# A DECISION SUPPORT FRAMEWORK FOR DESIGNING MULTI-SERVER QUEUES WITH FINITE CAPACITIES

Daniel S. Myers, Rollins College, Winter Park, FL, dmyers@rollins.edu

## ABSTRACT

This work presents a decision support framework for solving design problems in multi-server finite-buffer queuing systems with random arrivals. The key to the proposed approach is a collection of new, accurate two-moment approximations for the queue lengths, drop probabilities, and average customer residence times in the M/G/$c$/$k$ queue. These approximations are shown to be accurate for systems of practical interest, including queues with many servers, high utilizations, and high service time variability. The complete decision support framework combines these approximations with a mathematical programming formulation of the M/G/$c$/$k$ design problem.

## INTRODUCTION

Queueing systems with multiple servers and finite-capacity buffers occur in a wide variety of applications, including manufacturing, telecommunications, transportation, and service organizations. This paper presents a decision support framework for solving design problems in multi-server finite-buffer queuing systems with random arrivals. The key to the proposed approach is a collection of new, accurate two-moment approximations for the queue lengths, drop probabilities, and average customer residence times in the M/G/$c$/$k$ queue. Through comparisons to simulation, these approximations are shown to be accurate for a range of systems of practical interest.

Previous work has investigated approximations for multi-server queueing systems—for example, (Kimura, 1994; Kingman, 1962; Nozaki and Ross, 1978; Sakasegawa, 1977; Tijms, 1992)—and a range of problem formulations and solution approaches for finding good configurations in different contexts; (Smith, 2007) provides a survey of several relevant articles. This work proposes a set of new, robust approximations, building on previous research into queue length approximations (Myers and Vernon, 2012), and shows how these techniques may be combined with a general mathematical programming framework to provide decision support for system designers.

The rest of this article makes the following contributions. First, a formulation of a general design problem for the M/G/$c$/$k$ queue as a nonlinear integer mathematical programming problem. Second, presentation of new two-parameter approximation for the queue length distributions in M/G/$c$ queueing systems. Third, drawing on a result of Tijms (Tijms, 1992), an extension of the M/G/$c$ queue length approximation to derive and validate formulas for the drop probability and average residence times in the M/G/$c$/$k$ queueing system. Finally, presentation of an example design problem, including a search-based solution strategy.

## A DESIGN FRAMEWORK FOR M/G/$c$/$k$ QUEUEING SYSTEMS

Consider an M/G/$c$/$k$ queue having $c$ servers and a general service time distribution with mean $\bar{x}$ and standard deviation $\sigma$. The service distribution's coefficient of variation is $v_x = \sigma/\bar{x}$. The system has total capacity $k$, including the $c$ servers. Arriving customers that find the system full

(with a total of $k$ already waiting and in service) are *dropped* and must exit immediately without receiving service. The queue receives a random (Poisson) arrival stream at a rate $\lambda$. The server utilization (the fraction of time the server is busy) is $\rho = \lambda \bar{x}$.

Consider the problem of parameterizing the queue by determining the number of servers $c$ and the total number of allocated buffer spaces $b = k - c$. Three goals are readily apparent: first, minimizing the total cost of provisioning the queue's servers and buffer space; second, achieving a target throughput rate of served customers, which implies a bound on the fraction of dropped customers; and third, maintaining an acceptable bound on customers' expected residence times.

Let the cost of each server be $\alpha$ and the cost of each allocated buffer space be $\beta$. A straightforward linear cost function for provisioning the system with $c$ servers and total capacity $k$ is

$$z = \alpha c + \beta(k - c) \tag{1}$$

Suppose $f^{drop}(c,k)$ yields the proportion of dropped customers when the queue has $c$ servers and total capacity $k$. If the arrival rate is $\lambda$, then the rate at which non-dropped customers receive service and exit the system is

$$\Lambda = (1 - f^{drop}(c,k))\,\lambda \tag{2}$$

Let $\overline{R}(c,k)$ represent the expected total residence time experienced by a customer who enters the queue without being dropped, inclusive of both waiting and service times.

