Methodology and Computing in Applied Probability (2022) 24:1297–1321
https://doi.org/10.1007/s11009-021-09848-8

Difference Equations Approach for Multi-Server Queueing Models with Removable Servers

James J. Kim [1] · Douglas G. Down [2] · Mohan Chaudhry [3] · Abhijit Datta Banik [4]

Received: 4 January 2020 / Revised: 5 January 2021 / Accepted: 7 January 2021 /
Published online: 1 May 2021

Abstract
We consider an extended form of the $M^X/M/c$ queue with two types of server groups: Static as well as dynamic (which turn on/off in a state-dependent manner) servers. The two server groups may have homogenous or non-homogenous service rates. The model is further extended to feature setup and delayed-off times, finite capacity, and k staffing levels. This class of queues is solved via the difference equations approach, which addresses narratives in the literature and achieves higher numerical efficiency than the direct method. While the model of this queueing system is not new, the methodology for solving it is. Comparisons between our model and classic queues are provided followed by concluding remarks, including a summary of key observations.

Keywords Multi-server · Dynamic servers · Setup · Delayed-off · k staffing · Difference equations · System performance · Resource consumption

✉ James J. Kim
s25412@rmc.ca

Douglas G. Down
downd@mcmaster.ca

Mohan Chaudhry
chaudhry-ml@rmc.ca

Abhijit Datta Banik
adattabanik@iitbbs.ac.in

[1] Royal Canadian Air Force (RCAF), Ottawa, ON, Canada

[2] Department of Computing and Software, McMaster University, Hamilton, ON, Canada

[3] Department of Mathematics and Computer Science, Royal Military College of Canada, Kingston, ON, Canada

[4] School of Basic Sciences, Indian Institute of Technology, Bhubaneswar, India

1 Introduction

Until the mid-sixties, a queueing problem was considered solved if the solution was given in some form of a generating function or Laplace transform. This is because the inversion of a generating function was either considered unnecessary or a trivial problem. However, the inversion of transforms arising in queueing and other stochastic models (except for simple cases) is not as easy as it was once thought and hence such results can be difficult to exploit in solving practical problems. Many users expressed concerns that such solutions were inadequate. Kendall (1964) made a famous remark that queueing theory wears the Laplacian curtain. Kleinrock (1975) states "one of the most difficult parts of this method of spectrum factorization is to solve for the roots." In a similar account, Neuts (see Stidham Jr. (2001)) states "In discussing matrix-analytic solutions, I had pointed out that when the Rouché's roots coincide or are close together, the method of roots could be numerically inaccurate. When I finally got copies of Crommelin's papers, I was elated to read that, for the same reasons as I, he was concerned about the clustering of roots. In 1932, Crommelin knew; in 1980, many of my colleagues did not...." In this connection, see also Neuts (1981).

The preconceived notion of the above said risks associated with root-finding has carried through to the modern literature of queueing theory. In 2005, Mejia-Téllez makes the following statement: "If the batch size is large, the determination of these roots is difficult…." In a recent paper by Bar-Lev et al. (2007) that analyzes the $M/G^{(m,M)}/1$ queue, they introduce their model's characteristic equation as $A^{(M)}(z) - z^M$ where M is the maximum batch size. The polynomial $A^{(M)}(z)$ is the probability generating function (p.g.f) of the random variable $Y_n^{(M)}$ which corresponds to the number of job arrivals during the bulk-service period of M jobs from the n-th batch arrival. Bar-Lev et al. (2007) state that "this general solution requires the calculation of the zeros of $A^{(M)}(z) - z^M$ which in practice can result in numerical inaccuracies especially when the decision variable M assumes a large value…." In a recent work by Harris et al. (2000), they state that "the standard root-finding problem gets complicated particularly when the inter-arrival time distribution possesses a complicated non-closed form or non-analytic Laplace-Stieltjes transform (L-ST)."

It is evident that the idea of root-finding, an imperative step in inverting a generating function, continues to be dismissed by a large body of researchers based on the perceived risk of numerical inaccuracies and previous remarks made by prominent figures. New methodologies have emerged as workaround solutions. These include numerical convolutions (say, R^{*n} which is not easy to calculate for large n), the matrix-analytic method (which simplifies to a matrix-geometric method when the underlying distributions are of phase type (Neuts 1981)), not to mention different iterative algorithms or approximation methods. Abate and Whitt (1992) use a Fourier-series method to numerically invert generating functions as well as Laplace transforms. The Fourier-series method involves numerically integrating the transform by means of the trapezoidal rule. The greatest difficulty in this case is approximately calculating the infinite series obtained from the inversion integral.

Historically, when numerical software such as MAPLE and Mathematica could not find a large number of roots (they do now), a software package called QROOT developed by Chaudhry (1991) was used by him and his collaborators to find the roots and use them in solving several queueing models. The algorithm for finding such roots is available in some of their papers. It may be remarked here that MAPLE can now not only find roots that are close to each other (a concern expressed by several researchers) but even repeated roots. As root-finding algorithms continued

to be refined, several researchers revisited the problem of inverting a generating function via root-finding. Gouweleeuw (1996) states that it is more efficient to use the roots method to get explicit expressions for probabilities from generating functions. Similarly, Janssen and van Leeuwaarden (2005), who have successfully used the roots method, make the comment, "initially, the potential difficulties of root-finding were considered to be a slur on the unblemished transforms since the determination of the roots can be numerically hazardous and the roots themselves have no probabilistic interpretation. However, Chaudhry et al. (1990) have made every effort to dispel the skepticism towards root-finding in queueing theory." Daigle and Lucantoni (1991) state that "whenever the roots method works, it works blindingly fast."

The procedure and results of root-finding are found to be efficient and accurate by those who advocate the use of roots, and therefore, improve the generating function method. However, despite the availability of roots, having to construct and invert a generating function remains a laborious exercise. Such issues for the use of generating functions in solving multi-server queues are noted by Chaudhry and Kim (2016) who, in reviewing the work by Zhao (1994) on the $GI^X/M/c$ queue, write "despite the detailed analysis, his derivations to construct the p.g.f.'s and their inversions are evidently lengthy and consider several conditions that can be avoided." In solving the $M/M/c/setup$ queue, Gandhi et al. (2014) state that "generating function approaches involve guessing the form of the solution and then solving for the coefficients of the guess, often leading to long computations." Nevertheless, multi-server queues with removable servers (including the $M/M/c/setup$ queue) form an important class of queues that can be used in a wide variety of contexts. Besides the applications in modeling data centres (see Krioukov et al. (2010), Qin and Wang (2007), Horvath and Skadron (2008), Maccio and Down (2015), Phung-Duc (2015), etc.), there have been applications in retail service facilities (see Berman and Larson (2004), and Terekhov and Beck (2009)), and border-crossing stations (see Zhang (2009)). In our survey of the literature on multi-server queues with removable servers, we have concluded that the generating function approach is often disadvantaged over other methods such as the Recursive Renewal Reward (RRR) method, the matrix-analytic method, and recursive methods.

While such limitations of the generating function approach are widely acknowledged in the literature, it is also our opinion that the inconveniences of the generating function approach can be remedied by an alternate approach. Instead of formulating the generating function (i.e. the z-transform of the set of balance equations), we interpret a subset of the balance equations of the model as a set of difference equations. How we proceed to choose such a subset is illustrated in an intuitive manner. By leveraging the properties of the difference equations we are able to give a solution of a general form in terms of the roots of the model's characteristic equation. In essence, we achieve the solution in an explicit form in a straightforward manner instead of formulating and then inverting a generating function that leads to the same solution. The solution and its coefficients are entirely in terms of roots hence finding such roots is an essential step in our methodology. Once the roots are found, the coefficients can be easily computed.

The purpose of this paper is to demonstrate that our method, the difference equations approach (and therefore the use of roots), can effectively solve an advanced form of multi-server queues with removable servers. The paper is organized in the following manner: We first introduce the baseline model followed by various extensions that have either frequently appeared or are likely to be of interest (bulk arrival, non-homogenous servers, setup and delayed-off times, finite capacity, and multiple staffing levels). Each model has a unique 'root equation' which provides the required roots to find the steady-state distribution. This work is followed by the introduction of performance measures and then comparisons against the traditional queue (i.e. the model $M^X/M/c$). While we conclude that the method used here is

analytically simple and numerically efficient (when compared against the direct method), it is our hope that the difference equations approach can be considered as a useful tool in analyzing other variants of multi-server queueing control problems.

2 The Baseline Model: The $M^X/M/c + l/(m, n)$ Queue

Consider the $M^X/M/c$ queue with two types of servers: there are c static servers (with a common service rate μ) which remain turned on at all times. As well, there are l dynamic servers (with a common service rate μ_1) that immediately turn on whenever the number of jobs in the system reaches or exceeds an upper-threshold n and are immediately turned off whenever the number of jobs in the system falls to or below a lower-threshold m. We assume the relation between lower and upper bounds $(c + l \le m \le n - 2)$ is such that all l dynamic servers are turned off immediately whenever the number of jobs in the system becomes $c + l$ or smaller. Upon turning off the l dynamic servers, any jobs being served by those l dynamic servers rejoin the front of the queue. When $\mu_1 = \mu$, the l servers are called homogenous dynamic servers (and non-homogenous dynamic servers when $\mu_1 \neq \mu$).

Jobs arrive in batches of size X and the inter-batch arrival time distribution is exponential with rate λ. The maximum batch size is r, $(r < +\infty)$ and the batch-size distribution is $b_h = P(X = h)$, $(1 \le h \le r)$ with mean $E[X]$. We assume that jobs are served in a First-Come-First-Served (FCFS) manner. The traffic intensity of the model is $\rho = \frac{\lambda E[X]}{c\mu + l\mu_1} < 1$. We refer to this model as the baseline model or in an adaptation of Kendell's notation, the $M^X/M/c + l/(m, n)$ queue.

