} = 100$ for AvgC-LES, and $\delta = 0.1$ for E-LES. QL wins, which is no surprise, followed very closely by AvgC-LES. The other methods give significantly larger RRASE, and the best of them is our newly proposed E-LES. Avg-LES with $N_j \geq 2$, often used in practice, does worse than LES, which corresponds to $N_j = 1$. Ibrahim, Armony, and Bassamboo (2016) found similar behavior. P-LES turns out to be the worst predictor.

Table 1: RRASEs for the M/M/20+M example.

	LES	Avg-LES	P-LES	E-LES	AvgC-LES	QL
RRASE	46.9	49.4	59.2	43.6	32.9	32.1

Figure 1 displays the real delays and those predicted by QL, LES, AvgC-LES, and P-LES, as functions of the arrival times, for the 13th to 16th hour of one simulation run of a day. It gives an idea of how the prediction errors behave. Of course, this behavior differs across different days.

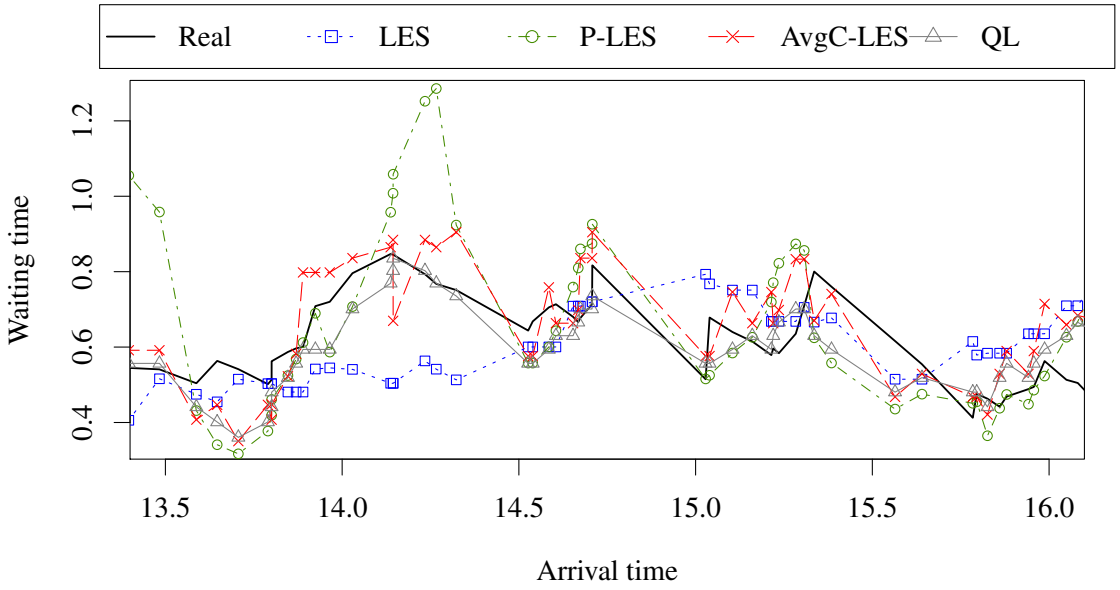

Figure 1: Comparing real delays vs. predictions by LES, P-LES, AvgC-LES, and QL.

4.3 An N-Model Call Center

We take a slightly more complex system with two call types and two agent groups, as in Thiongane, Chan, and L'Ecuyer (2015). A customer represents a phone call, and each server is an agent that can handle calls. Agents within the same group are homogeneous. Group 1 can serve only calls of type 1, and group 2 can serve all calls. This is illustrated in Figure 2, which also shows why this is called an N-model. We assume the following priority routing policy. There is an FCFS wait queue for each call type. Agents of group 2 always give priority to calls of type 2, even if a call of type 1 has waited longer. When a new call of type 1 arrives, the system will first try to assign this call to an idle agent of group 1. If none is available, it will try to match the call with an idle agent of group 2. If all agents are busy, this call joins the wait queue.

