Evaluating the Performance of the Erlang Models for Call Centers

Thomas R. Robbins
East Carolina University
College of Business
Department of Marketing and Supply Chain
3136 Bate Building
Greenville, NC 27858-4353, USA

Abstract

In this paper we evaluate two queuing models used to analyze call centers; the Erlang C and Erlang A models. The Erlang C is a simple model that ignores caller abandonment and is the most commonly used model. The Erlang A model allows for abandonment, but performance measures are more difficult to calculate. We compare the theoretical performance predictions of these models to a steady state simulation model where many of the simplifying assumptions used in standard analytical models are relaxed. Our findings support the assertion that the Erlang A model is more accurate, but we find that in contrast to the Erlang C model, Erlang A tends to be optimistically biased. Our findings indicate that neither model clearly dominates the other in all situations and that care must be taken to select the correct model based on call center conditions and the intended purpose of the model.

Keywords: Call Centers, Queuing Models. Simulation Analysis

Introduction

Call centers are examples of queuing systems; calls arrive, wait in a queue, and are then serviced by an agent. Call centers are typically modeled using the M/M/N queuing model, or in industry standard terminology - the Erlang C model. The Erlang C model makes many assumptions which are questionable in the context of a call center environment. Specifically, the Erlang C model assumes that calls arrive at a Poisson process with a known average rate, and that they are serviced by a defined number of statistically identical agents with service times that follow an exponential distribution. Most significantly, Erlang C assumes no abandonment. The model is used widely by both practitioners and academics.

Recognizing the deficiencies of the Erlang C model, many papers have advocated using alternative queuing models and staffing heuristics which account for conditions ignored in the Erlang C model. The most popular alternative is the Erlang A model, an extension of the Erlang C model that allows for caller abandonment. For example, in a widely cited review of the call center literature (Gans et al., 2003) , the authors state "*For this reason, we recommend the use of Erlang A as the standard to replace the prevalent Erlang C model*." Another widely cited paper examines empirical data collected from a call center (Brown et al., 2005) and these authors make a similar statement; "*using Erlang-A for capacity-planning purposes could and should improve operational performance. Indeed, the model is already beyond typical current practice (which is Erlang-C dominated), and one aim of this article is to help change this state of affairs*."

The purpose of this study is first, to examine the performance of each Erlang model under real–world conditions, and second to evaluate the assertion that the Erlang A model is a superior model. We conduct this analysis by performing a detailed simulation study. We develop a simulation model to predict steady state expected system performance based on a realistic set of modeling assumptions as identified in the literature. We compare key performance metrics from our simulation study to those predicted by the Erlang C and Erlang A models and seek to characterize the error in the theoretical predictions. In this paper we restrict the analysis to cases where the call center has sufficient capacity to handle all calls without abandonment; sometimes referred to as the *quality-driven* and the *quality and efficiency-driven* (QED) regimes (Gans et al., 2003).

Our findings confirm that the Erlang A model is indeed a more accurate model in the sense that it makes predictions which over a wide range of input conditions result in a lower error. However, we also find that Erlang A does not dominate Erlang C under all conditions; in other words, there are situations in which the Erlang C model provides a better estimate, even in cases where the abandonment level is non-negligible. Furthermore, we find that while the Erlang C model tends to provide a pessimistic estimate (*i.e.*, the system performs better than predicted), the Erlang A model often provides an optimistic estimate.

While it is well established that Erlang C based work force management systems tend to overstaff the call center (Gans et al., 2003), we conclude that the use of the Erlang A model may lead to understaffing. The remainder of this paper is organized as follows. In Section 2 we review the Erlang C and Erlang A models and highlight the relevant literature. In Section 3 we present a general model of a steady state call center environment and review the simulation model we developed to evaluate it. In Section 4 we evaluate the performance of the Erlang C model, while section 5 evaluates the performance of the Erlang A model. In Section 6 we compare the two models. We conclude in Section 7 with summary observations and identify future research questions.

Queuing Models and the Associated Literature

Call centers are often modeled as queuing systems. Queuing models are used to estimate system performance so that the appropriate staffing level can be determined in order to achieve a desired performance metric such as the *Average Speed to Answer*, or the *Abandonment Percentage*. The most common queuing model used for inbound call centers is the Erlang C model (Brown et al., 2005; Gans et al., 2003). The Erlang C model (M/M/N queue) is a very simple multi-server queuing. Calls arrive according to a Poisson process at an average rate of λ. By the nature of the Poisson process interarrival times are independent and identically distributed exponential random variables with mean λ^{-1}. Calls enter an infinite length queue and are serviced on a First Come – First Served (FCFS) basis. All calls that enter the queue are serviced by a pool of n homogeneous (statistically identical) agents. Service times follow an exponential distribution with a mean service time of μ^{-1}. The steady state behavior of the Erlang C queuing model is easily characterized. Following the derivation presented in (Gans et al., 2003) we define the *offered load (R)*, a unit-less quantity often referred to as the number of Erlangs, as

$$R \equiv \lambda / \mu \tag{1}$$

The *offered utilization (*ρ*)* (aka *utilization, traffic intensity* or *occupancy*) is defined as

$$\rho \equiv \lambda / (N\mu) = R/N \tag{2}$$

The offered utilization represents the proportion of available agent time spent handling calls under the assumption that all calls are serviced. Given the assumption that all calls are serviced, the traffic intensity must be strictly less than one or the system becomes unstable, *i.e.* the queue grows without bound. This system can be analyzed by solving a set of balance equations to find the steady state probability that all *N* agents are busy, or equivalently the proportions of callers that must wait for service. The proportion of callers that must wait prior to service is

$$P\{Wait > 0\} = 1 - \left(\sum_{m=0}^{N-1} \frac{R^m}{m!} \right) \Bigg/ \left(\sum_{m=0}^{N-1} \frac{R^m}{m!} + \left(\frac{R^m}{N!} \right) \left(\frac{1}{1 - R/N} \right) \right) \tag{3}$$

The Erlang C ProbWait function is a convex function, with values near 0 for low to moderate levels of utilization but increasing rapidly as utilization rises above 75%. Under Erlang C ProbWait reaches 100% as utilization reaches 100%.

Another relevant performance measure for call centers managers is the ***Average Speed to Answer*** (ASA).

$$\begin{aligned} ASA \equiv E[Wait] &= P\{Wait > 0\} \cdot E[Wait \mid Wait > 0] \\ &= P\{Wait > 0\} \cdot \left(\frac{1}{N} \right) \cdot \left(\frac{1}{\mu} \right) \cdot \left(\frac{1}{1 - \rho} \right) \end{aligned} \tag{4}$$

Like ProbWait the ASA curve is a convex function that increases very rapidly as utilization approaches 100%, but unlike ProbWait it has no upper limit.

A third important performance metric for call center managers is the ***Telephone Service Factor*** (TSF), also called the "service level." The TSF is the fraction of calls presented which are eventually serviced and for which the delay is below a specified level. For example, a call center may report the TSF as the percent of callers on hold less than 30 seconds. The TSF metric can then be expressed as

$$\begin{aligned} TSF \equiv P\{Wait \le T\} &= 1 - P\{Wait > 0\} \cdot P\{Wait > T \mid Wait > 0\} \\ &= 1 - P\{Wait > 0\} \cdot e^{-N\mu(1-\rho)T} \end{aligned} \tag{5}$$

The Erlang C TSF function is concave, remaining near 100% for low to moderate levels of utilization, but decreasing rapidly for utilization levels above 75%. TSF approaches 0 as utilization approaches 100%.

A fourth performance metric monitored by call center managers is the ***Abandonment Rate*;** the proportion of all calls that leave the queue (hang up) prior to service. Abandonment rates cannot be estimated directly using the Erlang C model because the model assumes no abandonment occurs.

A substantial amount of research analyzes the behavior of Erlang C model; much of it seeks to establish simple staffing heuristics based on asymptotic frameworks applied to large call centers. Halfin & Whitt (1981) develop a formal version of the square root staffing principle for M/M/N queues in what has become known as the Quality and Efficiency Driven (QED) regime. Borst, Mandelbaum et al. (2004) develop a framework for asymptotic optimization of a large call center with no abandonment. Janssen, van Leeuwaarden et al. (2011) develops a refinement of the square root staffing heuristic by expanding the Erlang C model. The Erlang C model makes many assumptions, several of which are not wholly accurate. In the case of the Erlang C model several assumptions are questionable, but clearly the most problematic is the no abandonment assumption, as even low levels of abandonment can dramatically impact system performance (Gans et al., 2003). Many call center research papers however analyze call center characteristics under the assumption of no abandonment (Borst et al., 2004; Gans & Zhou, 2007; Green, Kolesar, & Soares, 2003; Green, Kolesar, & Soares, 2001; Jennings, Mandelbaum, Massey, & Whitt, 1996; Wallace & Whitt, 2005).