Combining these design goals yields a nonlinear integer mathematical programming problem:

$$\min_{c,\,k} \ \alpha c + \beta(k - c) \tag{3}$$

subject to:

$$(1 - f^{drop}(c,k))\,\lambda \geq \Lambda_{min}$$
$$\overline{R}(c,k) \leq \overline{R}_{max}$$
$$c,\ k \text{ positive integers}$$

The objective minimizes the total cost of parameterizing the queue. The first constraint ensures that the system's throughput is at least $\Lambda_{min}$ and the second ensures that the expected customer residence time is no more than $\overline{R}_{max}$. Many variations can be incorporated into this basic model. The rest of this paper assumes the basic model of (3), but the solution strategies we discuss can be adapted to work with additional constraints or a modified objective function.

The major practical consideration in solving (3) is obtaining accurate estimates of the drop probabilities $f^{drop}(c,k)$ and the expected residence times $\overline{R}(c,k)$. This may be done by simulation, but that requires a potentially time-consuming step embedded within every iteration of the solution algorithm. Therefore, there is an advantage in developing *analytic* formulas for the drop probabilities and residence times that can be incorporated into numerical solution algorithms.

## TWO-MOMENT APPROXIMATIONS

### M/G/1 Queue Length

Lret $\pi_n$ denote the probability that the queue contains $n$ customers. Myers and Vernon established that the following approximation is accurate for a wide range of M/G/1 systems having one server and theoretically infinite capacity (Myers and Vernon, 2012):

$$\pi_n \approx \rho\, q^{j-1}\,(1-q) \qquad n \geq 1 \tag{4}$$

where

$$q = \frac{\lambda \bar{r}}{1+\lambda \bar{r}-\rho} = \frac{\rho(v_x^2+1)}{2+\rho(v_x^2-1)} \tag{5}$$

and $\bar{r}$ is the expected residual life of a customer in service,

$$\bar{r} = \frac{\overline{x}}{2}(1+v_x^2) \tag{6}$$

This formula is exact for M/M/1 systems: setting $v_x = 1$ gives $q = \rho$ and (4) reduces to the standard M/M/1 length distribution:

$$\pi_n = \rho^n(1-\rho) \tag{7}$$

### M/G/$c$ Queue Length

It can be shown that the values of $\pi_n$ for $n > c-1$ in the M/M/$c$ system are given by

$$\pi_n = \left(1 - \sum_{i=0}^{c-2} \pi_i\right)\, \rho^{n-(c-1)}\,(1-\rho) \tag{8}$$

where $\rho$ is the individual server utilization given by $\rho = \lambda/(c\,\mu)$. The equation resembles the standard length distribution for M/M/1; the first term is the probability that at least $c-2$ servers are occupied. This observation, which is exact for M/M/$c$, suggests an approach for approximating length probabilities in M/G/$c$: assume that the tail of the M/G/$c$ length distribution obeys the modified geometric scaling of (4) and substitute the scaling term $q$ from (5) for $\rho$.

$$\pi_n \approx \left(1 - \sum_{i=0}^{c-2} \pi_i\right)\, \rho\; q^{n-(c-2)}\,(1-q) \tag{9}$$

This approximation is valid for the tail probabilities, $n > c-1$. In the case where there are $c-1$ customers (corresponding to 1 empty server), equation (8) gives

$$\pi_{c-1} \approx \left(1 - \sum_{i=0}^{c-2} \pi_i\right)\, (1-\rho) \tag{10}$$