2.1 Balance Equations

Let $J(t)$ and $S(t)$ denote the number of jobs in the system and the state of servers, respectively, at time t. We form a Markov chain $\{X(t) = (J(t), S(t)); t \ge 0\}$ on the state space $\varphi = \{(i, s); i \ge 0, s = 0, 1\}$ where $s = 0$ and $s = 1$ indicate that the l dynamic servers are turned off and turned on, respectively. See Fig. 1 below for a simple example of transitions among different states.

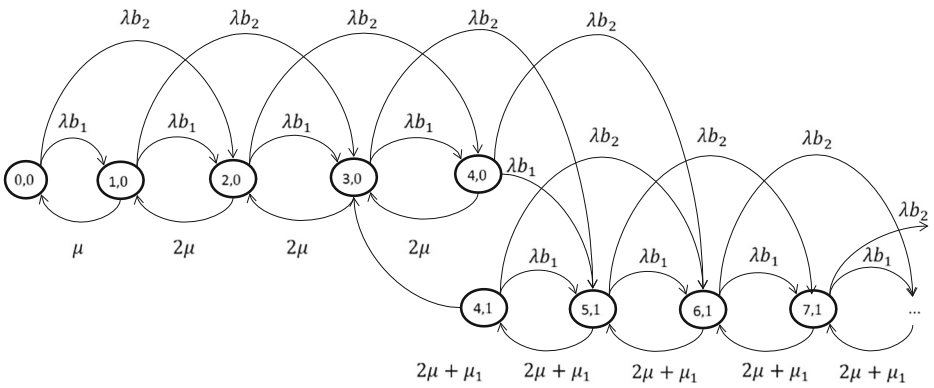


Fig. 1 Transition diagram of the baseline model when $c = 2$, $l = 1$, $r = 2$, $m = 3$,and $n = 5$

Let $\pi_{i,s} = \lim_{t\to\infty} P\{J(t) = i, S(t) = s\}$, $(i,s)\in\varphi$ be the joint steady-state distribution of $\{X(t)\}$. Note that $\pi_{i,\,s} > 0$ for the following regions of (i,s): $(0 \le i \le n-1, s = 0)$ and $(i \ge m+1, s = 1)$, and $\pi_{i,\,s} = 0$ otherwise. In this section of the paper we solve for $\pi_{i,\,s}$ using the roots method. The system dynamics can be described in terms of the following balance equations:

$$\lambda\pi_{0,0} = \mu\pi_{1,0} \tag{1}$$

$$(\lambda + i\mu)\pi_{i,0} = \lambda \sum_{h=1}^{min(i,r)} b_h\pi_{i-h,0} + (i+1)\mu\pi_{i+1,0}, (1\le i\le c-1) \tag{2}$$

$$(\lambda + c\mu)\pi_{i,0} = \lambda \sum_{h=1}^{min(i,r)} b_h\pi_{i-h,0} + c\mu\pi_{i+1,0}, (c\le i\le m-1) \tag{3}$$

$$\begin{aligned}(\lambda + c\mu)\pi_{i,0} = {} & \lambda \sum_{h=1}^{min(i,r)} b_h\pi_{i-h,0} + c\mu\left(1-\delta_{i,n-1}\right)\pi_{i+1,0} \\ & + \delta_{i,m}(c\mu + l_1\mu_1)\pi_{i+1,1}, (m\le i\le n-1)\end{aligned} \tag{4}$$

$$\begin{aligned}&(\lambda + c\mu + l\mu_1)\pi_{i,1} \\ &\quad = \lambda\left(1-\delta_{i,m+1}\right)\left(I\{n\le i\le n+r-1\}\sum_{h=i-n+1}^{\min(i,r)} b_h\pi_{i-h,0} + \sum_{h=1}^{\min(i-m-1,r)} b_h\pi_{i-h,1}\right) \\ &\qquad + (c\mu + l\mu_1)\pi_{i+1,1}, (m+1\le i\le n+r-1)\end{aligned} \tag{5}$$

$$(\lambda + c\mu + l\mu_1)\pi_{i,1} = \lambda \sum_{h=1}^{min(i-n,r)} b_h\pi_{i-h,1} + (c\mu + l\mu_1)\pi_{i+1,1}, (n+r\le i\le n+2r-1) \tag{6}$$

$$(\lambda + c\mu + l\mu_1)\pi_{i,1} = \lambda \sum_{h=1}^{min(i-n-r,r)} b_h\pi_{i-h,1} + (c\mu + l\mu_1)\pi_{i+1,1}, (i\ge n+2r) \tag{7}$$

where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise, and $I\{a\le i\le b\}$ is an indicator function such that $I\{a\le i\le b\} = 1$ for $a\le i\le b$ and 0 otherwise. As a remark, the balance equations above can be analytically extended to incorporate finite capacity (see Appendix 3), set up and delayed-off times (Sections 3 and 4), and k staffing levels (Section 5).

2.2 Direct Method

Given the balance equations of the baseline model, one could find the joint steady-state distribution via the direct method; we assume that the balance equation (7) terminates at N', where N' is chosen such that $\pi_{N',s}$ is an extremely small probability. Similar to the direct method employed by Chaudhry et al. (2001), we establish the following condition in determining N'; there exists a positive integer N' such that $\left|\pi_{N'-1,s} - \pi_{N',s}\right| < 10^{-50}$. A disadvantage

of the direct method would be in having to solve a large number of equations. While it could be extremely time-consuming to solve a very large number of simultaneous equations, picking a threshold larger than 10^{-50} leads to a lower N, but may result in numerical inaccuracies (possibly ranging from being a few decimal places off to even negative probabilities). In our baseline model, finding the joint steady-state distribution via the direct method involves solving a set of $N_D = N + n - m$ equations. Later in Section 6, we compare N_D against that of our method for each model extension.

2.3 Root Equation

In light of the transition diagram in Fig. 1, we identify the repeating portion of the state transition diagram as $\{\pi_{i,1}, i \geq n + r\}$ which corresponds to the balance equation (7). While balance equation (6) also qualifies as a repeating portion, our approach is to assume a general solution at a higher i (i.e. $i \geq n + r$) and then analytically exploit the balance equations backwards (i.e. $0 \leq i \leq n + r - 1$) to see which segment(s) of the transition diagram (both repeating and non-repeating portions) can be represented by our general solution (this is described in detail in Appendix 2). Doing so also reduces the number of equations, the benefit of which is numerically shown in Section 6. Therefore, in solving the baseline queue, we select the solution of a general form $\pi_{i,1} = Cz^i$, $(i \geq n + r)$ as it represents our chosen repeating portion, as well as satisfying the required properties of difference equations (Appendix 5). Substituting the solution of this general form into the balance equation (7) leads to

$$(\lambda + c\mu + l\mu_1)Cz^i = \lambda \sum_{h=1}^{r} b_h Cz^{i-h} + (c\mu + l\mu_1)Cz^{i+1}, (i \geq n + 2r)$$

or rearranged as

$$1 = \frac{1}{\lambda + c\mu + l\mu_1}\left[\lambda \sum_{h=1}^{r} b_h z^{-h} + (c\mu + l\mu_1)z\right] \tag{8}$$

We define expression (8) as the root equation of the model. Since (8) has r roots inside the unit circle $|z| = 1$, let these roots be $z_1, z_2, \ldots, z_r$ (see Appendix 1 for the proof). The solution of a general form becomes r-fold in that it becomes a geometric sum

$$\pi_{i,1} = \sum_{h=1}^{r} C_h z_h^i, (i \geq n + r) \tag{9}$$

where C_h for $1 \leq h \leq r$ are yet to be determined non-zero constants. As a remark, one may exploit the option of considering the balance equation (3) (instead of (7)) as the r-th order linear difference equation. However, doing so will lead to r roots that exist under the condition $\frac{\lambda E[X]}{c\mu} < 1$. This results in an incomplete solution since the joint steady-state distribution cannot be computed when $\rho < 1 \leq \frac{\lambda E[X]}{c\mu}$. Therefore, the balance equation (3) is deemed inappropriate to be the difference equation in determining the joint steady-state distribution.

2.4 Determining the Joint Steady-State Distribution

Given (9), let N_R be the total number of unknowns which is the sum of the following: The total number of unknown probabilities $\{\pi_{i,0}, 0 \leq i \leq n - 1\}$, $\{\pi_{i,1}, m + 1 \leq i \leq n + r - 1\}$, and the total number of

unknown constant terms C_h, $(1 \le h \le r)$. Therefore the unknown probabilities and constant terms can be found by generating $N_R = 2n - m + 2r - 1$ equations. Intuitively, these N_R equations can be generated from the balance equations (2) through (6) along with the normalizing condition:

$$\sum_{i=0}^{n-1} \pi_{i,0} + \sum_{i=m+1}^{\infty} \pi_{i,1} = 1 \tag{10}$$

The benefit of a difference equations approach is not just in being able to express the lengthy tail probabilities (i.e. $\pi_{i,1}$ for $i \ge n + r$) in terms of a single expression (9). By leveraging the root equation (8) and other properties of difference equations (Appendix 5), we can further reduce N_R in a systematic way (see Appendix 2 for details). Reducing N_R significantly increases the computational efficiency when compared against N_D from the direct method (see Section 6).