The Erlang C model assumes also that calls arrive according to a Poisson process. The interarrival time is a random variable drawn from an exponential distribution with a known arrival rate. Several authors assert that the assumption of a known arrival rate is problematic. Both major call center reviews have sections devoted to arrival rate uncertainty. Brown, Gans et al. (2005) perform a detailed empirical analysis of call center data. While they find that a time-inhomogeneous Poisson process fits their data, they also find that arrival rate is difficult to predict and suggest that the arrival rate should be modeled as a stochastic process. Many authors argue that call center arrivals follow a doubly stochastic process; a Poisson process where the arrival rate is itself a random variable (Aksin et al., 2007; Chen & Henderson, 2001; Whitt, 2006c). Arrival rate uncertainty may exist for multiple reasons. Call center managers attempt to account for these factors when they develop forecasts, yet forecasts may be subject to significant error. Robbins (2007) compares four months of week-day forecasts to actual call volume for 11 call center projects. He finds that the average forecast error exceeds 10% for 8 of 11 projects, and 25% for 4 of 11 projects. The standard deviation of the daily forecast to actual ratio exceeds 10% for all 11 projects. Steckley, Henderson et al. (2009) compare forecasted and actual volumes for nine weeks of data taken from four call centers. They show that the forecasting errors are large and modeling arrivals as a Poisson process with the forecasted call volume as the arrival rate can introduce significant error. Robbins, Medeiros et al. (2006) use simulation analysis to evaluate the impact of forecast error on performance measures demonstrating the significant impact forecast error can have on system performance. Sivan, Paul et al. (2009) develop a model for load forecasting and evaluate it with test data from an Israeli bank.

Several papers address staffing requirements when arrival rates are uncertain or time-varying. Bassamboo, Harrison et al. (2005) develop a model that attempts to minimize the cost of staffing plus an imputed cost for customer abandonment for a call center with multiple customer and server types when arrival rates are variable and uncertain. Harrison and Zeevi (2005) use a fluid approximation to solve the sizing problem for call centers with multiple call types, multiple agent types, and uncertain arrivals. Whitt (2006c) allows for arrival rate uncertainty as well as uncertain staffing, *i.e.* absenteeism, when calculating staffing requirements. Steckley, Henderson et al. (2004) examine the type of performance measures to use when staffing under arrival rate uncertainty. Robbins and Harrison (2010) develop a scheduling algorithm using a stochastic programming model that is based on uncertain arrival rate forecasts. Aktekin and Ekin (2016) develop staffing strategies for a call center model where arrival rate, service time, and abandonment distributions are all uncertain. Heching and Squillante (2014) apply a two-stage stochastic optimization approach to optimize service delivery centers. Roubos and Bhulai(2010) use an approximate dynamic programing technique to control queues with time-varying parameters. Feldman, Mandelbaum et al. (2008) develop a staffing model to maintain time-stable performance when arrival rates are time-varying. Lima, Maciel et al. (2014) develop a general call center evaluation approach and examine the issue of system downtime.

The Erlang C model also assumes that the service time follows an exponential distribution. The memoryless property of the exponential distribution greatly simplifies the calculations required to characterize the system's performance, and makes possible the relatively simple equations (3)-(5). If the assumption of exponentially distributed talk time is relaxed, the resulting queuing model is the $M/G/N$ queue, which is analytically intractable (Gans et al., 2003) and approximations are required. However, empirical analysis suggests that the exponential distribution is a relatively poor fit for service times. Most detailed analysis of service time distributions find that the lognormal distribution is a better fit (Brown et al., 2005; Gans et al., 2003; Mandelbaum A., Sakov A. , & S., 2001). Aktekin (2014) uses a Bayesian mixture model to analyze service time for a call center when multiple caller types are present. The fit of the Erlang C model in a call center environment is analyzed in Robbins, Medeiros et al. (2010).

Finally, the Erlang C model assumes that agents are *homogeneous*. More precisely, it is assumed that the service times follow the same statistical distribution independent of the specific agent handling the call. Empirical evidence supports the notion that some agents are more efficient than others and the distribution of call time is dependent on the agent to whom the call is routed. In particular more experienced agents typically handle calls faster than newly trained agents (Armony & Ward, 2008). Robbins (2007) demonstrated a statistically significant learning curve effect in an IT help desk environment.

The Erlang A Model

Given the prevalence of caller abandonment in modern call centers, the *no abandonment* assumption of the Erlang C model may be problematic. Unfortunately, models that allow for abandonment are significantly more complex and difficult to characterize. The simplest abandonment model is the $M/M/N+M$, or Erlang A model. The model was originally presented by Palm in a 1946 paper written in Swedish. It was presented in English in (Palm, 1957). The Erlang A model is presented in detail in Gans, Koole et al. (2003) and Mandelbaum and Zeltyn (2004).

Erlang A extends the Erlang C model by allowing abandonment. In the Erlang A model each caller possesses an exponentially distributed *patience time* with mean θ^{-1}. If the offered waiting time, the time a caller with infinite patience would be required to wait, exceeds the customer's patience time, the caller will abandon the queue and hang up (A Mandelbaum & Zeltyn, 2004). While the exponentially distributed patience time makes the calculations tractable, they are by no means straightforward. In particular, calculation of the performance metrics requires an evaluation of the incomplete Gamma function

$$\gamma(x,y) \equiv \int_0^y t^{x-1}e^{-t}dt, \quad x>0, \quad y\geq 0 \tag{6}$$

Details on how to calculate performance metrics for the Erlang A model are provided in Mandelbaum and Zeltyn (2009). Following their notation, we define the basic building blocks J as

$$J = \frac{e^{\lambda/\theta}}{\theta}\cdot\left(\frac{\theta}{\lambda}\right)^{\frac{n\mu}{\theta}}\cdot\gamma\left(\frac{n\mu}{\theta},\frac{\lambda}{\theta}\right) \tag{7}$$

and ε as

$$\varepsilon = \frac{\sum_{j=0}^{n-1}\frac{1}{j!}\left(\frac{\lambda}{\mu}\right)^j}{\frac{1}{(n-1)!}\left(\frac{\lambda}{\mu}\right)^{n-1}} \tag{8}$$

We can then calculate the probability of waiting as

$$P\{Wait>0\} = \frac{\lambda J}{\varepsilon+\lambda J}\cdot(1-\theta) \tag{9}$$

Garnett, Mandelbaum et al. (2002) outline a method for an exact calculation of the Erlang A performance metrics, and provide approximations based on an asymptotic analysis of the queue. Whitt (2006a) develops deterministic fluid models to provide simple first-order performance descriptions for multi-server queues with abandonment under heavy loads. Whitt (2016) develop diffusion approximations for the Erlang C and Erlang A models. Knessl and van Leeuwaarden (2015) develop a transient analysis of the Erlang A model.

The inclusion of abandonment has a profound effect on the performance of the queuing system. The impact of abandonment is discussed in detail in Garnett, Mandelbaum et al.(2002). First, the issue of system stability is no longer a concern. In an Erlang C system the traffic intensity defined in equation (2) must be strictly less than one for a steady state to exist; an intensity of one or more leads to an infinite queue size. Furthermore, even very low levels of caller abandonment can dramatically alter system performance.

Abandonment has an important impact on the shape of the performance metric curves. Due to the moderating effects of caller abandonment the curves are no longer strictly concave or convex; the ProbWait curve for example takes on an s-shape. The probability of waiting increases at an increasing level as congestion develops in the queue, but as congestion continues to grow abandonment levels begin to increase. As callers exit the queue in increasing numbers the probability of waiting continues to increase, but at a decreasing rate. The same effect applies to the ASA and TSF curves, both of which take on an s-shape under the Erlang A model.

Comparisons of Erlang C and Erlang A models are developed in Mandelbaum and Zeltyn (2004) and Garnett, Mandelbaum et al.(2002). Whitt (2005) examines the fit of the Erlang A model. Whitt (2006b) examines the sensitivity of the Erlang A model to changes in the model parameters. Several papers examine staffing and scheduling issues in call centers where abandonment is allowed (Avramidis, Gendreau, L'Ecuyer, & Pisacane, 2007; Bassamboo et al., 2005; Robbins & Harrison, 2010). Phung-Duc and Kawanishi(2014) evaluate call center performance when callers abandon and later retry their call.

In order to develop a tractable model, the Erlang A model assumes an exponentially distributed patience. Brown, Gans et al.(2005) examine abandonment and a customer's willingness to wait in detail. A customer's patience is in general an unobservable metric; since only customers whose patience expires abandon, the data is right censored. The exponential distribution of patience implies that the hazard rate for abandonment is constant over time. Brown, Gans et al. (2005) show hazard rate graphs estimated from empirical data for two different call types. Both graphs reveal hazard functions that are not constant; in contrast they show a sharp peak near the origin indicating a substantial portion of customers are unwilling to wait at all. Callers who abandon immediately are said to balk. The graphs also show another peak at 60 seconds after an announcement indicating a customer's position in the queue. The issue of delay estimating and announcing delay times, and the effect it can have on queues is evaluated in Ibrahim & Whitt (2008) and (Huang, Mandelbaum, et al (2017). The hazard function shows a general decline over the range of values plotted (0 to 400 secs.) Several other studies of patience curves have concluded that patience can be best modeled as a Weibull distribution (Gans et al., 2003). The Weibull distribution supports a constant, increasing, or decreasing hazard rate.