These equations require estimates of $\pi_0$ to $\pi_{c-2}$, which correspond to the number of occupied servers and do not involve any waiting customers. Therefore, it is reasonable to believe that the queue length probabilities for these states are primarily a function of the individual server uti-

lizations, even in cases where the service times are not exponentially distributed. Equation (4) is exact for systems with exponentially distributed service times, so it is reasonable to choose values that make equation (9) exact for M/M/$c$ systems. The relevant M/M/$c$ queue length distribution is (Harchol-Balter, 2013):

$$\pi_n = \pi_0 \frac{(c\rho)^n}{n!} \tag{11}$$

when $k \leq c$. The probability of finding the system empty is

$$\pi_0 = \left[ \sum_{j=0}^{c-1} \frac{(c\rho)^j}{j!} + \left( \frac{(c\rho)^c}{c!} \right) \left( \frac{1}{1-\rho} \right) \right]^{-1} \tag{12}$$

To estimate the M/G/$c$ length probabilities, first estimate the probabilities for $n \leq c-2$ using (11) and (12), then calculate approximate probabilities for $n \geq c-1$ using (9) and (10). Finally, normalize the estimates to ensure the approximate distribution sums to 1.

Figure 1 compares the performance of the M/G/$c$ length approximation against the results of simulation. Each figure plots the estimated 90th and 99th percentiles of the queue length distribution for increasing numbers of servers in systems with high variability hyperexponential service times ($v_x^2 = 20$) and utilizations of 75% and 95%, representing systems with moderate and heavy load. For clarity, the simulated reference queue lengths are plotted without error bars. Each simulation collected a sufficient number of observations to yield an error of 1% or less in its estimate.

The plots show that the M/G/$c$ length approximation accurately approximates the both the 90th and 99th percentiles of queue length. The highest errors occur at the 70% utilization levels, where the approximation overestimates the 99th percentile of queue length by as much as 8 positions—this is still less than a 15% error, which is accurate enough to support design insights. Comparisons against lower variability service times (not shown) are more accurate: corresponding results for a deterministic system are accurate to within a single queue position.

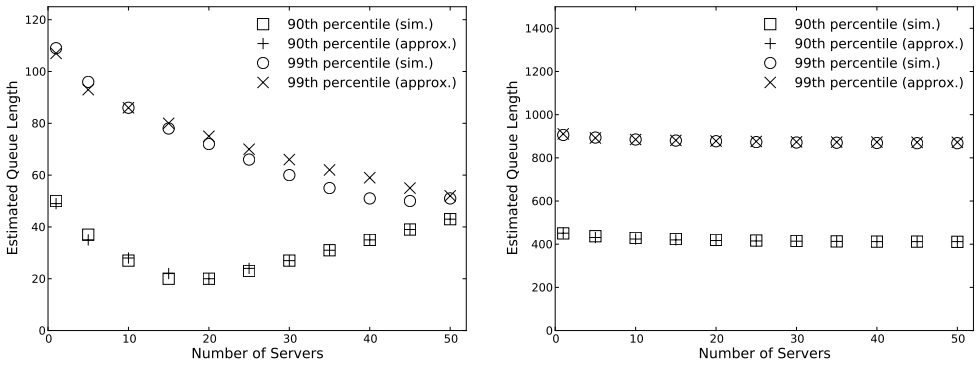
Figure 2 plots the 90th and 99th percentiles of queue length as a function of server utilization in a system with five servers. The plots show that the approximation is accurate across a range of utilization levels for both the zero-variability deterministic service distribution and the high-variability hyperexponential distribution. These results are representative of those obtained for systems with more than 5 servers.

**M/G/$c$/$k$ Drop Probability**

Tijms proposed an approximation for the drop probability in M/G/$c$/$k$ queues (Tijms, 1992):

$$f^{drop}(c,k) \approx \frac{(1-\rho)\left(1-\sum_{j=0}^{k-1}\pi_j^{(\infty)}\right)}{1-\rho\left(1-\sum_{j=0}^{k-1}\pi_j^{(\infty)}\right)} \tag{13}$$