3 Extension to the $M^X/M/c + l/(m, n)/setup$ Queue

The baseline model can be extended to feature a set up time; consider a situation where the l dynamic servers are initially turned off and then an upper-threshold has been reached due to a batch arrival. Instead of turning on immediately, all l dynamic servers go through an exponentially distributed set up time in a collective manner. After this setup time has elapsed, the l dynamic servers are turned on and begin to serve. Let A denote a generic setup time with mean $E[A] = {}^{1}/_{\alpha}$. With the definition of ρ remaining unchanged, the stability condition ($\rho < 1$) also holds in this extended model. In Kendall's notation we denote this extension as an $M^X/M/c + l/(m, n)/setup$ queue. It has the joint steady-state distribution $\{\pi_{i,0}, i \ge 0\}$ and $\{\pi_{i,1}, i \ge m + 1\}$. The normalizing condition is defined as

$$\sum_{i=0}^{\infty} \pi_{i,0} + \sum_{i=m+1}^{\infty} \pi_{i,1} = 1 \tag{11}$$

See Fig. 2 below for a simple example of transitions among different states.

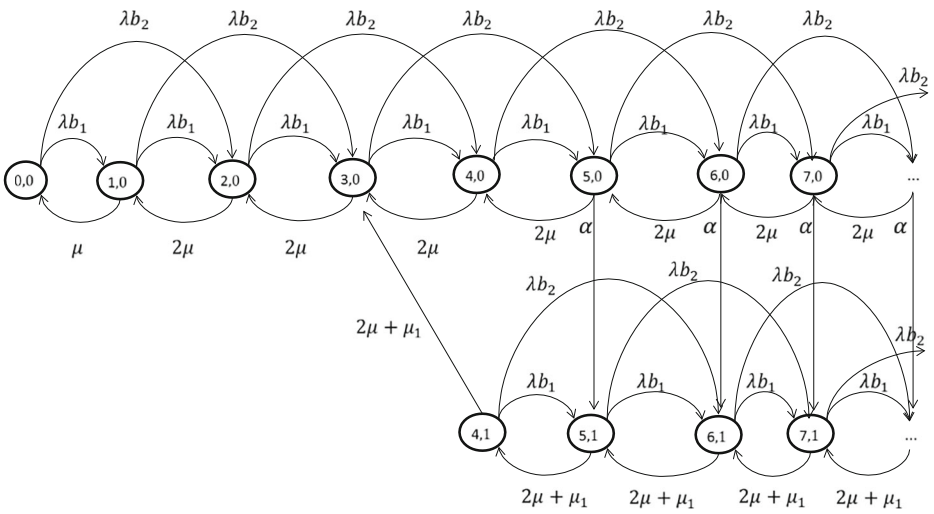


Fig. 2 Transition diagram of the $M^X/M/c + l/(m, n)/setup$ queue when $c = 2$, $l = 1$, $r = 2$, $m = 3$,and $n = 5$

The balance equations that describe the system dynamics of the $M^X/M/c + l/(m, n)/setup$ queue are provided: While the balance equations (1) through (3) from Section 2.1 remain unchanged, the balance equation (4) is modified by replacing $\delta_{n-1,\,n-1}$ with 0. Similarly, the balance equation (5) is modified as follows:

$$(\lambda + c\mu + l\mu_1)\pi_{i,1} = \lambda \sum_{h=1}^{min(i-m-1,r)} b_h \pi_{i-h,1} + (c\mu + l\mu_1)\pi_{i+1,1}, (m+2 \le i \le n-1) \tag{12}$$

In addition, the following two balance equations are added:

$$(\lambda + \alpha + c\mu)\pi_{i,0} = \lambda \sum_{h=1}^{\min(i,r)} b_h \pi_{i-h,0} + c\mu\pi_{i+1,0}, (i \ge n) \tag{13}$$

$$(\lambda + c\mu + l\mu_1)\pi_{i,1} = \lambda \sum_{h=1}^{min(i-m-1,r)} b_h \pi_{i-h,1} + \alpha\pi_{i,0} + (c\mu + l\mu_1)\pi_{i+1,1}, (i \ge n) \tag{14}$$

3.1 Root Equation and Determining the Joint Steady-State Distribution

To determine the joint steady-state distribution $\{\pi_{i,s}, i \ge 0, s = 0, 1\}$ we interpret the balance equation (13) as an r-th order linear difference equation. By doing so, we select the solution of a general form $\pi_{i,0} = Cz^i$, $(i \ge n + r)$ given that the repeating portions of the transition diagram span the region $i \ge n + r$. Substituting the solution of this general form into the balance equation (13) leads to

$$1 = \frac{1}{\lambda + \alpha + c\mu}\left(\lambda \sum_{h=1}^{r} b_h z^{-h} + c\mu z\right) \tag{15}$$

The root equation (15) has r roots inside the unit circle $|z| = 1$ (this can be proved in a similar manner as described in Appendix 1). Let the roots of (15) be $z_1, z_2, \ldots, z_r$. The general solution becomes r-fold of the form

$$\pi_{i,0} = \sum_{h=1}^{r} C_h z_h^i, (i \ge n + r) \tag{16}$$

Depending on the size of r, (16) could also hold for $(n \le i \le n + r - 1)$ which can be proved in a similar manner as described in Appendix 2. In solving for the joint steady-state distribution the direct method would result in having to solve a set of $N_D = 2N' - m + 1$ equations whereas expressing the queue length in terms of the geometric sum results in having to solve a set of $N_R = N' + n - m + r$ equations; a numerical comparison between the values N_D and N_R is made in Section 6. As a remark, the rationale for choosing the balance equation (13) as the difference equation in lieu of the other balance equations that represent the repeating portions is as

follows: The balance equation (3) is unsuitable for the same reasons as indicated in Section 2.3. The balance equation (12) is also unsuitable since it requires the general solution $\pi_{i,1} = Cz^i$, $(m + r + 2 \le i \le n)$ which imposes a more stringent assumption $(m + r + 2 \le n)$ while the original region is $m + 2 \le n$. Lastly, the balance equation (14) is unsuitable since it requires the general solution $\pi_{i,0} + \pi_{i,1} = Cz^i$, $(i \ge n + r)$ that leads to a root equation with $r - 1$ roots inside the unit circle and the r-th root on the unit circle (i.e. $|z_r| = 1$).

4 Extension to the $M^X/M/c + l/(m, n)/delayedoff$ Queue

The baseline model (or the extended model featuring a setup time) can be extended to feature a delayed-off time; consider a situation where the number of jobs in the system has dropped to m. Instead of turning off immediately, the l dynamic servers remain collectively turned on for an exponentially distributed period of time before removal. Let B denote a generic delayed-off time with mean $E[B] = {}^1/_\beta$. We denote this extension as the $M^X/M/c + l/(m, n)/delayedoff$ queue with the joint steady-state distribution $\{\pi_{i,0}, 0 \le i \le n-1\}$ and $\{\pi_{i,1}, i \ge 0\}$. The normalizing condition is given by

$$\sum_{i=0}^{n-1} \pi_{i,0} + \sum_{i=0}^{\infty} \pi_{i,1} = 1 \tag{17}$$

See Fig. 3 below for a simple example of transitions among different states.

The balance equations that describe the system dynamics of the $M^X/M/c + l/(m, n)/delayedoff$ queue are obtained by modifying the earlier ones. From Section 2.1, we replace

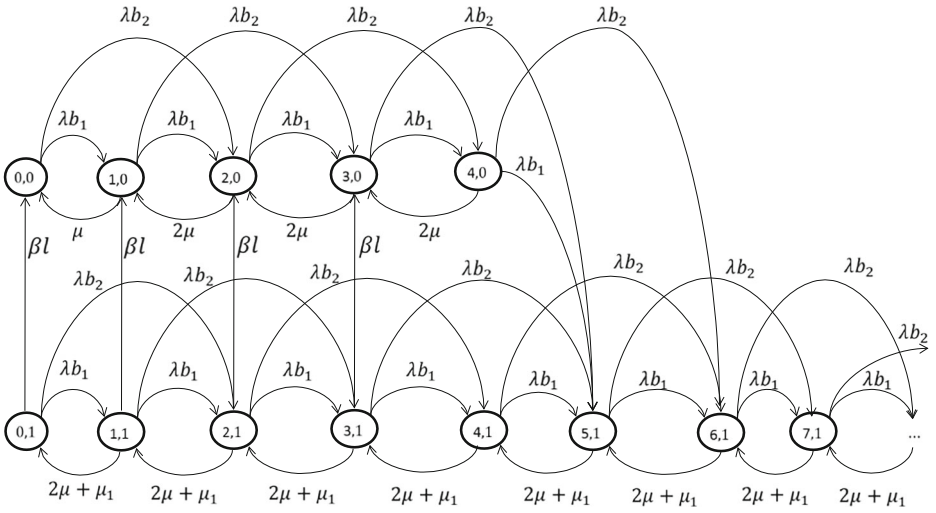


Fig. 3 Transition diagram of the $M^X/M/c + l/(m, n)/delayedoff$ queue when $c = 2$, $l = 1$, $r = 2$, $m = 3$,and $n = 5$

$\delta_{m+1,\,m+1}$ with 0 and replace the expression '$\min(i-m-1, r)$' with '$\min(i, r)$' in the balance equation (5). The balance equations (6) and (7) remain unchanged. In addition the balance equations (1) through (4) are modified as follows:

$$\lambda\pi_{0,0} = \mu\pi_{1,0} + \beta\pi_{0,1} \tag{18}$$

$$(\lambda+\beta)\pi_{0,1} = (c\mu + l\mu_1)\pi_{1,1} \tag{19}$$

$$(\lambda+i\mu)\pi_{i,0} = \lambda\sum_{h=1}^{min(i,r)} b_h\pi_{i-h,0} + (i+1)\mu\pi_{i+1,0} + \beta\pi_{i,1}, (1\le i\le c-1) \tag{20}$$

$$(\lambda+c\mu)\pi_{i,0} = \lambda\sum_{h=1}^{min(i,r)} b_h\pi_{i-h,0} + c\mu\pi_{i+1,0} + \beta\pi_{i,1}, (c\le i\le m) \tag{21}$$

$$(\lambda+c\mu+l\mu_1+\beta)\pi_{i,1} = \lambda\sum_{h=1}^{min(i,r)} b_h\pi_{i-h,1} + (c\mu+l\mu_1)\pi_{i+1,1}, (1\le i\le m) \tag{22}$$

$$(\lambda+c\mu)\pi_{i,0} = \lambda\sum_{h=1}^{min(i,r)} b_h\pi_{i-h,0} + c\mu\left(1-\delta_{i,n-1}\right)\pi_{i+1,0}, (m+1\le i\le n-1) \tag{23}$$

As a remark, the balance equations for the $M^X/M/c+l/(m,n)/setup/delayedoff$ queue can be easily obtained by combining the balance equations (18) through (22) with (13), (14), and the following modified balance equations: Balance equation (23) is modified by replacing $\delta_{n-1,\,n-1}$ with 0. The lower bound of the range for balance equation (12) is modified to $i \ge m+1$ (versus $i \ge m+2$) and in balance equations (12) and (14), each instance of '$\min(i-m-1, r)$' is replaced with '$\min(i, r)$'. The normalizing condition for the $M^X/M/c+l/(m,n)/setup/delayedoff$ queue is $\sum_{i=0}^{\infty}\pi_{i,0} + \sum_{i=0}^{\infty}\pi_{i,1} = 1$. See Fig. 4 below for a simple example of transitions among different states.