While many papers have noted the deficiencies of the Erlang C model, and advocated the use of the Erlang A model, a systematic analysis of the error associated with each model is lacking. Our paper seeks to close this gap in the literature.

Call Center Simulation

The Modified Model

In this section we present a revised model of a call center, relaxing several key assumptions discussed previously. In our model calls arrive at a call center according to a Poisson process. Calls are forecasted to arrive at an average rate of $\hat{\lambda}$. The realized arrival rate is λ, where λ is a normally distributed random variable with mean $\hat{\lambda}$ and standard deviation σ_λ. The time required to process a call by an average agent is a lognormally distributed random variable with mean μ^{-1} and standard deviation σ_μ. We define the offered utilization ($\hat{\rho}$) as the system utilization that would occur if all calls were serviced by agents of average productivity.

Arriving calls are routed to the agent who has been idle for the longest time if one is available. If all agents are busy the call is placed in a FCFS queue. When placed in queue a proportion of callers will balk; *i.e.* immediately hang up. Callers who join the queue have a patience time that follows a Weibull distribution. The Weibull distribution has two parameters, a shape parameter (α) and a scale parameter (β). When the shape parameter equals one the Weibull distribution is identical to the exponential distribution. (Law, 2007). If wait time exceeds their patience time the caller will abandon. Calls are serviced by agents who have variable relative productivity r_i. An agent with a relative productivity level of 1 serves calls in the average service time. An agent with a relative productivity level of 1.5 serves calls in 1.5 times the average service time. Agent productivity is assumed to be a normally distributed random variable with a mean of 1 and a standard deviation of σ_r.

Experimental Design

In order to evaluate the performance of the Erlang C and Erlang A models against the simulation model, we conduct a series of designed experiments. Based on the assumptions for our call center discussed previously, we define the following set of nine experimental factors. High and low values are defined to cover a broad range of call center scenarios. The forecasted arrival rate in the simulation is a quantity derived from other experimental factors by

$$\hat{\lambda} = \hat{\rho} N \mu \qquad (10)$$

Given the relatively large number of experimental factors, a well-designed experimental approach is required to efficiently evaluate the experimental region. A standard approach to designing computer simulation experiments is to employ either a full or fractional factorial design (Law, 2007). However, the factorial model only evaluates corner points of the experimental region and implicitly assumes that responses are linear in the design space. Given the anticipated non-linear relationship of errors, we chose instead to implement a Space Filling Design based on Latin Hypercube Sampling as discussed in (Santner, Williams, & Notz, 2003). Given a set of *d* experimental factors and a desired sample of *n* points, the experimental region is divided into n^d cells.

A sample of *n* cells is selected in such a way that the centers of these cells are uniformly spread when projected onto each of the *d* axes of the design space. We chose our design point as the center of each selected cell. This experimental design allows us to select an arbitrary number of points for any experiment.

	Factor	**Low**	**High**
1	Number of Agents	10	100
2	Offered Utilization ($\hat{\rho}$)	65%	95%
3	Talk Time (mins)	2	20
4	Patience Shape Factor β	60	600
5	Forecast Error CV	0	.2
6	Patience Scale Factor α	.75	1.25
7	Talk time CV	.75	1.25
8	Probability of Balking	0	.25
9	Agent Productivity Standard Deviation	0	.15

Table 1-Experimental Factors

Simulation Model

Our call center model is evaluated using a straightforward discrete event simulation model. The purpose of the model is to predict the long term, steady state behavior of the queuing system. The model generates random numbers using a combined multiple recursive generator (CMRG) based on the Mrg32k3a generator described in (L'Ecuyer, 1999). Common random numbers are used across design points to reduce output variance. To reduce any start up bias we use a warm up period of 5,000 calls, after which all statistics are reset. The model is then run until 25,000 calls have been serviced and summary statistics are collected. For each design point we repeat this process for 500 replications and report the average value across replications. Our primary analysis is based on an experiment with 1,000 design points.

The specific process for each replication is as follows. The input factors are chosen based on the experimental design. The average arrival rate is calculated based on the specified talk time, number of agents, and offered utilization rate according to equation(10). A random number is drawn and the realized arrival rate is set based on the probability distribution of the forecast error. That arrival rate is then used to generate Poisson arrivals for the replication. Agent productivities are generated using a normal distribution with mean one and standard deviation σ_p. Each new call generated includes an exponentially distributed interarrival time, a lognormally distributed nominal talk time, a Weibull distributed time before abandonment, and a Bernoulli distributed balking indicator. When the call arrives it is assigned to the longest idle agent, or placed in the queue if all agents are busy. If sent to the queue the simulation model checks the balking indicator. If the call has been identified as a balker it is immediately abandoned, if not an abandonment event is scheduled based on the realized time to abandon. Once the call has been assigned to an agent, the realized talk time is calculated as the product of the nominal talk time and the agent's productivity. The agent is committed for the realized talk time. When the call completes the agent processes the next call from the queue, or if no calls are queued becomes idle. If a call is processed prior to its time to abandon, the abandonment event is cancelled. If not, the call is abandoned and removed from the queue. Over the course of the simulation we collect statistics on the proportion of customers forced to wait, the average speed to answer, the abandonment rate, and the TSF defined as the proportion of total callers serviced with a wait time less than 30 seconds. Extensive testing of our simulation model verifies that all metrics are calculated consistently with the Erlang C and Erlang A predictions when the simulation is configured to support those model's assumptions.

After all replications of the design point have been executed the results are compared to the theoretical predictions of the Erlang C and Erlang A models. In each case we calculate the error as the difference between the theoretical value and the simulated value. For the Erlang C model, we use the standard analytical calculations using the same values of arrival rate, talk time, and the number of agents used in the simulation. When testing against the Erlang A model the comparison is a bit more complicated. The first challenge is we wish to eliminate any approximation errors in our comparison, so rather than use an approximate calculation for the Erlang A model we rerun the simulation configured to be consistent with the Erlang A model assumptions, *i.e.* no balking, homogeneous agents, exponential talk time and exponential patience. The simulation is run using common random numbers from the original simulation. We feel that this approach allows us to focus on the error associated with the Erlang A assumptions, rather than the numerical issues associated with estimating Erlang A performance measures. The second challenge is how to set the patience parameter for the Erlang A calculation. Recall that this parameter is not directly observable since data is heavily censored.

Since we are attempting to fit the Erlang A model to observed data, we approximate the Erlang A parameter θ as in (Gans et al., 2003) and (Brown et al., 2005) by $\theta = P\{Abandon\}/E[Wait]$.

Erlang C Experimental Analysis

Summary Observations

We conducted an experiment with 1,000 design points. Based on our analysis we can make the following summary observations:

- The Erlang C model is, on average, subject to a reasonably large error over this range of parameter values.
- Measurement errors are strongly correlated across performance measures.
- The Erlang C model is on average pessimistically biased (the real system performs better than predicted) but may become optimistically biased when utilization is high and arrival rates are uncertain.
- Measurement error is high when the real system exhibits high levels of abandonment. The error is strongly positively correlated with realized abandonment rate and predicted ASA.
- The Erlang C model is most accurate when the number of agents is large and utilization is low.
- Errors decrease as caller patience increases.

We will now review our experimental results in more detail.

Correlation and Magnitude of Errors

The magnitude of errors generated by using the Erlang C model across our test space is high on average, and very high in some cases. The errors across the key metrics are correlated with each other, and highly correlated with the realized abandonment rate. Table 2 shows a correlation matrix of the errors generated from the Erlang C model and the abandonment rate calculated from the simulation.

One of the key challenges we face when evaluating this model's performance is choosing what metric to evaluate. We have identified five key performance metrics for the model. The correlation analysis shows that while there is a statistically significant correlation between the metrics, the correlation is far from complete. Table 3 shows the summary statistics for each of the performance metrics across the design space.