The day is divided into 10 periods of one hour. In each period, the arrival process is Poisson with a constant arrival rate. The service times and patience times are exponential with constant rates. In our numerical example, for call type 1, the vector of the arrival rates is $\lambda_1 = (25,34,43,48,51,57,42,34,22,18)$ per hour, the mean service time is $\mu_1^{-1} = 21$ minutes, and the mean patience is $\nu_1^{-1} = 46.7$ minutes. For call type 2, these parameters are respectively $\lambda_2 = (26,40,47,59,68,59,48,43,39,29)$, $\mu_2^{-1} = 11$, and $\nu_2^{-1} = 30$. The staffing vectors (number of agents of each group in each period) are $s_1 = (4,6,9,10,9,9,9,8,5,5)$ for group 1 and $s_2 = (4,7,9,10,9,8,7,8,6,5)$ for group 2. We simulate 100 independent days (replications).

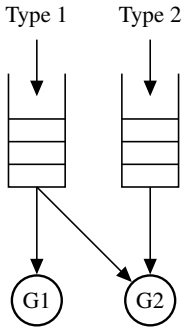

Figure 2: Diagram of an N-model system with 2 call types and 2 agent groups.

We find that 88% of served calls of type 1 were handled by group 1 (and the others by group 2). Table 2 shows some performance measures for the two call types, aggregated over the day.

Table 2: Average performance measures of the N-model example.

Performance measures	Type 1	Type 2
Delay probability (%)	94.0	97
Abandonment ratio (%)	33	23
Average queue length	9.7	5.5
Average waiting time (AWT) (Sec.)	938	426
Conditional AWT (Sec.)	1151	465

Table 3 reports the RRASEs for the two call types, for various predictors. We took $N_j = 7$ for Avg-LES, $N_{j,k} = 100$ for AvgC-LES, and $\delta = 0.2$ for E-LES. Note that QL does not apply in this case. In addition to the DH predictors, we also tried the RS and ANN predictors from Thiongane, Chan, and L'Ecuyer (2015), mentioned in the introduction. The RS and ANN predictors perform better, but they require a learning phase that is very costly and have many parameters. We use them as benchmarks for comparison. Among the DH predictors, AvgC-LES gives the best performance and comes close to RS and ANN. P-LES performs very poorly. LES, Avg-LES, and E-LES have comparable performance.

Table 3: RRASE for each type call, for the N-model example.

Type	LES	Avg-LES	P-LES	E-LES	AvgC-LES	RS	ANN
1	49.9	52.1	70.2	46.7	37.3	36.4	35.1
2	62.9	67.1	94.6	61.0	47.3	44.3	42.3

4.4 A Larger Call Center Based on Real Data

We consider a larger example inspired by the data from a subset of calls and agents in a real call center of a utility provider in Quebec, Canada. The center operates from 8 a.m. to 6 p.m. in a day. These opening hours are divided into 40 time periods of 15 minutes. The entire call center handles 96 call types with 375 agent groups, but we have selected the 6 call types having the largest volumes and 8 agent groups that can serve them, as in Chan, Koole, and L'Ecuyer (2014).

The skill sets of these 8 agent groups are $S_1 = \{1,3,4,5\}$, $S_2 = \{1,2\}$, $S_3 = \{3,5\}$, $S_4 = \{3,5,6\}$, $S_5 = \{1,3,5\}$, $S_6 = \{1,2,3,5\}$, $S_7 = \{3,5,6\}$, and $S_8 = \{1,3,5,6\}$. The arrivals are Poisson with constant rate $\lambda_{j,p}$ in period p for each type j. The average arrival rates per period, aggregated over the day, for the six call types are: $35.5, 6.0, 98, 6.5, 29$, and 3.5. Patience times are exponential with means (for the six call types): $(52, 36, 41, 51, 41, 15)$. The service times are lognormal, with different parameters for each pair of agent group g and call type j that can be served by this group. The estimated means vary from 5.14 to

11.3 minutes, and the standard deviations range from 5.88 to 22.0 minutes. The parameters were slightly altered from those of the real center for confidentiality reason. This example has the particularity that the arrival rates and the staffing change significantly during the day.