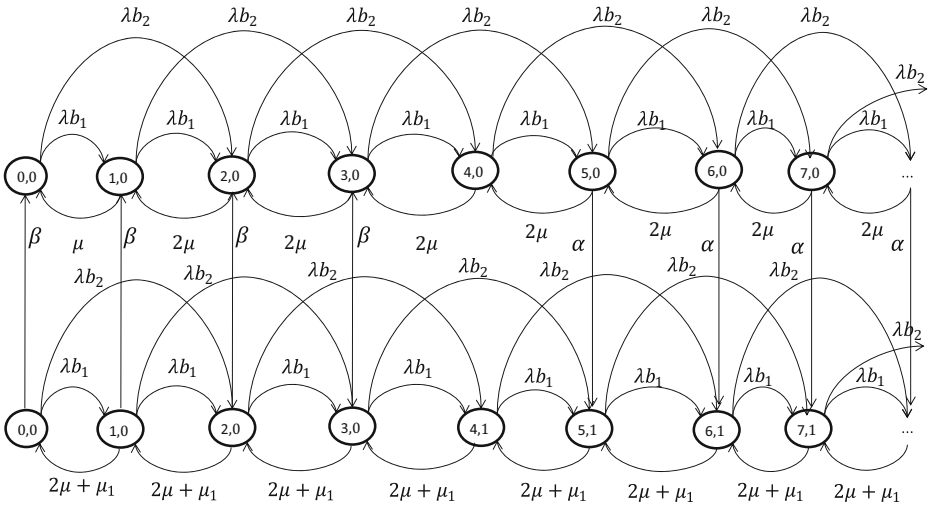


Fig. 4 Transition diagram of the $M^X/M/c+l/(m,n)/setup/delayedoff$ queue when $c=2$, $l=1$, $r=2$, $m=3$,and $n=5$

4.1 Root Equations and Determining the Joint Steady-State Distribution

The general solutions and the root equations of the $M^X/M/c+l/(m,n)/delayedoff$ and the $M^X/M/c+l/(m,n)/setup/delayedoff$ queues are identical to that of the baseline and the $M^X/M/c+l/(m,n)/setup$ queue, respectively. Let N_R be the total number of unknown probabilities and unknown constant coefficients of the solution. In the model $M^X/M/c+l/(m,n)/delayedoff$ there are $N_R=2(n+r)$ (versus $N_D=N'+n+1$) unknowns whereas in the model $M^X/M/c+l/(m,n)/setup/delayedoff$ there are $N_R=N'+n+r+2$ (versus $N_D=2(N'+1)$) unknowns. The same number of equations can be generated from the respective balance equations and the normalizing condition.

5 Extension to the $M^X/M/c+l/(m,n)$ Queue with k Staffing Levels

The baseline model (or all previous extensions) can be extended to feature k ($k\geq 2$) staffing levels such that groups of servers (say $l_1, l_2, \ldots, l_k$ each with service rate $\mu_1, \mu_2, \ldots, \mu_k$) are turned on sequentially in an aggregate manner as the number of jobs in the system grows to surpass a sequence of increasing upper-thresholds (say $n_1, n_2, \ldots, n_k$). These server groups are then turned off in the reverse order (i.e. $l_k, l_{k-1}, \ldots, l_1$) as the number of jobs in the system drops below a decreasing sequence of lower-thresholds (say $m_k, m_{k-1}, \ldots, m_1$). With these additional staffing levels, the following properties must hold: $c+\sum_{s=1}^{k} l_s \leq m_s \leq n_s - 2$ for ($1\leq s\leq k$) and the traffic intensity $\rho=\frac{\lambda}{c\mu+\sum_{s=1}^{k} l_s}\mu_s < 1$. We call this extended system an $M^X/M/c+l/(m,n)$ queue with k staffing levels. The corresponding joint steady-state distribution is given by $\{\pi_{i,s}, i\geq 0, 0\leq s\leq k\}$. The normalizing condition is

$$\sum_{i=0}^{n_1-1}\pi_{i,0}+\sum_{s=1}^{k-1}\sum_{i=m_s+1}^{n_{s+1}-1}\pi_{i,s}+\sum_{i=m_k+1}^{\infty}\pi_{i,k}=1 \tag{24}$$

The balance equations that describe the system dynamics of the $M^X/M/c+l/(m,n)$ queue with k staffing levels can be derived in a similar manner to previous models. The balance Equations (1) and (2) from Section 2.1 remain unchanged and the remainder of the balance equations are provided in Appendix 3.

5.1 Root Equation and Determining the Joint Steady-State Distribution

In solving the $M^X/M/c+l/(m,n)$ queue with k staffing levels via the difference equations approach, we substitute the general solution $\pi_{i,k}=Cz^i$, ($i\geq n_k+r$) into the balance equation (44) such that it leads to the root equation

$$1=\frac{1}{\lambda+c\mu+\sum_{s=1}^{k} l_s\mu_s}\left[\lambda\sum_{h=1}^{r} b_h z^{-h}+\left(c\mu+\sum_{s=1}^{k} l_s\mu_s\right)z\right] \tag{25}$$

equation (25) has r roots inside the unit circle $|z| = 1$ (this can be proved similarly to the result in Appendix 1); let these roots be $z_1, z_2, \ldots, z_r$. The general solution becomes r-fold of the form

$$\pi_{i,k} = \sum_{h=1}^{r} C_h z_h^i, (i \geq n_k + r) \tag{26}$$

where C_h, $(1 \leq h \leq r)$ are non-zero constants. The joint steady-state distribution of the $M^X/M/c + l/(m, n)$ queue with k staffing levels can be found by solving the $N_R = m_1 + \sum_{s=1}^{k-1} (n_s - 2m_s - 1 + m_{s+1}) + 2(n_k - m_k) + r - 1$ equations. As a remark, N_R can be systematically reduced by leveraging (25) and (26) in a similar manner as shown in Appendix 1.

6 Numerical Comparison against the Direct Method

In this section we show numerical comparisons between N_R and N_D for each model. Results were verified by evaluating a state probability independently at N (i.e. direct and difference equations method) and then matching them.

Table 1 $\lambda = 2.0$, $c = 5$, $l = 5$, $\mu = 0.5$, $\mu_1 = 0.5$, $m = c + l$, $n = m + 2$, $b_1 = 0.5$, $b_2 = 0.5$

	Model	Other input parameters	Verification	N_R	N_D
1	Baseline	*N/A*	$\sum_{h=1}^{r} C_h z_h^{307} = -2825.3089(-0.2899)^{307} + 5.6430(0.5899)^{307} = 1.8133\text{x}10^{-49}$ $\pi_{307,0} = 1.8133\text{x}10^{-49}$	15	309
2	Baseline with setup time	α=0.4	$\sum_{h=1}^{r} C_h z_h^{441} = 1566.1528(-0.3391)^{441} + 0.8742(0.7729)^{441} = 4.0643\text{x}10^{-50}$ $\pi_{441,0} = 4.0643\text{x}10^{-50}$	445	873
3	Baseline with delayed-off time	β=0.4	$\sum_{h=1}^{r} C_h z_h^{308} = -2453.4767(-0.2899)^{308} + 4.4119(0.6899)^{308} = 9.7806\text{x}10^{-50}$ $\pi_{308,0} = 9.7806\text{x}10^{-50}$	28	321
4	Baseline with setup and delayed-off times	α=0.4, β=0.4	$\sum_{h=1}^{r} C_h z_h^{432} = -88.4065(-0.3391)^{432} + 0.6674(0.7729)^{432} = 3.1514\text{x}10^{-49}$ $\pi_{432,0} = 3.1514\text{x}10^{-49}$	448	884
5	Baseline with k staffing levels	$k = 2,\ n_1 = 12, m_1 = 10\ n_2 = 16, m_2 = 14$	$\sum_{h=1}^{r} C_h z_h^{529} = -22872.8180(-0.3090)^{529} + 1.7094(0.8090)^{529} = 3.4869\text{x}10^{-49}$ $\pi_{529,0} = 3.4869\text{x}10^{-49}$	20	532

As a remark, in Table 1, a significant reduction from N_D to N_R is achieved in rows 1, 3, and 5, whereas the reduction from N_D to N_R is approximately halved in rows 2 and 4. The reason for such numerical behaviour is as follows. The presence of setup times (i.e. rows 2 and 4) results in partial expression of the queue length in terms of roots. In other words, we can express $\pi_{i,0} = Cz^i$, $(i \geq n + r)$ but the expression $\pi_{i,0} + \pi_{i,1} = Dy^i$, $(i \geq n + r)$ does not hold (see Section 3.1 for further explanation). Therefore, we must treat $\{\pi_{i,1}, i \geq n + r\}$ as unknowns while we are able to express $\{\pi_{i,0}, i \geq n + r\}$ entirely in terms of the roots of equation (15). Because we have nearly halved the number of unknowns, N_R in our approach, when compared against N_D, is approximately halved. On the contrary, in rows 1, 3, and 5 both $\{\pi_{i,1}, i \geq n + r\}$ and $\{\pi_{i,0}, i \geq n + r\}$ are entirely expressed in terms of the roots. This results in a greater reduction in N_R.