	Simulated Abandonment Rate	*Prob Wait Error*	*ASA Error*	*TSF Error*	*Utilization Error*
Simulated Abandonment Rate	1.000				
Prob Wait Error	.867	1.000			
ASA Error	.766	.722	1.000		
TSF Error	-.880	-.987	-.759	1.000	
Utilization Error	.970	.861	.745	-.873	1.000

Table 2 – Correlation Matrix for Erlang C Model

	PW ECE	ASA ECE	TSF ECE	AB ECE	UT ECE
Min	-8.0%	-2.80	-51.6%	-14.3%	0.0%
Avg	8.0%	33.40	-7.8%	-2.4%	2.9%
Max	49.4%	1,098.96	3.6%	0.0%	14.0%
Skew	1.30	5.79	-1.56	-1.46	1.33

Table 3- Erlang C Error Statistics

Each statistic has a wide range of errors and those errors are imperfectly correlated. Our goal is to evaluate the fit of the model overall, across all five metrics. To facilitate an overall assessment of model fit we perform a cluster analysis. Specifically, we cluster all one thousand design points into 3 clusters, using a k-means clustering algorithm. The inputs are the normalized errors in each of the five metrics. The clusters are identified green for best, yellow for middle, and red for worst in terms of performance. Summary metrics for the clusters are provided in

Table 4 4. The cluster analysis shows that most of the design points, 665, are included in the low error cluster. The Erlang C model provides a good estimate for points in this cluster, with errors generally less than 2%. The second largest group is the moderate error/yellow cluster, with 27.8% of the observations. Here errors are becoming reasonably high, with the average error in ProbWait forecast at nearly 18%, and an average ASA error of more than 48 seconds. The smallest group is the high error/red which contains 6.1% of the data points. Here the Erlang C is a very poor predictor. The average error in the ProbWait measure is nearly 35%, and the average ASA error is nearly 5 minutes. Abandonment averages more than 8.5%, which of course is predicted to be 0 by the Erlang C model.

Cluster	Obs	PW ECE	ASA ECE	TSF ECE	AB ECE	UT ECE
Green	661	1.25%	3.30	-1.19%	-1.04%	1.54%
Yellow	278	17.99%	48.28	-17.20%	-4.27%	4.89%
Red	61	34.90%	291.73	-36.67%	-8.59%	8.99%

Table 4- Average Erlang C Errors by Cluster

Overall, the errors are generally pessimistically skewed, the system behaves in general better than predicted. Recall that the error is calculated as the predicted value minus the observed value, so a positive value for ProbWait means less calls waited than were predicted to wait. Similarly calls were answered faster than predicted, the service level was higher than predicted, and agent utilization is lower than predicted. Since the Erlang C model assumes no abandonment, actual abandonment is always higher than predicated. The pessimistic nature of the Erlang C model is most clearly demonstrated by the ASA prediction. Figure 1 shows the predicted and realized ASA values across the design space with points coded based on the cluster assignment. While the model regularly predicts calls waiting more than 30 seconds to be answered, (26.3% of design points) a wait time in excess of 30 seconds occurs only 1 time out of 1,000.

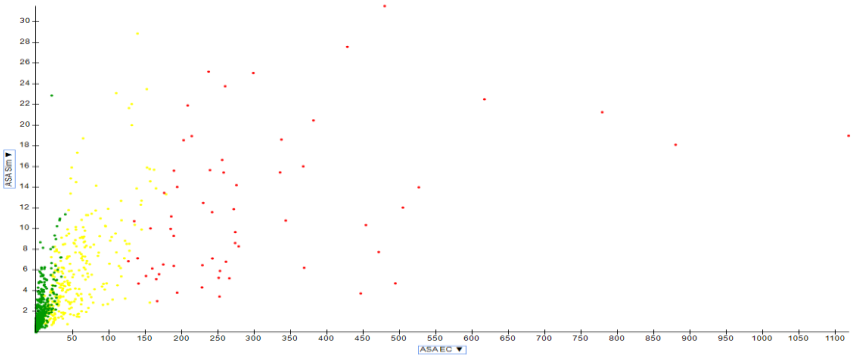

Figure 1-Erlang C Predicted vs Actual ASA

To further illustrate the predictive performance of the Erlang C model we view the scatter plot matrix of the five performance metric errors shown in Figure 2. From the matrix we can see that low error (green) cluster is tightly packed, the medium error cluster is less tightly packed, and the bad (red) cluster is spread out. This shows that when predictions are bad, they can be very bad. The 4th column of the matrix is also illustrative. This column shows that the abandonment error, which is equivalent to the realized abandonment rate. These plots clearly show that the error in each metric is strongly related to the abandonment rate.

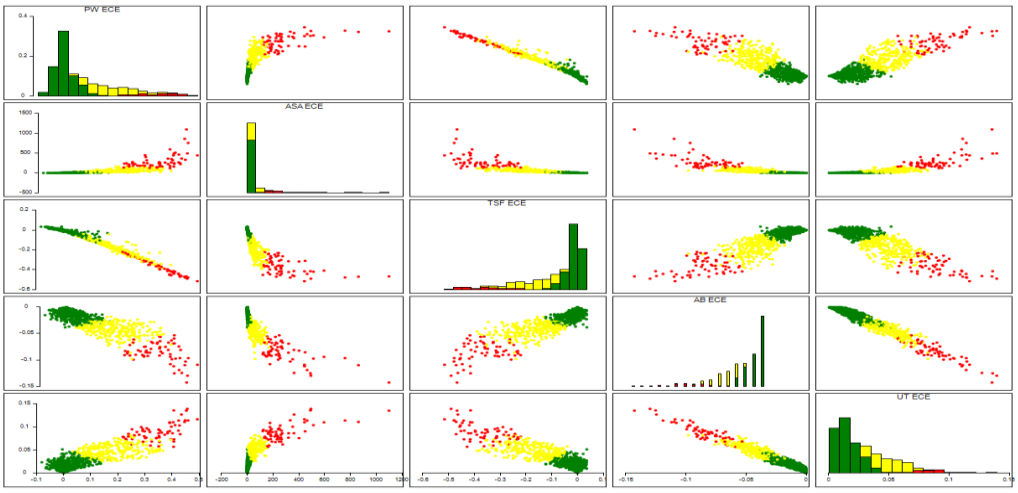

Figure 2- Scatter Plot Matrix of Erlang C Errors

Drivers of Erlang C Error

Having established that error rates are high under the Erlang C model, we now turn our attention to characterizing the drivers of that error. As discussed in the previous section, Erlang C errors are highly correlated with the realized abandonment rate. The notion that abandonment is a major driver of errors in the Erlang C model is further illustrated in Figure 3. This graph shows the error in the ProbWait measure on the vertical axis and the abandonment rate from the simulation analysis on the horizontal axis.

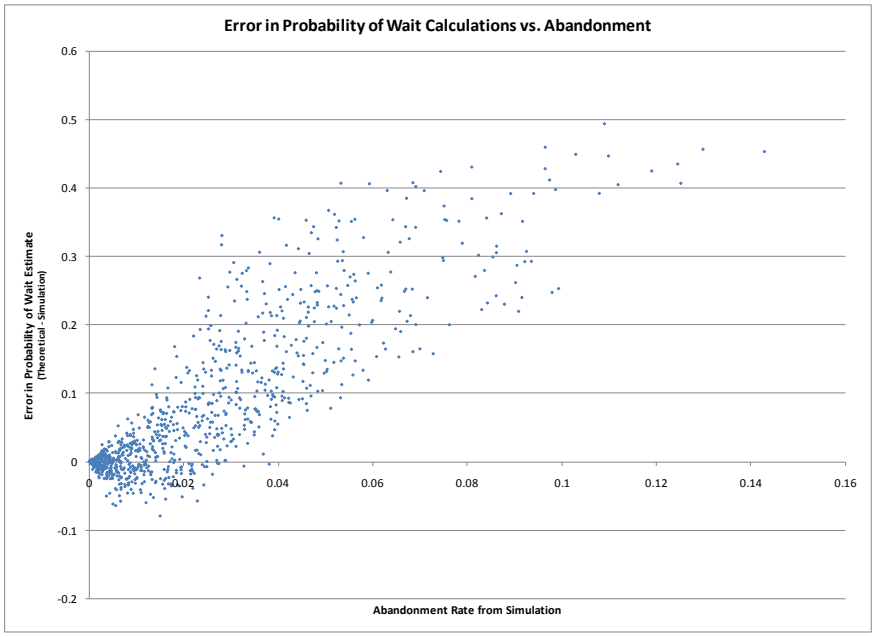

Figure 3-Scatter Plot of Erlang C Errors and Abandonment Rate

The graph clearly shows that as abandonment increases, the error in the ProbWait measure increases as well. The graph also reveals the optimistic errors, *i.e.* errors in which the system performed worse than predicted, only occur with relatively low abandonment rates. The average abandonment rate for optimistic predictions was .74%. The graph also reveals that significant error can be associated with even low to moderate abandonment rates. For example, for all test points with abandonment rates of less than 5%, the average error for ProbWait is 4.8%. For test points in which abandonment ranged between 2% and 5% the average ProbWait error is 12.2%.

Abandonment however, is an output, not an input of our model. We now seek to determine the relative impact of the 9 input parameters. We ran an Importance Test based on an F Test Statistic and developed the following graph. This test examines the effectiveness for which each input can predict the cluster for the design point.