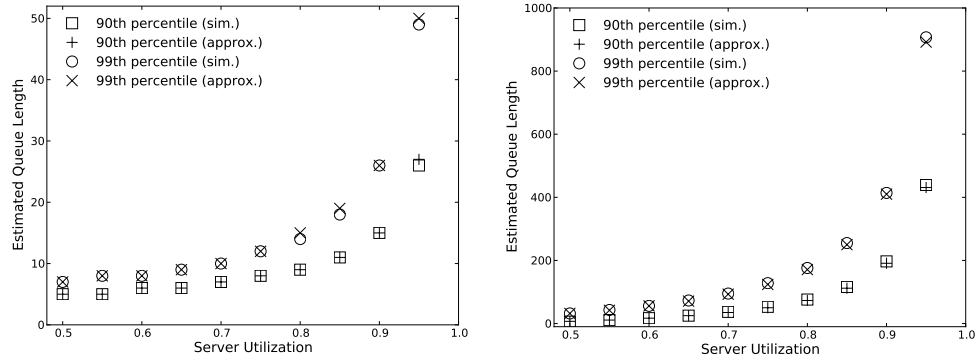
where $\pi_j^{(\infty)}$ is the probability of $j$ customers in the corresponding infinite-buffer queue and $\rho = \lambda/(c\mu)$ is the per-server traffic intensity. This formula is exact for M/M/$c$/$k$ and M/G/1/1 systems.




(a) 70% server utilization

(b) 95% server utilization

Figure 1: Percentiles of queue length vs. servers, hyperexponential service times ($v_x^2 = 20$)




(a) Deterministic

(b) Hyperexponential ($v_x^2 = 20$)

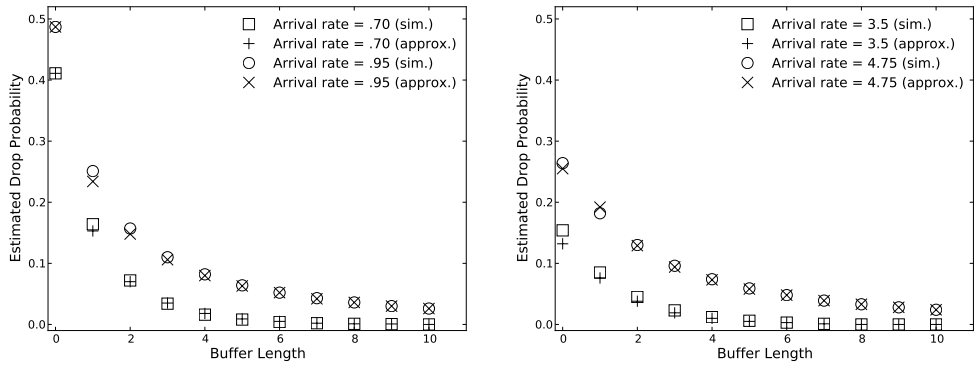
Figure 2: 90th and 99th percentiles of queue length vs. server utilization, 5 servers

Because excess customers can be rejected by the system, the offered per-server load $\rho$ can be greater than 1 in M/G/$c$/$k$ systems. However, because $\pi_j^{(\infty)}$ is only defined when the infinite-buffer system is stable, (13) is only valid when $\rho < 1$.

We propose to use the M/G/$c$ queue length approximation to obtain $\pi_j^{(\infty)}$ and equation (13) to estimate the drop probability. Figures 3 and 4 validate this approach. Each figure plots the simulated and approximated drop probabilities at 70% and 95% server utilization in systems with 1 and 5 servers. In the deterministic case (Figure 3), the approximation is consistently within 1% of the simulated estimate. In the hyperexponential case (Figure 4), the largest error is only 3% and the majority of points are within 1%.