7 Performance Measure and Trade-off Analysis

In this section we first introduce a list of performance measures (Table 2) followed by a trade-off analysis between those performance measures (Tables 3, 4, 5 and 6). The performance measures can be largely divided into two categories; system performance and resource consumption. The system performance indicates how well the system is performing whereas the resource consumption indicates the efforts consumed in achieving the corresponding level of system performance. There is a trade-off between the two categories, the extent of which also depends on other factors such as the upper-threshold, mean batch size, and traffic intensity. Using the joint steady-state distribution from earlier sections of this paper we are able to derive all performance measures. As a remark, while some of our performance measures are deducible from a p.g.f., others, such as the switching cost rate, are more conveniently found from the distribution itself.

Using our performance measures in Table 2, we conducted a trade-off analysis between the system performance and resource consumption. Throughout Tables 3, 4, 5 and 6, we have taken the $M^X/M/c + l/(m, n)/K$ (and $M^X/M/c/K$) queues where for each upper-threshold ($n = 8$, 13, 18, 23, 28, 33,and 38) we have represented each corresponding performance measure as a horizontal colour-coded bar (different colours represent different performance measures while the height of each bar corresponds to the magnitude of that performance measure). The height of each bar is based on the ratio to the maximum entry of that metric in the table such that a full bar height corresponds to the maximum entry in that table. We have utilized the parameters $c = 2$, $l = 4$, $\mu = 2.0$, $\mu_1 = 2.5$, $\lambda = 1.5$, $m = c + l$, $K = 70$, and $\varepsilon = {}^1/_\lambda$ throughout Tables 3, 4, 5 and 6. Our findings are summarized in four observations.

Table 2 Performance measures under the two categories

	Performance measure	Formula	Description
System performance	Mean queue length (L_q)	$L_q = \sum_{i=0}^{n-c-1} i\pi_{c+i,0} + \sum_{i=m-c-l+1}^{\infty} i\pi_{c+l+i,1}$	On the right-hand side, the first geometric sum provides the mean queue length of the model when the l dynamic servers are turned off. Similarly, the second geometric sum provides the mean queue length when the l dynamic servers are turned on.
	Average waiting time in queue (W_q)	$W_q = \frac{L_q}{\lambda E[X]}$	The mean waiting-time-in-queue of a randomly positioned job within an incoming batch
	Probability of partial job loss (P_{PJL})	$P_{PJL} = \sum_{i=K-r+1}^{K} \sum_{h=K+1-i}^{r} b_h \pi_{i,1}^{*}$	Probability of a portion of an incoming batch being rejected due to limited capacity. ($\pi_{i,1}^{*}$ corresponds to $\pi_{i,1}$ from Appendix 4.1).
	Probability of total job loss (P_{TJL})	$P_{TJL} = \sum_{i=K-r+1}^{K} \sum_{h=K+1-i}^{r} b_h \pi_{i,1}^{**}$	Probability of an entire incoming batch being rejected due to limited capacity. ($\pi_{i,1}^{**}$ corresponds to $\pi_{i,1}$ from Appendix 4.2).
Resource consumption	Mean number of idle static servers (I_s)	$I_s = \sum_{i=0}^{c-1} (c-i)\pi_{i,0}$	I_s is useful in determining the magnitude of wasted resources and energy, since the static servers remain turned on at all times.
	Switching cost rate (f_l)	$f_l = \varepsilon \sum_{i=0}^{n-1} \sum_{h=n-i}^{r} \lambda b_h \pi_{i,0}$	The switching cost rate that is incurred by turning on the l dynamic servers. Note that ε is the switching cost for turning on the l dynamic servers (per switch). As a remark, in the presence of setup times (see Section 3), the switching cost rate becomes $f_l = \varepsilon \sum_{i=n}^{\infty} \alpha\pi_{i,0}$.
	Dynamic server utilization (u_l)	$u_l = \sum_{i=m+1}^{\infty} \pi_{i,1}$	The probability of dynamic servers' utilization. As a remark, the summation ends at K in lieu of ∞ if the model features a finite capacity (see Appendix 4).

Table 3 $E[X] < c$ with $b_1 = 1.0$

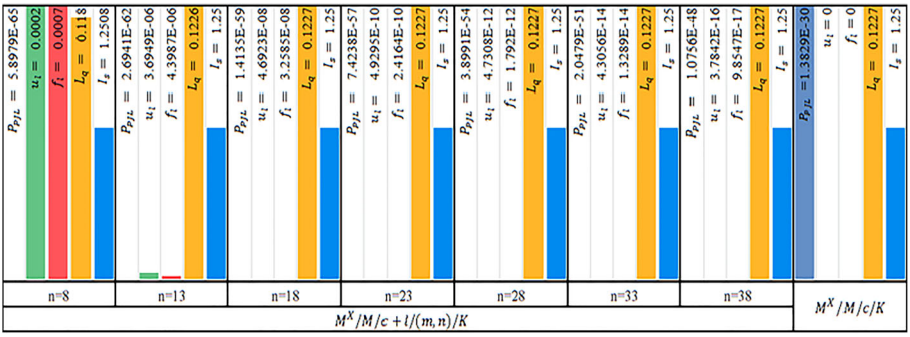


Table 4 $c < E[X] < c + l$ with $b_1 = 0.5$ and $b_3 = 0.5$

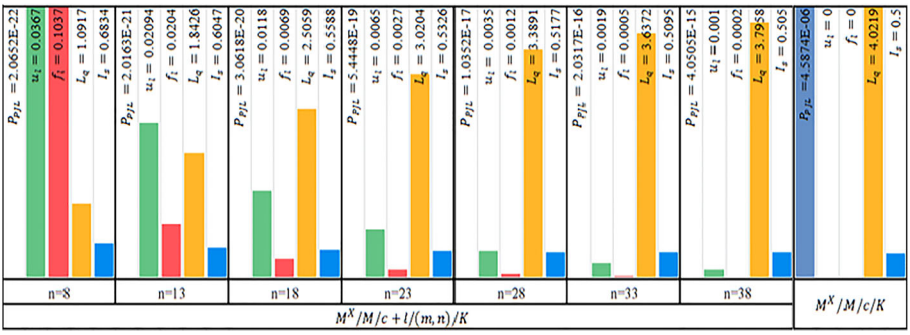


Table 5 $E[X] = c + l$ with $b_1 = 0.5$ and $b_{11} = 0.5$

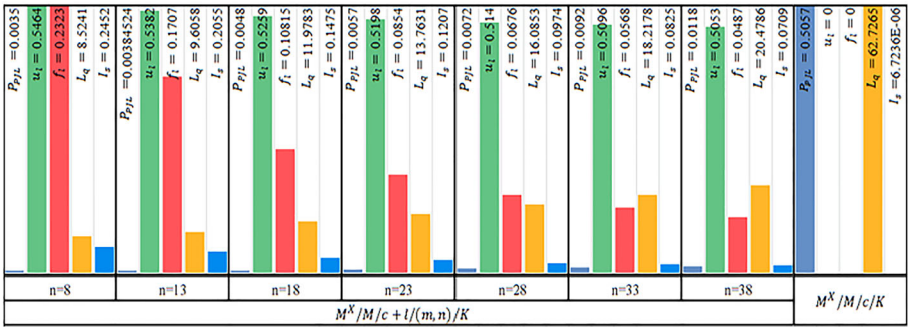
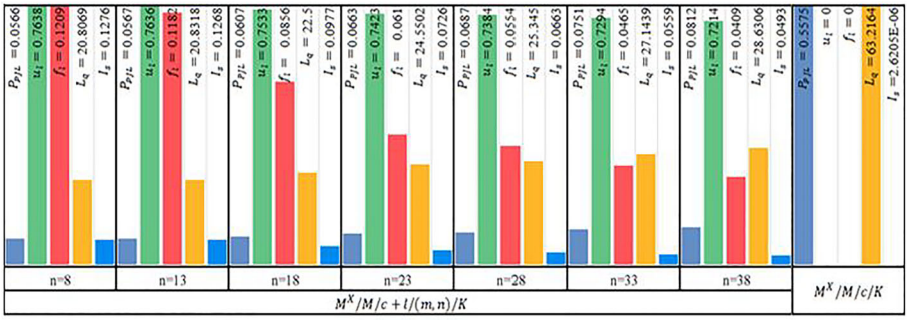


Table 6 $E[X] > c + l$ with $b_1 = 0.5$ and $b_{15} = 0.5$



Observation 1: Across all tables the $M^X/M/c/K$ queue results in the largest P_{PJL} and L_q. As the mean batch size increases, the probability of dynamic server utilization becomes larger. Conversely, as the mean batch size decreases, the probability of dynamic server utilization becomes smaller.