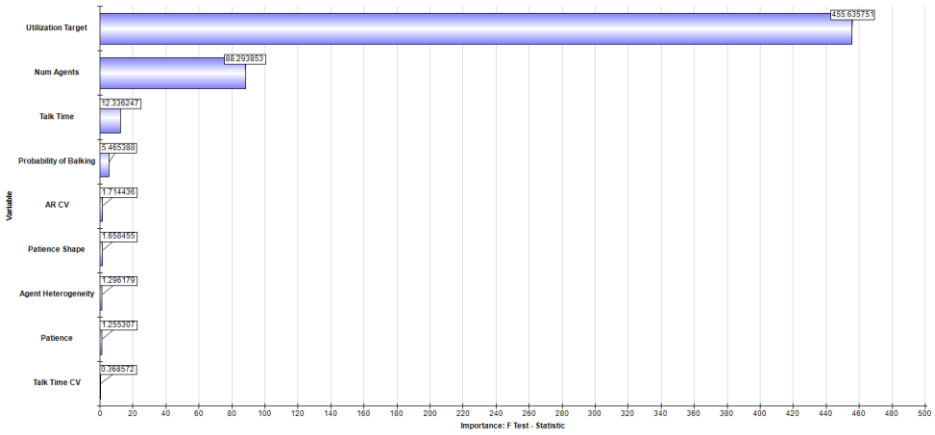

Figure 4-Erlang C Importance Test

The utilization target is clearly the dominant factor, with the number of agents a distant second. This makes sense given the distorting effects of abandonment. Environments with high offered utilization and a small pool of agents are susceptible to high abandonment rates. To further analyze this phenomenon, we ran a separate experiment, varying the number of agents from 10 to 100 for different offered utilization rates, holding other parameters constant. The results are illustrated in Figure 5.

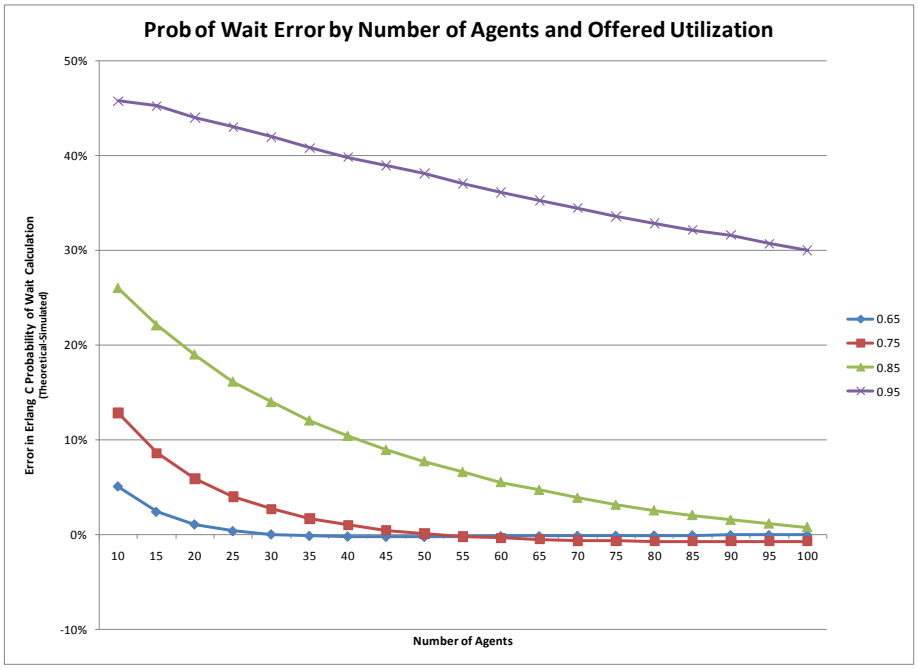

Figure 5 - Erlang C ProbWait Errors by Call Center Size and Utilization

This graph demonstrates that the Erlang C model tends to provide relatively poor predictions for small call centers. This error tends to decrease as the size of the call center increases. However, the graph also illustrates that for busy centers the error remains high. For a very busy call center, running at 95% offered utilization, the error rate remains at 30%, even with a pool of 100 agents. The errors tend to track with abandonment; abandonment rates increase with utilization and decrease with the agent pool.

Conclusion

The Erlang C model is commonly applied to predict queuing system behavior in call center applications. Our analysis shows that when we test the Erlang C model over a range of reasonable conditions, predicted performance measures are subject to large errors.

The Erlang C model works reasonably well for large call centers with low to moderate utilization rates, but factors that tend to generate caller abandonment; *i.e.* high utilization, small agent pools, and impatient callers, cause the model error to become quite large. While the model tends to provide a pessimistic estimate, arrival rate uncertainty will either reduce that bias or lead to an optimistic bias. It is clear that great care must be taken before using the Erlang C model to make any calculations that require a high level of precision in a real call center environment.

Erlang A Experimental Analysis

Summary Observations

We utilize the same 1,000 design points for the analysis of the Erlang A model. In summary we find the following

- Erlang A errors are, on average relatively small.
- Errors across measures are correlated at a statistically significant level.
- Errors tend to be optimistic, *i.e.* the system performs worse than predicted.
- Arrival Rate uncertainty has the largest impact on Erlang A prediction error.

Correlation and Magnitude of Errors

The magnitude of errors generated by using the Erlang A model across our sample is on average relatively low. Errors exhibit a moderately strong correlation as illustrated in Table 5. The level of correlation is statistically significant, but less so than for the Erlang C model. ASA in particular has a much lower correlation with ProbWait and TSF than in the Erlang C model.

	Simulated Abandonment Rate	*Prob Wait-Error*	*ASA-Error*	*TSF-Error*	*Abandonment Rate-Error*
Simulated Abandonment Rate	1.000				
Prob Wait-Error	-.005	1.000			
ASA-Error	-.622	.468	1.000		
TSF-Error	.360	-.790	-.776	1.000	
Abandonment Rate-Error	-.033	.783	.432	-.823	1.000

Table 5 – Correlation matrix for the Erlang A Model

Correlations between measurement errors are statistically significant at the .01 level, with the magnitudes of the correlations moderate to high. The errors in the ASA and TSF measures correlate strongly with the realized abandonment rate, though the ProbWait error does not. Errors are correlated with the realized abandonment, but the correlation is fairly weak. Statistics for the error in each measure are summarized in Table 6. While the error rate can be non-trivial, they are much less than the error rates seen with the Erlang C model. The magnitude of the error when using the Erlang A model is relatively low, but negatively (optimistically) biased, the system behaves in general worse than predicted. Again, recall that the error is calculated as the predicted value minus the observed value, so a negative value for ProbWait means more calls waited than were predicted to wait. Similarly, calls were answered slightly slower than predicted, the service level was lower than predicted, and as was the case with the Erlang C model agent utilization is lower than predicted. The errors for ProbWait, ASA, and TSF have opposite signs as compared to the Erlang C model and are skewed in the opposite direction.

	PW EAE	ASA EAE	TSF EAE	AB EAE	UT EAE
Min	-9.1%	-15.05	-1.8%	-2.6%	-0.3%
Avg	-1.3%	-1.02	1.2%	-0.3%	1.2%
Max	3.3%	0.83	13.2%	1.3%	4.8%
Skew	-1.14	-2.87	2.05	-1.03	1.07

Table 6 - Erlang A Error Statistics

We once again perform a cluster analysis to group the data points based on a total assessment across all errors. The clusters are once again ranked green for best, yellow for middle, and red for worst in terms of overall performance.

The results are shown in Table 7. Just under 60% of the design points are in the low error cluster. Points in this cluster are very accurate, with the average for all percentage-based metrics within 1% and the ASA average at less than a second. The second largest group is the medium error/yellow cluster. The errors here are much larger than the green cluster, but still relatively small, especially as compared to the Erlang C model. The smallest group is the red or high error group. At 11.5% there are a non-trivial number of points in this space, yet the errors remain relatively small, with ProbWait and TSF errors just under 5%. Figure 6 shows a scatter plot matrix of the 5 error measures.