We simulated 100 independent days, as for the previous examples. Table 4 shows the aggregated performance measures for the six call types. Note that the average queue length varies significantly across the call types (from about 3 to 120). Call type 6, whose queue length and average wait time are smaller, actually has high priority for all groups that can serve it. Call type 3 has a lower priority and the highest volume.

Table 4: Average performance measures of the larger example.

Performance measures	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6
Delay probability (%)	98.1	99.5	98.6	97.9	97.8	99.2
Abandonment ratio (%)	34.2	37	43	34.4	39.2	13.6
Average queue length	41.3	6.65	120	6.95	38.2	2.96
Average waiting time (Sec.)	1037	1078	1610	1053	1125	208

Table 5 shows the RRASEs for the six call types. We used $N_j = 10$ for Avg-LES, $N_{j,k} = 200$ for AvgC-LES, and $\delta = 0.4$ for E-LES. As expected, the more expensive RS gives the best predictions for all call types, AvgC-LES is the best performer (by far) among the DH methods, and P-LES is the worst performer. The difference of accuracy between RS and AvgC-LES is larger here than for the previous example, except for call type 6. The explanation is that this call type has high priority for all groups that can serve it and its arrival rate does not vary much with time. For the other call types, we have larger variation in arrival rates and in staffing, and this affects the accuracy of DH predictors. Ibrahim and Whitt (2009a) have also observed that DH predictors lose their accuracy when the staffing and the arrival rate vary significantly. However, we find here that AvgC-LES loses its accuracy less rapidly compared to the other DH predictors.

Table 5: RRASEs for the 6 call types of the larger example.

Type	LES	Avg-LES	P-LES	E-LES	AvgC-LES	RS
1	24.6	25.4	45.1	23.2	13.0	8.9
2	35.6	34.8	95.6	34.2	22.7	12.9
3	20.3	21.6	28.4	20.4	16.9	11.4
4	41.3	55.7	67.1	39.1	22.4	15.9
5	26.9	28.7	31.0	25.2	22.9	18.9
6	94.5	96.1	130	93.2	65.8	62.7

5 CONCLUSION

We extend the family of delay-history predictors for service systems by introducing two new delay predictors, based on simple heuristics. The first idea is to exploit the more recent but incomplete delay information of customers still waiting in queue (E-LES). Their final wait times are estimated by using a simple extrapolation of their progression in the queue. The other idea proposes an empirical version of the QL formula in the context of multiskill systems, using historical data. For each queue size, a conditional expected wait time is estimated from the past delays of customers who found the same queue length in front of them when they arrived (AvgC-LES). In a single queue system, our new predictors are better than other simple predictors that we know and in addition, we observe that AvgC-LES is very close to the optimal QL predictor. For realistic multiskill systems, which typically have time-varying arrival rates and staffing, our predictors also perform better than other DH predictors. Although they do not beat methods from machine learning, their

advantages are that they are simpler to implement, have few parameters, and require no training. They represent interesting simple alternatives to more complex predictors.

ACKNOWLEDGMENTS

This work has been supported by grants from NSERC-Canada and Hydro-Québec, a Canada Research Chair, an Inria International Chair, to P. L'Ecuyer, and a "Bourse de la Francophonie" to M. Thiongane.

REFERENCES

Ang, E., S. Kwasnick, M. Bayati, E. L. Plambeck, and M. Aratow. 2016. "Accurate Emergency Department Wait Time Prediction". *Manufacturing & Service Operations Management* 18 (1): 141–156.

Armony, M., N. Shimkin, and W. Whitt. 2009. "The Impact of Delay Announcements in Many-Server Queues with Abandonments". *Operations Research* 57:66–81.

Buist, E. 2009. *Simulation des centres de contacts*. Ph. D. thesis, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, Canada.