Observation 2: A high switching cost rate coincides with a high chance of the number of customers in the system crossing above and below the upper and lower-thresholds, respectively. It is observed that the switching cost rate is highest when the mean batch size is identical to the system's lower-threshold (i.e. Table 5); a higher chance of crossing above and below the upper and lower-thresholds requires moderately sized batches as well as a moderate rate of batch arrivals. While all tables have identical rates of batch arrivals, each table has different mean batch size. Table 5 appears to have the most moderate mean batch size which contributes to it having the highest switching cost rate.

Observation 3: The impact of dynamic servers on the system is more pronounced when the mean batch size is higher: When the mean batch size is smaller than the system's total capacity (i.e. Tables 3 and 4), adding dynamic servers leads to a relatively small drop in L_q while I_s remains relatively unchanged. When the mean batch size is larger than the system's total capacity (i.e. Tables 5 and 6), adding dynamic servers leads to a sharp decrease in L_q.

Observation 4: P_{PJL} increases with n at a modest rate; it is expected that P_{PJL} will increase at a much faster rate when the mean batch size is larger.

To conclude, we summarize our findings in terms of when the dynamic servers appear to be effective (or ineffective). When the mean batch size is very small (i.e. $E[X] < c$), the dynamic servers appear to be ineffective across all values of n. When the mean batch size is relatively small (i.e. $c < E[X] < c + l$) the dynamic servers contribute effectively in lowering the queue size only when they are turned on at smaller values of n. For higher values of n, the dynamic servers appear to be ineffective. For larger mean batch size (i.e. $E[X] \geq c + l$), in general, the dynamic servers effectively contribute to reducing the queue size even at smaller values of n.

8 Conclusion

In this paper, we have demonstrated that the difference equations approach stands as a reliable tool in treating advanced forms of multi-server bulk arrival queues. Through this work, what would

have otherwise been done via the generating functions approach has been greatly simplified by intuitively choosing a set of balance equations as difference equations. Doing so relies heavily on the finding of Rouché's roots; a critical step in a solution procedure that has been heavily criticized by some researchers due to the perceived risk of numerical inaccuracies, and the laborious and ambiguous steps involved in constructing and inverting a generating function (see Section 1). Such issues are compounded when extending to bulk arrival queues as multiple roots are often involved. Nevertheless, we have successfully demonstrated that our method can handle an advanced form of quasi birth and death process that features bulk arrival, setup and delayed-off times, finite capacity, non-homogenous dynamic servers, and k staffing levels. In the future, our plan is to apply our method in solving non-Markovian and semi-Markovian models that feature working vacations and are formulated in discrete-time.

Appendix 1: Proving the existence of roots

In this appendix we prove that (8) has r roots inside the unit circle $|z| = 1$. We first multiply both sides of the root equation (8) by z^r yielding

$$z^r = \frac{1}{\lambda + c\mu + l\mu_1}\left[\lambda \sum_{h=1}^{r} b_h z^{r-h} + (c\mu + l\mu_1)z^{r+1}\right]$$

Let $f(z) = z^r$ and $g(z) = -\frac{1}{\lambda + c\mu + l\mu_1}\left[\lambda \sum_{h=1}^{r} b_h z^{r-h} + (c\mu + l\mu_1)z^{r+1}\right]$ such that $f(z) + g(z) = 0$. Consider the magnitudes of $f(z)$ and $g(z)$ on the contour $|z| = 1 - \tau$ where τ is positive and sufficiently small. This gives

$$|f(z)| = (1-\tau)^r = 1 - r\tau + o(\tau)$$

and

$$|g(z)| \le \frac{1}{\lambda + c\mu + l\mu_1}\left[\lambda \sum_{h=1}^{r} b_h |z|^{r-h} + (c\mu + l\mu_1)|z|^{r+1}\right]$$

Letting $|z| = 1 - \tau$ in the right-hand side of the above expression leads to the following:

$$\begin{aligned} |g(z)| &\le \frac{1}{\lambda + c\mu + l\mu_1}\left[\lambda \sum_{h=1}^{r} b_h (1-(r-h)\tau) + (c\mu + l\mu_1)(1-(r+1)\tau) + o(\tau)\right] \\ &\le \frac{1}{\lambda + c\mu + l\mu_1}\left[\lambda + c\mu + l\mu_1 - (\lambda + c\mu + l\mu_1)r\tau + \lambda E[X]\tau - (c\mu + l\mu_1)\tau + o(\tau)\right] \end{aligned}$$

Using the definition of ρ from Section 2, the above expression can be rearranged to give

$$|g(z)| \le 1 - r\tau - \frac{(c\mu + l\mu_1)(1-\rho)\tau}{\lambda + c\mu + l\mu_1} + o(\tau)$$

The fact that $\rho < 1$ implies that $|g(z)| < |f(z)|$ on $|z| = 1 - \tau$. Since $f(z)$ and $g(z)$ satisfy the conditions of Rouché's theorem it follows that (8) has r roots inside the unit circle.

Appendix 2: Reduction of N_R

In this Section we demonstrate that (9) is also true for $n \le i \le n+r-1$. The benefit of doing so is in the analytical reduction of N_R by r which subsequently enables even further reduction of N_R (the effect of such a reduction is demonstrated in Table 1). We begin this procedure by letting $i = n+2r-1$ in the balance equation (6) and expressing probabilities using (9) where applicable:

$$(\lambda + c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{n+2r-1} = \lambda \sum_{h=1}^{r-1} b_h \sum_{j=1}^{r} C_j z_j^{n+2r-1-h} + \lambda b_r \pi_{n+r-1,1} + (c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{n+2r}$$

This can be rearranged:

$$(\lambda + c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{n+2r-1}\left[1 - \frac{1}{\lambda + c\mu + l\mu_1}\left\{\lambda \sum_{h=1}^{r} b_h z_j^{-h} - (c\mu + l\mu_1)z_j\right\} + \frac{\lambda b_r z_j^{-r}}{\lambda + c\mu + l\mu_1}\right] = \lambda b_r \pi_{n+r-1,1}$$

Applying (8) to the above expression and given that λ and b_r are both strictly positive, we have

$$\pi_{n+r-1,1} = \sum_{h=1}^{r} C_h z_h^{n+r-1}$$

By letting $i = n+2r-2, n+2r-3, \ldots, n+r+1, n+r$, we have the following result:

$$\pi_{i,1} = \sum_{h=1}^{r} C_h z_h^{i}, (n \le i \le n+r-1) \tag{27}$$

By deriving (27), we have reduced N_R by r, it went from $2n-m+2r-1$ to $2n-m+r-1$.

Appendix 2.1: Further reduction of N_R

Further reduction of N_R is desired as it enables efficient numerical computations. To perform such a reduction we must distinguish and treat each of the following two cases separately: Case 1 occurs when $r \ge n$ and Case 2 occurs when $r < n$.

Appendix 2.1.1: Case 1: $r \ge n$

In this case, an incoming batch of size h, $(1 \le h \le r)$ could be equal to or larger than n so that the l dynamic servers are turned on immediately upon arrival of the batch. Using (27) we concluded that there are $N_R = 2n-m+r-1$ unknowns: $\{\pi_{i,0}, 0 \le i \le n-1\}$, $\{\pi_{i,1}, m+1 \le i \le n-1\}$, and C_h, $(1 \le h \le r)$. To further reduce N_R we let $i = n+r-1$ in balance equation (5) and express $\pi_{i,1}$ with (27) where applicable. Doing so gives

$$(\lambda + c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{n+r-1} = \lambda\left(b_r \pi_{n-1,0} + \sum_{h=1}^{r} b_h \pi_{n+r-1-h,1}\right) + (c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{n+r}$$

The above expression is then rearranged to yield

$$(\lambda + c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{n+r-1}\left[1-\frac{1}{\lambda + c\mu + l\mu_1}\left\{\lambda\sum_{h=1}^{r} b_h z_j^{-h}-(c\mu + l\mu_1)z_j\right\}+\frac{\lambda}{\lambda + c\mu + l\mu_1}b_r z_j^{-r}\right]$$
$$= \lambda b_r\left(\pi_{n-1,0}+\pi_{n-1,1}\right)$$

Applying (8) to the above expression and given that λ and b_r are both strictly positive, we have

$$\sum_{j=1}^{r} C_j z_j^{n-1} = \pi_{n-1,0}+\pi_{n-1,1}$$

We let $i = n+r-2$, $n+r-3$, …, $m+r+2$, $m+r+1$ in balance equation (5) and prove that

$$\sum_{j=1}^{r} C_j z_j^{i} = \pi_{i,0}+\pi_{i,1}, (m+1\le i\le n-1) \tag{28}$$

We proceed further for the remaining values of i (i.e. $i = m+r$, $m+r-1$, …, $r+1$, r). Let $i = m+r$ in balance equation (5) and express $\pi_{i,0}+\pi_{i,1}$ with (28) where applicable. Doing so gives

$$(\lambda + c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{m+r} = \lambda\left(\sum_{h=m+r-n+1}^{r} b_h\pi_{m+r-h,0}+\sum_{h=1}^{r-1} b_h\pi_{m+r-h,1}\right)+(c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{m+r+1}$$

which can be rearranged to

$$(\lambda + c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{m+r}$$
$$= \lambda\left\{\sum_{h=1}^{m+r-n} b_h\pi_{m+r-h,1}+\sum_{h=m+r-n+1}^{r-1} b_h\left(\pi_{m+r-h,0}+\pi_{m+r-h,1}\right)+b_r\pi_{m,0}\right\}+(c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{m+r+1}$$

or

$$(\lambda + c\mu + l\mu_1)\sum_{j=1}^{r} C_j z_j^{m+r}\left[1-\frac{1}{\lambda + c\mu + l\mu_1}\left\{\lambda\sum_{h=1}^{r} b_h z_j^{-h}+(c\mu + l\mu_1)z_j\right\}+\frac{b_r z_j^{-r}}{\lambda + c\mu + l\mu_1}\right] = \lambda b_r\pi_{m,0}$$

Applying (8) to the above expression and given that λ and b_r are both strictly positive, we have

$$\pi_{m,0} = \sum_{j=1}^{r} C_j z_j^{m}$$

By letting $i = m+r-1$, $m+r-2$, …, $r+1$, r in balance equation (5), it can be shown that

$$\sum_{j=1}^{r} C_j z_j^{i} = \pi_{i,0}, (0\le i\le m) \tag{29}$$

Therefore when $r \ge n$, by deriving expression (28) and (29) we have further reduced N_R by n so that it is reduced from $2n-m+r-1$ to $n-m+r-1$. The needed N_R equations can be generated from the balance equations such that $\{\pi_{i,s}, i \ge 0, s = 0, 1\}$ can be explicitly expressed as

$$\pi_{i,s} = \begin{cases} \sum_{l=1}^{r} C_l z_l^i, (0 \le i \le m, s = 0) \\ already\ determined, (m+1 \le i \le n-1, s = 0) \\ \sum_{l=1}^{r} C_l z_l^i - \pi_{i,0}, (m+1 \le i \le n-1, s = 1) \\ \sum_{l=1}^{r} C_l z_l^i, (i \ge n, s = 1) \end{cases} \tag{30}$$

where the '*already determined*' probabilities are those that are simultaneously found along with the C_h's when solving the N_R equations.