Cluster	Obs	PW ECE	ASA ECE	TSF ECE	AB ECE	UT ECE
Green	599	-0.21%	-0.45	0.40%	0.01%	0.64%
Yellow	286	-2.22%	-1.40	1.76%	-0.59%	1.59%
Red	115	-4.53%	-3.03	4.20%	-1.42%	2.71%

Table 7 - Average Erlang A Errors by Cluster

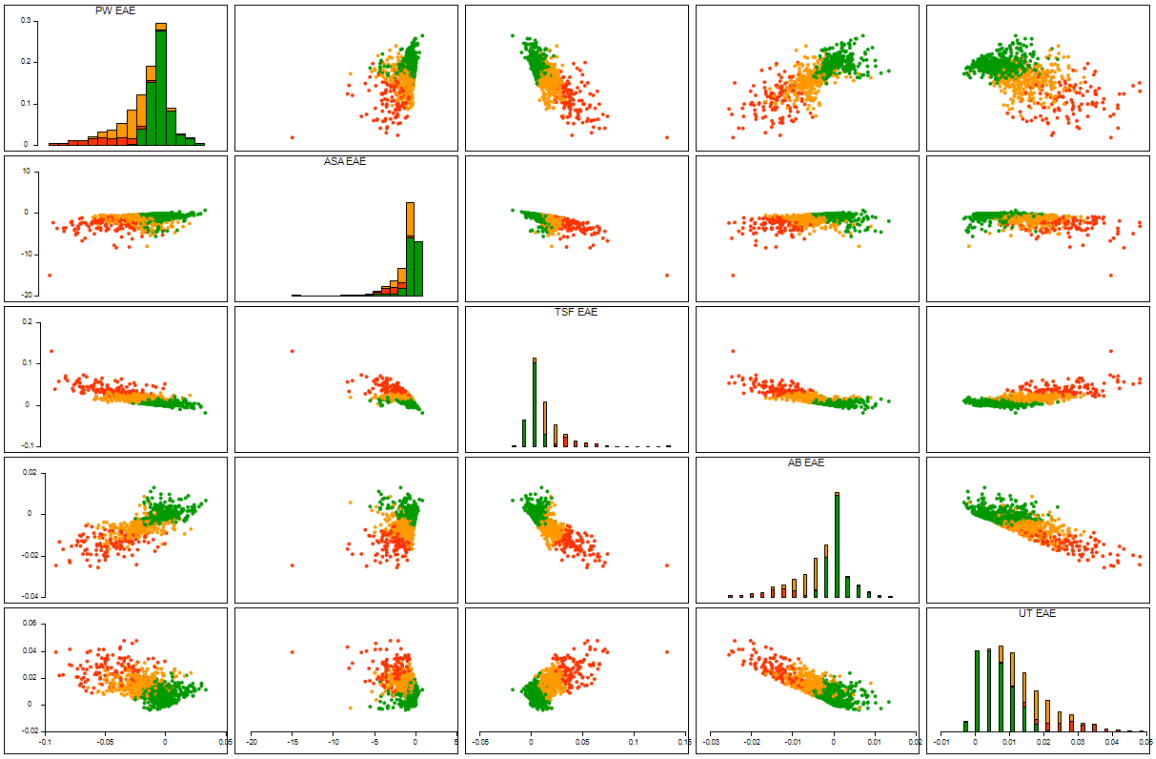

Figure 6-Erlang A Error Scatter Plot Matrix

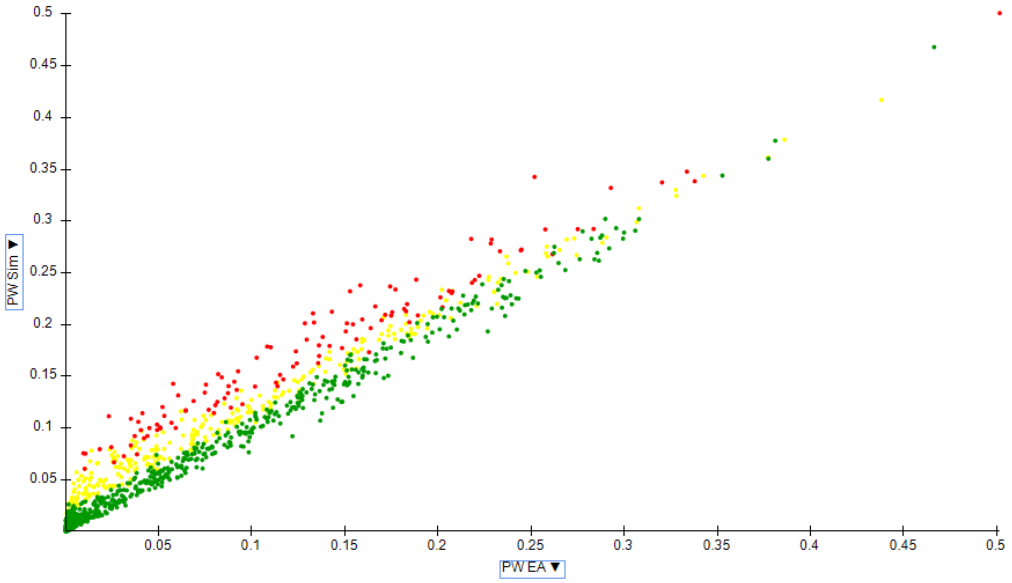

Figure 7-Erlang A ProbWait Predicted vs Actual

We examine the performance of the Erlang A prediction compared to the simulated value in Figure 7. This scatter plot show the ProbWait value predicted by the model and that obtained from the simulation. The scatter graph show that the error in ProbWait is generally independent of the expected wait. High and low errors occur for low wait probabilities and high wait probabilities. The higher error conditions tend to be optimistic errors, scenarios where the system performs worse than expected.

Drivers of Erlang A Errors

As we did with the Erlang C model, we now turn our attention to determining the drivers of the relative error. We again run an importance test to determine which inputs are most influential in determining a points error cluster.
The results are shown in Figure 8. By far the most influential factor is Arrival Rate Uncertainty. This issue is further illustrated in Figure 9, which shows the error in the ProbWait metric vs. the ARCV parameter. For low values of ARCV, precise arrival rate prediction, the error is low and often positive. As ARCV increases the error in ProbWait increases dramatically and is mostly negative. Recall that the ARCV parameter defines the uncertainty in the arrival rate, not the overall error. Scenarios with high ARCV correspond to conditions where the arrival rate is accurate on average, but varies from period to period more than predicted by the Poisson distribution.

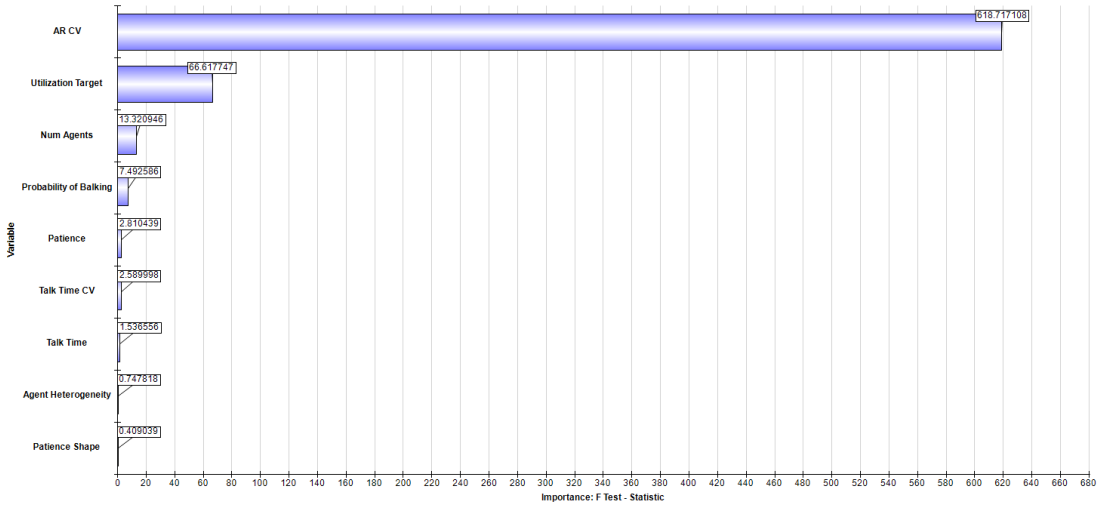

Figure 8-Erlang A Importance Test

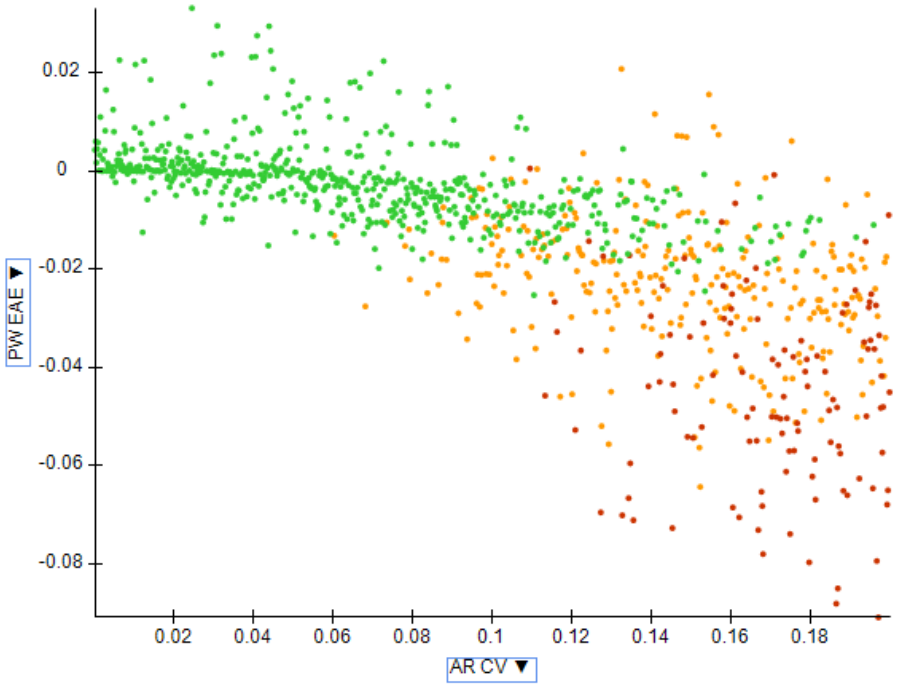

Figure 9- Erlang A ProbWait error vs Arrival Rate Uncertainty

Conclusion

Overall the Erlang A model is reasonably accurate model of a call center's behavior. The average error rates across all design points are relatively low. The model can however, under certain conditions, exhibit significant error rates. The model is most susceptible to variability/uncertainty in the arrival rate. Unlike the Erlang C model, the highest errors tend to be optimistic, the call center performs worse than predicted.