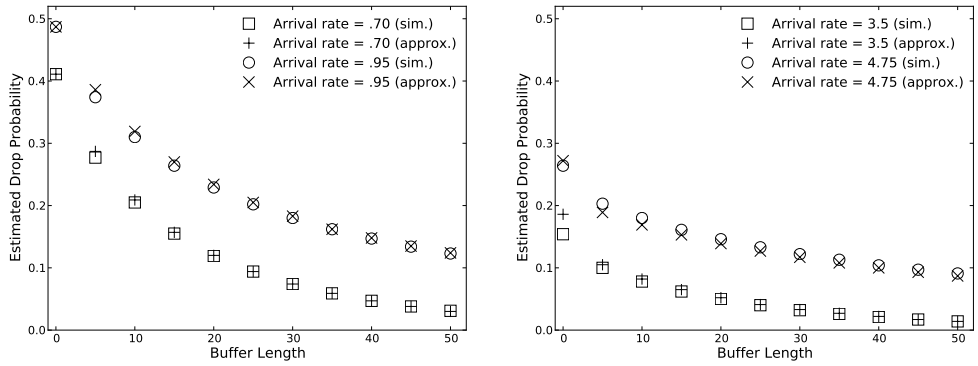
### M/G/$c$/$k$ Average Residence Time

Recall that the average system throughput is given by $\Lambda = (1 - f^{drop}(c,k))\,\lambda$, which can now be estimated using equation (13). Therefore, given an approximation for the average occupancy $\overline{N}$, we can determine the average residence time $\overline{R}$ using Little's result, $\overline{N} = \Lambda\overline{R}$ (Little, 1961). The




(a) 1 server (b) 5 servers

Figure 3: Drop probability vs. buffer length, deterministic service times




(a) 1 server (b) 5 servers

Figure 4: Drop probability vs. buffer length, hyperexponential service times ($v_x^2 = 20$)

average system occupancy in the M/G/$c$/$k$ system is, by definition,

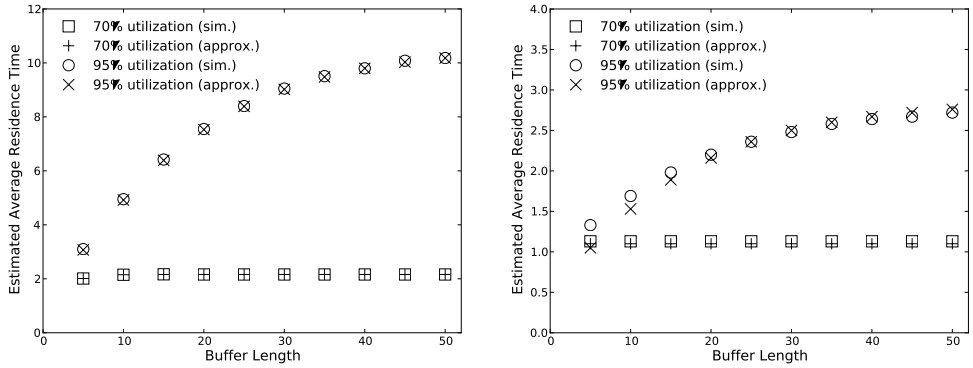
$$\overline{N} = \sum_{j=0}^{k} j\,\pi_j \tag{14}$$

To estimate $\overline{N}$, assume that the finite-buffer queue length probabilities are related to the corresponding infinite buffer system by the scaling term $q$:

$$\pi_n \approx \frac{\pi_n^{(\infty)}}{1-q^{k+1}} \tag{15}$$

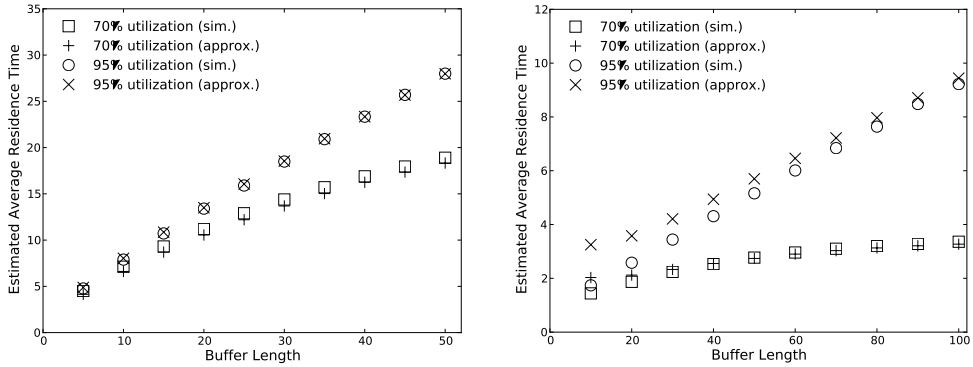
This approximation, which is exact for M/M/1/$k$, assumes that queue lengths in the M/G/$c$/$k$ system behave like a M/G/1/$k$ system with service rate $c\mu$. This is somewhat ad-hoc, but is reasonably accurate if the buffer size is not too small.