Appendix 2.1.2: Case 2: $r < n$

In this case, we assume that an incoming batch of size h, $(1 \le h \le r)$ will prompt the l dynamic servers to turn on immediately upon arrival of the batch. With (27) found we have concluded that there are $N_R = 2n - m + r - 1$ unknowns: $\{\pi_{i,0}, 0 \le i \le n-1\}$, $\{\pi_{i,1}, m+1 \le i \le n-1\}$, and C_h, $(1 \le h \le r)$. In reducing N_R for Case 2 we must further consider two subcases: $n - r \le m$ and $n - r > m$. As a remark, readers will later see that both of these subcases lead to the reduction of N_R by r. However, such a separation needs to be made as the expressions for $\pi_{i,s}$ for each subcase are different.

Appendix 2.1.2.1: Subcase 1: $n - r \le m$

The procedure to compute $\{\pi_{i,0}, 0 \le i \le n-1\}$, $\{\pi_{i,1}, m+1 \le i \le n-1\}$, and C_h, $(1 \le h \le r)$ when $n - r \le m$ follows the same procedure as provided in Appendix 2.1.1 up to the derivation of (28). However, after (28), instead of letting $i = m + r$, $m + r - 1$, …, $r + 1$, r in the balance equation (5), we let $i = m + r$, $m + r - 1$, …, $n + 1$, n as $r < n$. Doing so leads to

$$\sum_{j=1}^{r} C_j z_j^i = \pi_{i,0}, (n - r \le i \le m) \tag{31}$$

Therefore when $n - r \le m$, by deriving expression (31) we have further reduced N_R by r, from $2n - m + r - 1$ to $2n - m - 1$. The needed N_R equations can be generated from the balance equations such that $\{\pi_{i,s}, i \ge 0, s = 0, 1\}$ can be explicitly expressed as

$$\pi_{i,s} = \begin{cases} already\ determined, (0 \le i \le n - r - 1, s = 0) \\ \sum_{l=1}^{r} C_l z_l^i, (n - r \le i \le m, s = 0) \\ already\ determined, (m+1 \le i \le n-1, s = 0) \\ \sum_{l=1}^{r} C_l z_l^i - \pi_{i,0}, (m+1 \le i \le n-1, s = 1) \\ \sum_{l=1}^{r} C_l z_l^i, (i \ge n, s = 1) \end{cases} \tag{32}$$

where the '*already determined*' probabilities are those that are simultaneously found along with the C_h's when solving the N_R equations.

Appendix 2.1.2.2: Subcase 2: $n-r>m$

The procedure to compute $\{\pi_{i,0}, 0\le i\le n-1\}$, $\{\pi_{i,1}, m+1\le i\le n-1\}$, and C_h, $(1\le h\le r)$ when $n-r>m$ is slightly different than the procedure provided in Appendix 2.1.2.1. Instead of letting $i=n+r-1, n+r-2, \ldots, m+r+2, m+r+1$ in the balance equation (5) in Appendix 2.1.2.1, we let $i=n+r-1, n+r-2, \ldots, n+1, n$ as $n-r>m$. Doing so leads to

$$\sum_{j=1}^{r} C_j z_j^i = \pi_{i,0} + \pi_{i,1}, (n-r\le i\le n-1) \tag{33}$$

Therefore when $n-r>m$, by deriving expression (33) we have further reduced N_R by r (as done in Appendix 2.1.2.1). The needed N_R equations can be generated from the balance equations such that $\{\pi_{i,s}, i\ge 0, s=0,1\}$ can be explicitly expressed as

$$\pi_{i,s} = \begin{cases} already\ determined, (0\le i\le n-r-1, s=0,1) \\ already\ determined, (n-r\le i\le n-1, s=0) \\ \sum_{l=1}^{r} C_l z_l^i - \pi_{i,0}, (n-r\le i\le n-1, s=1) \\ \sum_{l=1}^{r} C_l z_l^i, (i\ge n, s=1) \end{cases} \tag{34}$$

where the '*already determined*' probabilities are those that are simultaneously found along with the C_h's when solving the N_R equations.

Appendix 3: Balance equations for the extension to k staffing levels

The transition dynamics of the $M^X/M/c+l/(m,n)$ queue with k staffing levels are provided. While the balance equations (1) and (2) from the baseline model remain unchanged, the rest of the balance equations are modified to the following:

$$(\lambda + c\mu)\pi_{i,0} = \lambda \sum_{h=1}^{min(i,r)} b_h \pi_{i-h,0} + c\mu\pi_{i+1,0}, (c\le i\le m_1-1) \tag{35}$$

$$\begin{aligned} \left(\lambda + c\mu + \sum_{j=1}^{s-1} l_j\mu_j\right)\pi_{m_s,s-1} &= \lambda \sum_{h=m_s-n_1+1}^{\min(r,m_s)} b_h \pi_{m_s-h,0} + \lambda \sum_{j=1}^{s-2} \sum_{h=m_s-n_{j+1}+1}^{\min\left(r,m_s-m_j-1\right)} b_h \pi_{m_s-h,j} \\ &+ \lambda \sum_{h=1}^{\min(r,m_s-m_{s-1}-1)} b_h \pi_{m_s-h,s-1} + \left(c\mu + \sum_{j=1}^{s} l_j\mu_j\right)\pi_{m_s+1,s} \\ &+ \left(c\mu + \sum_{j=1}^{s-1} l_j\mu_j\right)\pi_{m_s+1,s-1}, (1\le s\le k) \end{aligned} \tag{36}$$

$$\left(\lambda + c\mu + \sum_{j=1}^{s-1} l_j\mu_j\right)\pi_{i,s-1} = \lambda \sum_{h=i-n_1+1}^{\min(r,i)} b_h\pi_{i-h,0} + \lambda \sum_{j=1}^{s-2} \sum_{h=i-n_{j+1}+1}^{\min\left(r,i-m_j-1\right)} b_h\pi_{i-h,j} + \lambda \sum_{h=1}^{\min(r,i-m_{s-1}-1)} b_h\pi_{i-h,s-1} + \left(c\mu + \sum_{j=1}^{s-1} l_j\mu_j\right)\pi_{i+1,s-1}, (m_s + 1 \le i \le n_s - 2, 1 \le s \le k) \quad (37)$$

$$\left(\lambda + c\mu + \sum_{j=1}^{s-1} l_j\mu_j\right)\pi_{n_s-1,s-1} = \lambda \sum_{h=n_s-n_1}^{\min(r,n_s-1)} b_h\pi_{n_s-1-h,0} + \lambda \sum_{j=1}^{s-2} \sum_{h=n_s-n_{j+1}}^{\min\left(r,n_s-m_j-2\right)} b_h\pi_{n_s-1-h,j} + \lambda \sum_{h=1}^{\min(r,n_s-m_{s-1}-2)} b_h\pi_{n_s-1-h,s-1}, (1 \le s \le k) \quad (38)$$

$$\left(\lambda + c\mu + \sum_{j=1}^{s} l_j\mu_j\right)\pi_{m_s+1,s} = \left(c\mu + \sum_{j=1}^{s} l_j\mu_j\right)\pi_{m_s+2,s}, (1 \le s \le k) \quad (39)$$

$$\left(\lambda + c\mu + \sum_{j=1}^{s} l_j\mu_j\right)\pi_{i,s} = \lambda \sum_{h=1}^{\min(r,i-m_s-1)} b_h\pi_{i-h,s} + \left(c\mu + \sum_{j=1}^{s} l_j\mu_j\right)\pi_{i+1,s}, (m_s + 2 \le i \le n_s - 1, 1 \le s \le k) \quad (40)$$

$$\left(\lambda + c\mu + \sum_{j=1}^{s} l_j\mu_j\right)\pi_{i,s} = \lambda \sum_{h=i-n_1+1}^{\min(r,i)} b_h\pi_{i-h,0} + \lambda \sum_{j=1}^{s-1} \sum_{h=i-n_{j+1}+1}^{\min\left(r,i-m_j-1\right)} b_h\pi_{i-h,j} + \lambda \sum_{h=1}^{\min(r,i-m_s-1)} b_h\pi_{i-h,s} + \left(c\mu + \sum_{j=1}^{s} l_j\mu_j\right)\pi_{i+1,s}, (n_s \le i \le n_s + r - 1, 1 \le s \le k) \quad (41)$$

$$\left(\lambda + c\mu + \sum_{j=1}^{s} l_j\mu_j\right)\pi_{i,s} = \lambda \sum_{h=1}^{\min(r,i-n_s)} b_h\pi_{i-h,s} + \left(c\mu + \sum_{j=1}^{s} l_j\mu_j\right)\pi_{i+1,s}, (n_s + r \le i \le m_{s+1} - 1, 1 \le s \le k-1) \quad (42)$$

$$\left(\lambda + c\mu + \sum_{j=1}^{k} l_j\mu_j\right)\pi_{i,k} = \lambda \sum_{h=1}^{\min(r,i-n_k)} b_h\pi_{i-h,k} + \left(c\mu + \sum_{j=1}^{k} l_j\mu_j\right)\pi_{i+1,k}, (n_k + r \le i \le n_k + 2r-1) \tag{43}$$

$$\left(\lambda + c\mu + \sum_{j=1}^{k} l_j\mu_j\right)\pi_{i,k} = \lambda \sum_{h=1}^{\min(r,i-n_k-r)} b_h\pi_{i-h,k} + \left(c\mu + \sum_{j=1}^{k} l_j\mu_j\right)\pi_{i+1,k}, (i \ge n_k + 2r) \tag{44}$$