Comparing the Erlang C and A Models

Overview

In this section we compare the relative performance of the Erlang C and Erlang A models. We compare prediction errors between the two models for each of the 1,000 points in our experimental design. Figure 10 shows a scatter plot of error in the ProbWait calculation for each observed point.

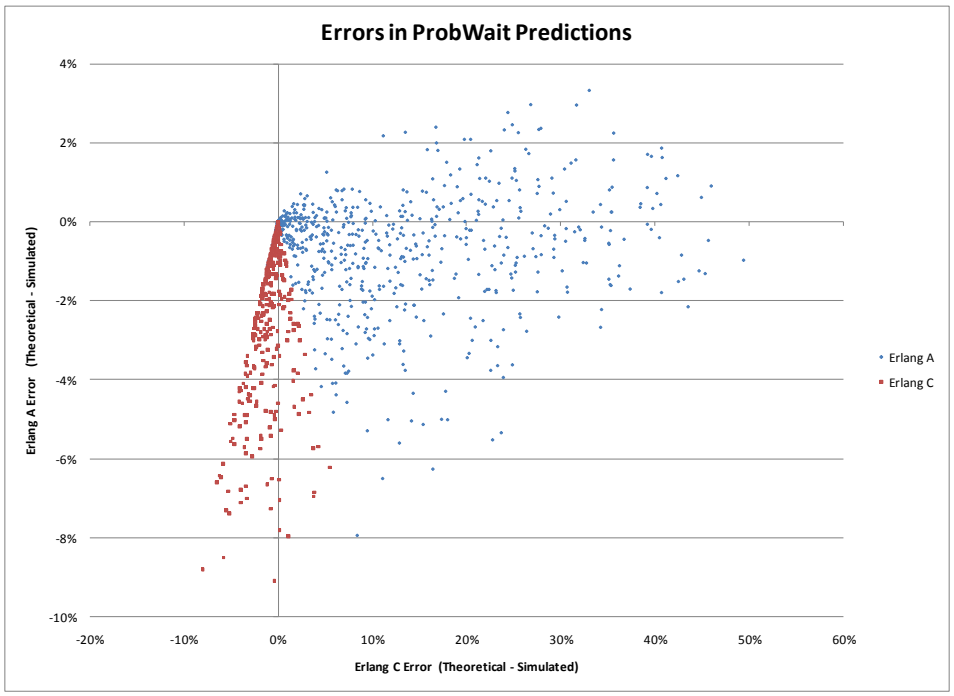

Figure 10 – Comparing ProbWait Errors for Erlang C and A

Note the different scales; Erlang C error occurs over a range of -8% to 50%, while Erlang A error occurs over a range of -9.1% to 3.3%. The average error of the Erlang C model is 7.96%, while the average error from the Erlang A model is -1.28%. The overall assertion that the Erlang A model is more accurate is in general supported by the data; the average error is smaller, and the range of errors is much smaller. However, as the figure shows the model is not universally more accurate, the graph is configured to indicate which model has the lower absolute error at each point. Of the 1,000 points tested, the absolute value of the error from the Erlang A model is less than the absolute value of the Erlang C model error 63.5% of the time, while the Erlang C model was more accurate 36.5% of the time.

One of the key observations of our analysis has been the pessimistic nature of the Erlang C estimate, and the somewhat optimistic nature of the Erlang A estimate. To further investigate this issue, we calculate the prediction percentile for each design point, *i.e.* the proportion of observations where the realized proportion of callers waiting was less than then that predicted by each model.

Erlang C has a percentile score higher than or equal to the Erlang A score 100% of the time. The Erlang C model is quite conservative, with a percentile value of 100 12.9% of the time. The percentile score exceeds 95 25.9% of the time. In some cases, the Erlang A has relatively low percentile scores, as low as 32.6, but overall the Erlang A has a somewhat surprisingly high percentile value, greater than 50% in 93.5% of the test points. This implies that even for points with an optimistic bias, performance will be better than predicted in many cases but far worse in some cases. Due to arrival rate uncertainty, performance measures such as ProbWait are more variable than predicted by a model which assumes known arrival rates. Furthermore, the distribution of ProbWait tends to have a relatively high positive skew.

Drivers of Relative Performance

We have seen that the Erlang A model is not universally more accurate than the Erlang C model. An interesting question is under what conditions is the Erlang C model more accurate. To better understand this we segregated the design points into two groups, as illustrated in Figure 10; those where the absolute error of the Prob Wait metric in the Erlang C model was smaller, and those where the absolute error of the Erlang A model was smaller.

	Erlang C			Erlang A			Overall	
	Min	Avg	Max	Min	Avg	Max	Average	p value
Num Agents	15	65.0	100	10	49.8	100	55	2.72E-19
Utilization Target	65.02%	74.74%	90.73%	65.11%	82.75%	94.99%	80%	1.47E-48
Talk Time	2.05	10.46	19.92	2.01	11.28	19.99	11	.0172
Patience	61.4	342.5	597.6	60.3	323.5	599.7	330	.0673
AR CV	0.035	0.136	0.200	0.000	0.081	0.200	0.1	5.02E-52
Talk Time CV	0.753	1.008	1.250	0.750	0.996	1.249	1	.2313
Patience Shape	0.753	1.002	1.249	0.750	0.999	1.250	1	.7370
Probability of Balking	0.01%	11.02%	24.84%	0.06%	13.28%	24.99%	12.5%	2.36E-06
Agent Heterogeneity	0.000	0.071	0.150	0.000	0.077	0.150	0.075	.0339
Abandonment	0.01%	0.92%	3.99%	0.00%	3.17%	14.29%	2.40%	3.72E-50

Table 8 - Conditions of Model Accuracy

Table 8 summarizes this data. For each group it calculates the minimum, maximum, and average values of each design parameter in each group. It also presents the *p* value generated from a simple hypothesis test that the mean values of each group is the same. Factors such as caller patience, the shape of the patience distribution, and the variability of talk time have little impact on which model is more accurate. Agent Heterogeneity has a moderate impact, with higher levels of heterogeneity present in cases where Erlang A is more accurate. Similarly, moderately longer talk times are present in the Erlang A group.

The most significant factors are the number of agents in the call center, their utilization, the uncertainty of arrival rates, and the probably of balking. Erlang C tends to be more accurate in call centers with large pools of agents that are moderately utilized, and arrival rates that are less certain. As we have seen, arrival rate uncertainty tends to reduce the error in Erlang C predictions, while increasing the error in Erlang A predictions. Higher levels of balking tend to make the Erlang A model the preferred model. Looking at the realized abandonment rate, it is not surprising that Erlang A is more accurate when abandonment levels are high.

Summary and Conclusions

Erlang C is a model commonly applied to the analysis of call centers, and often used as the basis for determining staffing level requirements. Erlang C is a relatively simple model that makes many assumptions that are clearly suspect in the context of a call center. Many authors now advocate the use of the more realistic and more complex Erlang A model. We have conducted a comprehensive simulation analysis that shows that the Erlang C model is in fact subject to significant prediction error and that the Erlang A model is on average much more accurate under steady state conditions.

However, our analysis also finds that while the Erlang C is conservative, making pessimistic predictions most of the time, the Erlang A model often makes overly optimistic predictions. The optimistic bias of the Erlang A model is driven in large part by arrival rate uncertainty, a condition that somewhat paradoxically reduces the error of the Erlang C model. The results of our study suggest that care must be taken when using the Erlang A model to make staffing decisions; particularly in cases where arrival rates are subject to significant uncertainty and service level requirements are strict. Erlang C is the safer bet in that staffing based on an Erlang C prediction is more likely to result in the service level being met, even if the arrival rate is uncertain.