Figures 5 and 6 compare the residence time predictions produced by this method to simulation estimates in example systems with 1 and 5 servers; these results are representative of other server




(a) 1 server (b) 5 servers

Figure 5: Expected residence time vs. buffer length, deterministic service times




(a) 1 server (b) 5 servers

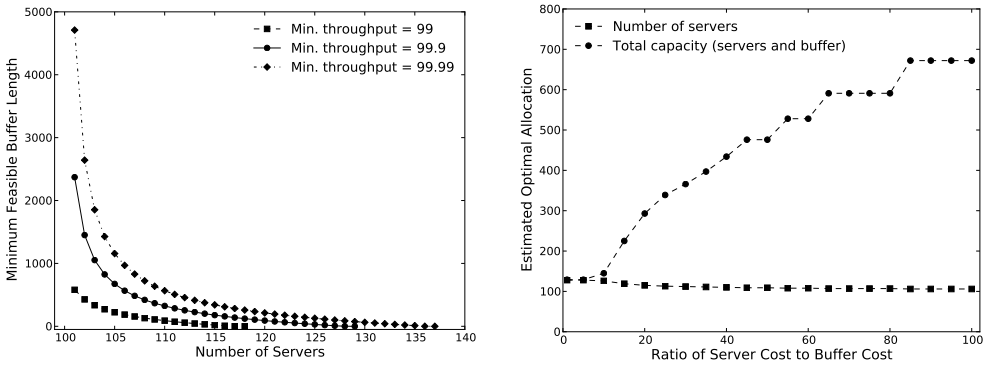
Figure 6: Expected residence time vs. buffer length, hyperexponential service times ($v_x^2 = 20$)

configurations. The residence time approximation is accurate in both single-server systems, but has moderate errors in multi-server systems with small buffers. For example, in the hyperexponential system with 5 servers and one buffer space, the approximation predicts a residence time of 3.25 at the 95% utilization level, compared to a simulated estimate of only 1.74. The accuracy improves quickly as the buffer size increases, consistent with the observation that an M/G/$c$/$k$ system with large $k$ behaves more like a single-server system from the perspective of customers in the queue.

## AN EXAMPLE DESIGN PROBLEM

This section applies the results of the previous sections to an example design problem: the arrival rate is $\lambda = 100$ customers per second; customers' service is distributed according to a two-stage hyperexponential distribution with $v_x^2 = 20$ and average service rate $\mu = 1$; the required average residence time is $\overline{R}_{max} = 10$; the cost of a buffer space $\beta$ is normalized to 1.

Figure 7(a) plots the boundaries of the space of feasible system designs for three values of the required throughput $\Lambda_{min}$ corresponding to loss rates of 1%, .1% and .01%, respectively. Increasing the buffer length increases the rate of successfully served customers $\Lambda$ because fewer customers




(a) Feasible buffer lengths vs. servers for three service levels

(b) Estimated optimal $(c,k)$ allocations for $\Lambda_{min} = 99.9$

Figure 7: Results for an example design problem

will be dropped, but increases $\overline{R}$, since arriving customers must wait, on average, in a longer line.