Appendix 4: Extension to the $M^X/M/c + l/(m, n)/K$ queue

The baseline model can be extended to feature a finite capacity such that the total number of jobs held by the system is finite. Therefore, the $M^X/M/c + l/(m, n)$ queue with finite capacity can house up to K, $(1 \le K < +\infty)$ jobs in the system where K includes the jobs in queue as well as those being served by both the static and dynamic servers (if any). Therefore we have the $M^X/M/c + l/(m, n)/K$ queue with the joint steady-state distribution $\{\pi_{i,\,s}, 0 \le i \le K, s = 0, 1\}$ and the normalizing condition

$$\sum_{i=0}^{n-1} \pi_{i,0} + \sum_{i=m+1}^{K} \pi_{i,1} = 1 \tag{45}$$

With the introduction of finite capacity, an incoming batch can be rejected if its size (h) exceeds the available space ($K - i$). When $h > K - i$ the model $M^X/M/c + l/(m, n)/K$ is subject to one of the following two rejection policies: partial rejection of a batch occurs when out of h jobs the $K - i$ jobs are admitted into the system and the remaining $(h - K + i)$ jobs are rejected. Total rejection of a batch occurs when, given the same condition, the entire batch is rejected. The balance equations that describe the system dynamics of the $M^X/M/c + l/(m, n)/K$ queue can be derived by modifying the balance equation (7) from Section 2.1 to incorporate each rejection policy. These are provided in the following two sections.

Appendix 4.1: $M^X/M/c + l/(m, n)/K$ queue with partial rejection

$$(\lambda + c\mu + l\mu_1)\pi_{i,1} = \lambda \sum_{h=1}^{min(i-n-r,r)} b_h\pi_{i-h,1} + (c\mu + l\mu_1)\pi_{i+1,1}, (n + 2r \le i \le K-1) \tag{46}$$

$$(c\mu + l\mu_1)\pi_{K,1} = \lambda \sum_{j=1}^{r} \sum_{h=j}^{r} b_h\pi_{K-j,1} \tag{47}$$

Appendix 4.2: $M^X/M/c + l/(m, n)/K$ queue with total rejection

$$(\lambda + c\mu + l\mu_1)\pi_{i,1} = \lambda \sum_{h=1}^{min(i-n-r,r)} b_h\pi_{i-h,1} + (c\mu + l\mu_1)\pi_{i+1,1}, (n + 2r \le i \le K-r-1) \tag{48}$$

$$\left(\lambda \sum_{h=1}^{K-i} b_h + c\mu + l\mu_1\right)\pi_{i,1} = \lambda \sum_{h=1}^{min(i-n-r,r)} b_h \pi_{i-h,1} + (c\mu + l\mu_1)\pi_{i+1,1}, (K-r \le i \le K-1) \quad (49)$$

$$(c\mu + l\mu_1)\pi_{K,1} = \lambda \sum_{h=1}^{r} b_h \pi_{K-h,1} \quad (50)$$

The above balance equations can be solved via the difference equations approach as demonstrated in earlier sections of this paper.

Appendix 5: Properties of difference equations

The difference equations approach we introduced in solving the baseline model and its extensions is based on interpreting the model's balance equations as difference equations. By doing so, we can express the solution in terms of roots by leveraging the well-established properties of linear difference equations. As discussed in Chaudhry and Templeton (1983), an equation of the type

$$a_0 f_{x+n} + a_1 f_{x+n-1} + \ldots + a_{n-1} f_{x+1} + a_n f_x = b_x, (x = 1, 2, \ldots)$$

where the a_i are known constants, f_i are unknown functions to be determined, and b_x is a given function of x, is called a nonhomogeneous linear difference equation of order n. If $b_x = 0$, for all x, then it is called a homogenous linear difference equation with constant coefficients. A general solution to the above nonhomogeneous equation consists of two parts:

1. A linear combination of all solutions to the homogeneous equation; and
2. A particular solution to the nonhomogeneous equation.

The solution to the homogeneous part of the equation proceeds along the following lines. Letting $f_x = Cz^x$ in the homogeneous equation leads to

$$a_0 C z^{x+n} + a_1 C z^{x+n-1} + \ldots + a_{n-1} C z^{x+1} + a_n C z^x = 0$$

and

$$a_0 z^n + a_1 z^{n-1} + \ldots + a_{n-1} z + a_n = 0$$

The last equation in z, being an n-th degree equation, gives n roots (real or complex, distinct or coincident). As a consequence, assuming that the roots are distinct, the general solution of the homogeneous part is written as

$$f_x = \sum_{j=1}^{n} C_j z_j^x$$

Acknowledgements We thank the two anonymous reviewers whose constructive feedback and suggestions have helped improve and clarify this manuscript. The second and third authors were supported by the Discovery Grant program of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Abate J, Whitt W (1992) The Fourier-series method for inverting transforms of probability distributions. Queueing Syst 10:5–88

Bar-Lev SK, Parlar M, Perry D, Stadije W, van der Duyn Schouten FA (2007) Applications of bulk queues to group testing models with incomplete identification. Eur J Oper Res 183(1):226–237

Berman O, Larson RC (2004) A queueing control model for retail services having back room operations and cross-trained workers. Comput Oper Res 31:201–222

Chaudhry ML (1991) QROOT Software Package. A&A Publications, 395 Carrie Crescent, Kingston

Chaudhry ML, Kim JJ (2016) Analytically elegant and computationaly efficient results in terms of roots for the $GI^X/M/c$ queueing system. Queueing Syst 82(1–2):237–257

Chaudhry ML, Templeton JGC (1983) A first course in bulk queues. Wiley, New York

Chaudhry ML, Harris CM, Marchal WG (1990) Robustness of root finding in single-server queueing models. INFORMS J Comput 2:273–286

Chaudhry ML, Gupta UC, Goswami V (2001) Modeling and analysis of discrete-time multiserver queues with batch arrivals: $GI^X/Geom/m$. INFORMS J Comput 13(3):172–180

Daigle JN, Lucantoni DM (1991) Queuing systems having phase-dependent arrival and service rates. In: Stewart WJ (ed) Numerical Solutions of Markov Chains. Marcel Dekker, Inc, New York

Gandhi A, Doroudi S, Harchol-Balter M, Scheller-Wolf A (2014) Exact analysis of the $M/M/k/setup$ class of Markov chains via recursive renewal reward. Queueing Syst 77(2):177–209

Gouweleeuw FN (1996) A general approach to computing loss probabilities in finite-buffer queues, Ph.D. thesis, Vrije Universiteit, Amsterdam

Harris CM, Brill PH, Fischer MJ (2000) Internet-type queues with power-tailed Interarrival times and computational methods for their analysis. INFORMS J Comput 12(4):261–271

Horvath T, Skadron K (2008) Multi-mode energy management for multi-tier server clusters. In: Proceedings of the 17th International conference on parallel architectures and compilation techniques, PACT, 270–279

Janssen AJEM, van Leeuwaarden JSH (2005) Analytic computation schemes for the discrete-time bulk service queue. Queueing Syst 50:141–163

Kendall DG (1964) Some recent work and further problems in the theory of queues. Theor Prob Appl 9:1–13

Kleinrock L (1975) Queueing Systems, Vol. I: Theory. Wiley, New York

Krioukov A, Mohan P, Alspaugh S, Keys L, Culler D, Katz R (2010) NapSAC: design and implementation of a power-proportional web cluster. In: Proceedings of the First ACM SIGCOMM Workshop on Green Networks. Green Networking 10:15–22

Maccio VJ, Down DG (2015) On optimal policies for energy-aware servers. Perform Eval 90:36–52

Neuts M (1981) Matrix-geometric solutions to stochastic models-an algorithmic approach. The Johns Hopkins University Press, Baltimore

Phung-Duc T (2015) Multiserver queues with finite capacity and setup time. In: Gribaudo M, Manini D, Remke A (eds) Analytical and stochastic Modelling techniques and applications. ASMTA 2015. Lecture notes in computer science, vol 9081. Springer, Cham

Qin W, Wang Q (2007) Modeling and control design for performance management of web servers via an IPV approach. IEEE Trans Control Syst Technol 15(2):259–275

Stidham S Jr (2001) Applied probability in operations research: a retrospective. University of North Carolina, Department of Operations Research, Chapel Hill

Terekhov D, Beck JC (2009) An extended queueing control model for facilities with front room and back room operations and mixed-skilled workers. Eur J Oper Res 198(1):223–231

Zhang ZG (2009) Performance analysis of a queue with congestion-based staffing policy. Manag Sci 55(2):240–251

Zhao YQ (1994) Analysis of the $GI^X/M/c$ model. Queueing Syst 15:347–364