The analysis in this study is restricted to cases where the call center possesses the capacity to handle all calls presented; a requirement for the Erlang C model to have a defined output. In some call center environments staffing costs are significantly higher than the implied cost of customer delay and the number of agents is limited so that essentially all customers must wait before receiving service and agent utilization is close to 100%. This environment is sometimes referred to as the *efficiency-driven regime* (Gans et al., 2003). In this environment the offered utilization is in excess of 100%, so the Erlang C model becomes unstable and cannot be used to predict performance. The Erlang A model on the other hand allows for abandonment and can be used to predict performance in this regime. A separate study is warranted to determine the accuracy of the Erlang A model in this environment. Our analysis is also focused on long term average performance under steady-state conditions. We do not examine the impact of shifting arrival rates that occurs in real scenarios. In practice many call centers divide the day into a series of short intervals and assume that steady state is achieved in each period. A further study could examine the impact of this assumption on model accuracy.

References

Aksin, Z., Armony, M., & Mehrotra, V. (2007). The Modern Call-Center: A Multi-Disciplinary Perspective on Operations Management Research. *Production and Operations Management, 16*(6), 665-688.

Aktekin, T. (2014). Call center service process analysis: Bayesian parametric and semi-parametric mixture modeling. *European Journal of Operational Research, 234*(3), 709-719.

Aktekin, T., & Ekin, T. (2016). Stochastic call center staffing with uncertain arrival, service and abandonment rates: A Bayesian perspective. *Naval Research Logistics (NRL), 63*(6), 460-478.

Armony, M., & Ward, A. R. (2008). *Fair Dynamic Routing in Large-Scale Heterogeneous-Server Systems*. Working Paper. Stern School of Business, NYU.

Avramidis, A. N., Gendreau, M., L'Ecuyer, P., & Pisacane, O. (2007). *Simulation-Based Optimization of Agent Scheduling in Multiskill Call Centers.* Paper presented at the 2007 Industrial Simulation Conference.

Bassamboo, A., Harrison, J. M., & Zeevi, A. (2005). Design and Control of a Large Call Center: Asymptotic Analysis of an LP-based Method. *Operations Research, 54*(3), 419-435.

Borst, S., Mandelbaum, A., & Reiman, M. I. (2004). Dimensioning Large Call Centers. *Operations Research, 52*(1), 17-35.

Braverman, A., G. Dai, J., & Feng, J. (2016). *Stein's Method for Steady-state Diffusion Approximations: An Introduction through the Erlang-A and Erlang-C Models* (Vol. 6).

Brown, L., Gans, N., Mandelbaum, A., Sakov, A., Haipeng, S., Zeltyn, S., & Zhao, L. (2005). Statistical Analysis of a Telephone Call Center: A Queueing-Science Perspective. *Journal of the American Statistical Association, 100*(469), 36-50.

Chen, B. P. K., & Henderson, S. G. (2001). Two Issues in Setting Call Centre Staffing Levels. *Annals of Operations Research*(108), 175-192.

Feldman, Z., Mandelbaum, A., Massey, W. A., & Whitt, W. (2008). Staffing of Time-Varying Queues to Achieve Time-Stable Performance. *Management Science, 54*(2), 324-338.

Gans, N., Koole, G., & Mandelbaum, A. (2003). Telephone call centers: Tutorial, review, and research prospects. . *Manufacturing & Service Operations Management, 5*(2), 79-141.

Gans, N., & Zhou, Y.-P. (2007). Call-Routing Schemes for Call-Center Outsourcing. *Manufacturing & Service Operations Management, 9*(1), 33-51.

Garnett, O., Mandelbaum, A., & Reiman, M. I. (2002). Designing a Call Center with impatient customers. *Manufacturing & Service Operations Management, 4*(3), 208-227.

Green, L. V., Kolesar, P., & Soares, J. (2003). An Improved Heuristic for Staffing Telephone Call Centers with Limited Operating Hours. *Production and Operations Management, 12*(1), 46-61.

Green, L. V., Kolesar, P. J., & Soares, J. (2001). Improving the SIPP Approach for Staffing Service Systems That Have Cyclic Demands. *Operations Research, 49*(4), 549-564.

Halfin, S., & Whitt, W. (1981). Heavy-Traffic Limits for Queues with Many Exponential Servers. *Operations Research, 29*(3), 567-588.

Harrison, J. M., & Zeevi, A. (2005). A Method for Staffing Large Call Centers Based on Stochastic Fluid Models. *Manufacturing & Service Operations Management, 7*(1), 20-36.

Heching, A. R., & Squillante, M. S. (2014). Optimal capacity management and planning in services delivery centers. *Performance Evaluation, 80*, 63-81.

Huang, J., Mandelbaum, A., Zhang, H., & Zhang, J. (2017). Refined Models for Efficiency-Driven Queues with Applications to Delay Announcements and Staffing. *Operations Research, 65*(5),

Ibrahim, R., & Whitt, W. (2008, 7-10 Dec. 2008). *Real-time delay estimation in call centers.* Paper presented at the 2008 Winter Simulation Conference.

Janssen, A. J. E. M., van Leeuwaarden, J. S. H., & Zwart, B. (2011). Refining Square-Root Safety Staffing by Expanding Erlang C. *Operations Research, 59*(6), 1512-1522.

Jennings, O. B., Mandelbaum, A., Massey, W. A., & Whitt, W. (1996). Server Staffing to Meet Time-Varying Demand. *Management Science, 42*(10), 1383-1394.

Knessl, C., & van Leeuwaarden, J. S. H. (2015). Transient analysis of the Erlang A model. *Mathematical Methods of Operations Research, 82*(2), 143-173. doi:10.1007/s00186-015-0498-9

L'Ecuyer, P. (1999). Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators. *Operations Research, 47*(1), 159-164.

Law, A. M. (2007). *Simulation modeling and analysis* (4th ed.). Boston: McGraw-Hill.

Lima, M. A. d. Q. V., Maciel, P. R. M., Silva, B., & Guimarães, A. P. (2014). Performability evaluation of emergency call center. *Performance Evaluation, 80*, 27-42.

Mandelbaum, A., & Zeltyn, S. (2004). *Service Engineering in Action: The Palm/Erlang-A Queue, with Applications to Call Centers Draft, December 2004*.

Mandelbaum, A., & Zeltyn, S. (2009). *The M/M/n+G Queue: Summary of Performance Measures*. Technical Note, Technion, Israel Institute of Technology.

Mandelbaum A., Sakov A. , & S., Z. (2001). *Empirical Analysis of a Call Center*. Technion - Israel Institute of Technology.

Palm, C. (1957). Research on telephone traffic carried by full availability groups. *Tele, 1*, 107.

Phung-Duc, T., & Kawanishi, K. i. (2014). Performance analysis of call centers with abandonment, retrial and after-call work. *Performance Evaluation, 80*, 43-62.

Robbins, T. R. (2007). *Managing Service Capacity Under Uncertainty - Unpublished PhD Dissertation* http://myweb.ecu.edu/robbinst/PDFs/Dissertation%20Final.pdf Pennsylvania State University- Smeal College of Business.

Robbins, T. R., & Harrison, T. P. (2010). A stochastic programming model for scheduling call centers with global Service Level Agreements. *European Journal of Operational Research, 207*, 1608-1619.

Robbins, T. R., Medeiros, D. J., & Dum, P. (2006). *Evaluating Arrival Rate Uncertainty in Call Centers.* Paper presented at the 2006 Winter Simulation Conference, Monterey, CA.

Robbins, T. R., Medeiros, D. J., & Harrison, T. P. (2010). *Does the Erlang C model fit in real call centers?* Paper presented at the 2010 Winter Simulation Conference, Austin, TX.

Roubos, D., & Bhulai, S. (2010). Approximate dynamic programming techniques for the control of time-varying queuing systems applied to call centers with abandonments and retrials. *Probab. Eng. Inf. Sci., 24*(1), 27-45.

Santner, T. J., Williams, B. J., & Notz, W. (2003). *The design and analysis of computer experiments*. New York: Springer.

Sivan, A.-N., Paul, D. F., & Avishai, M. (2009). WORKLOAD FORECASTING FOR A CALL CENTER: METHODOLOGY AND A CASE STUDY (Vol. 3, pp. 1403-1447).

Steckley, S. G., Henderson, S. G., & Mehrotra, V. (2009). Forecast Errors in Service Systems. *Probability in the Engineering and Informational Sciences*(23), 305-332.

Steckley, S. G., Henderson, W. B., & Mehrotra, V. (2004). *Service System Planning in the Presence of a Random Arrival Rate*. Working paper Cornell University.

Wallace, R. B., & Whitt, W. (2005). A Staffing Algorithm for Call Centers with Skill-Based Routing. *Manufacturing & Service Operations Management, 7*(4), 276-294.

Whitt, W. (2005). Engineering Solution of a Basic Call-Center Model. *Management Science, 51*(2), 221-235.

Whitt, W. (2006a). Fluid Models for Multiserver Queues with Abandonments. *Operations Research, 54*(1), 37-54.

Whitt, W. (2006b). Sensitivity of Performance in the Erlang A Model to Changes in the Model Parameters. *Operations Research, 54*(2), 247-260.

Whitt, W. (2006c). Staffing a Call Center with Uncertain Arrival Rate and Absenteeism. *Production and Operations Management, 15*(1), 88-102.