The optimal configuration must be one of the points on the boundary of the feasible region, because any point on the interior is strictly dominated by neighbor that also satisfies the constraints but allocates fewer buffer spaces. Therefore, it is straightforward to find an optimal configuration by using a binary search to find the minimum feasible buffer length for increasing numbers of servers $c$. The search ends upon finding a configuration that satisfies the constraints using only servers and no additional buffer space.

Figure 7(b) plots the estimated optimal solutions obtained by search for $\Lambda_{min} = 99.9$ against increasing values of the server cost $\alpha$. The plot shows both the number of servers $c$ in each configuration and the total system capacity $k$, which includes both servers and buffer space. When the server cost is low, the optimal strategy is to simply allocate enough servers to meet the constraints with no additional buffer space. As the server cost increases, the best configuration is one with the minimum number of servers that can satisfy the residence time bound and the minimum buffer length that satisfies the throughput requirement.

## RELATED WORK

Given the analytical challenges of multi-server queues with non-exponential service times, there is a long history of bounds and approximations for these systems; the result of Tijms (Tijms, 1992) is one example. Kimura provides a survey of several analytic approximations in multi-server systems with both finite and infinite buffers in (Kimura, 1994). There is also a long history of designing optimal allocation policies for queues in particular applications. A survey of classical work on design and parameterization is (Crabill et al., 1977). In the context of this work, (Smith, 2007) is highly relevant. He investigates a collection of optimization problems for a system of parallel M/G/$c$/$k$ queues and proposes an alternate approximation for the drop probability, based on a different result of Tijms, Tijms (1986), which interpolates between the blocking probabilities for the M/D/1/$k$ and M/M/1/$k$ system. The resulting technique does not have a convenient closed form expression, though it is suitable for numerical evaluation. Smith uses this approximation

to investigate the characteristics of several optimization problems that are similar to (3). The approximations in this paper can be expressed conveniently in closed form, and have been validated against both low- and high-variability service time distributions.

## CONCLUSION

This work has presented a practical decision support framework for capacity planning problems in multi-sever queues with finite storage capacity. The proposed framework combines a mathematical programming formulation of the design problem with a set of novel two-moment approximations for important measures in the M/G/$c$/$k$ queue. Comparisons to simulation have shown that these approximations are accurate for many systems of practical interest. There are several opportunities for future projects in this area. In particular, we are interested in applying the two-moment approximations and decision support framework to real-world capacity planning problems and developing an open-source software implementation.

## REFERENCES


Thomas B Crabill, Donald Gross, and Michael J Magazine. A classified bibliography of research on optimal design and control of queues. *Operations Research*, 25(2):219–232, 1977.

M. Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

Toshikazu Kimura. Approximations for multi-server queues: System interpolations. *Queueing Systems*, 17(3-4):347–382, 1994.

JFC Kingman. On queues in heavy traffic. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 383–392, 1962.

John D.C. Little. A proof for the queuing formula: L= $\lambda$ w. *Operations research*, 9(3):383–387, 1961.

Daniel S Myers and Mary K Vernon. Estimating queue length distributions for queues with random arrivals. *ACM SIGMETRICS Performance Evaluation Review*, 40(3):77–79, 2012.

Shirley A Nozaki and Sheldon M Ross. Approximations in finite-capacity multi-server queues by poisson arrivals. *Journal of Applied Probability*, 15(4):826–834, 1978.

Hirotaka Sakasegawa. An approximation formula $L_q \approx \alpha \cdot \rho^{\beta}/(1-\rho)$. *Annals of the Institute of Statistical Mathematics*, 29(1):67–75, 1977.

J MacGregor Smith. Multi-server, finite waiting room, m/g/c/k optimization models. *INFOR: Information Systems and Operational Research*, 45(4):257–274, 2007.

H.C. Tijms. *Stochastic Modelling and Analysis: A Computational Approach*. John Wiley and Sons, Great Britain, 1986.

H.C. Tijms. Heuristics for finite-buffer queues. *Probability in the Engineering and Informational Sciences*, 6(03):277–285, 1992.