Buist, E., and P. L'Ecuyer. 2005. "A Java Library for Simulating Contact Centers". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 556–565. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Chan, W., G. Koole, and P. L'Ecuyer. 2014. "Dynamic Call Center Routing Policies Using Call Waiting and Agent Idle Times". *Manufacturing & Service Operations Management* 16 (4): 544–560.

Dong, J., E. Yom Tov, and G. Yom Tov. 2016. "The Impact of Delay Announcements on Hospital Network Coordination and Waiting Times". Working paper.

Ibrahim, R., M. Armony, and A. Bassamboo. 2016. "Does the Past Predict the Future? The Case of Delay Announcements in Service Systems". *Management Science*. Forthcoming.

Ibrahim, R., and W. Whitt. 2009a. "Real-Time Delay Estimation Based on Delay History". *Manufacturing and Services Operations Management* 11:397–415.

Ibrahim, R., and W. Whitt. 2009b. "Real-Time Delay Estimation in Overloaded Multiserver Queues with Abandonments". *Management Science* 55 (10): 1729–1742.

Ibrahim, R., and W. Whitt. 2011. "Real-Time Delay Estimation Based on Delay History in Many-Server Service Systems with Time-Varying Arrivals". *Production and Operations Management* 20 (5): 654–667.

Jouini, O., Y. Dallery, and Z. Aksin. 2011. "Call Centers with Delay Information: Models and Insights". *Manufacturing and Service Operations Management* 13 (4): 534–548.

Senderovich, A., M. Weidlich, A. Gal, and A. Mandelbaum. 2014. "Queue Mining – Predicting Delays in Service Processes". In *Advanced Information Systems Engineering: 26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16-20, 2014. Proceedings*, 42–57. Cham: Springer International Publishing.

Senderovich, A., M. Weidlich, A. Gal, and A. Mandelbaum. 2015. "Queue Mining for Delay Prediction in Multi-Class Service Processes". *Information Systems* 53:278–295.

Thiongane, M., W. Chan, and P. L'Ecuyer. 2015. "Waiting Time Predictors for Multiskill Call Centers". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. Rosetti, 3073–3084. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tibshirani, R. 1999. "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society* 7 (0): 267–288.

Whitt, W. 1999. "Predicting Queueing Delays". *Management Science* 45 (6): 870–888.

Whitt, W. 2004. "Efficiency-Driven Heavy-Traffic Approximations for Many-Server Queues with Abandonments". *Management Science* 50 (10): 1449–1461.

AUTHOR BIOGRAPHIES

MAMADOU THIONGANE is a PhD student at the Université de Montréal, Canada. His main research interests are in estimating wait times and modeling service times in call centers. He is currently working on the development of wait time predictors in multi-skill call centers. His email address is thiongam@iro.umontreal.ca.

WYEAN CHAN is a postdoctoral fellow at the Université de Montréal, Canada. He holds a PhD degree in Computer science from the Université de Montréal. His main research projects are in stochastic optimization, simulation, and wait time estimation in call centers. He is a key contributor in the development of simulation and optimization software for call centers available at http://simul.iro.umontreal.ca. His email address is chanwyea@iro.umontreal.ca.

PIERRE L'ECUYER is Professor in the Département d'Informatique et de Recherche Opérationnelle, at the Université de Montréal, Canada. He holds the Canada Research Chair in Stochastic Simulation and Optimization and an Inria International Chair in Rennes, France. He is a member of the CIRRELT and GERAD research centers. His main research interests are random number generation, quasi-Monte Carlo methods, efficiency improvement via variance reduction, sensitivity analysis and optimization of discrete-event stochastic systems, and discrete-event simulation in general. He served as Editor-in-Chief for *ACM Transactions on Modeling and Computer Simulation* from 2010 to 2013. He is currently Associate Editor for *ACM Transactions on Mathematical Software*, *Statistics and Computing*, and *International Transactions in Operational Research*. More information can be found on his web page: http://www.iro.umontreal.ca/~lecuyer.