An automated formulation of queues with multiple geometric batch processes

David J Thornley, Harf Zatschler, Peter G Harrison

Department of Computing, Imperial College London
{djt,hz3,pgh}doc.ic.ac.uk

Abstract. We present an automated formulation mechanism that facilitates the inclusion of batches of geometrically distributed size in Markov modulated, multiprocessor queues of finite or infinite capacity. This provides a practical approach for the analytical modelling of many present day communication and computer systems, *e.g.* the Internet and mobile networks. Geometric distributions can be scaled and superimposed to produce a range of convex probability mass functions. The geometric distribution produces an infinite range of batch sizes, which creates unbounded transitions in queue length, leading to Kolmogorov balance equations with an unbounded, possibly infinite, number of terms. Our method centres on producing equivalent transformed Kolmogorov balance equations of minimal, finite range. The key contribution is the mechanical derivation of these transformed equations. Previously such equations had to be derived from first principles for every variant of every applicable queueing system in a highly error-prone procedure.

1 Introduction

Batched processes extend the domain of application of queueing models, for example modelling the bursty traffic observed in present day communication systems. Markov modulation can represent alternate modes of operation of a queueing system, for example the operational or broken status of a server, different types of traffic and certain types of correlation. The simple Markov modulated MM M/M/c queue has been well understood for many years, usually being solved by considering a two dimensional Markov chain (modulation phase × queue length).

We derive an automated formulation mechanism that facilitates the inclusion of batches of geometrically distributed size in Markov modulated, multiprocessor queues of finite or infinite capacity with an arbitrary number of arrival and processing completion streams, to yield a complete set of transformed balance equations soluble using spectral expansion or matrix geometric methods.

The geometric distribution produces an infinite range of batch sizes, which creates unbounded transitions in the queue length, leading to Kolmogorov equations with an unbounded, possibly infinite, number of terms. Previous analyses *e.g.* [5] derive equivalent transformed Kolmogorov equations with a finite number of terms, solved by the spectral expansion method [1]. In these, the transformed equations are derived by hand from first principles, for every model and model-variant, in a highly error-prone procedure. The key contribution in this report is the *mechanical* derivation of the transformed Kolmogorov equations. We present a formulation and solution methodology that allows the piecewise, automated construction of equilibrium solutions (when they exist) for the state occupation probabilities of these Markovian queues. This methodology allows the addition of arbitrarily many batched arrival streams of either positive or negative customers and various queueing disciplines in a Markov modulated environment.

In the next section we define our notation and terminology, specifying the class of queueing systems in section 2. The notation is introduced constructively, to provide a complete description of a very large class of queue models. The crux of this paper is the presentation of the elimination function described in section 3. This provides the basis for the equation transformation mechanism which enables us to render any geometrically batched queueing behaviour into a form solvable by matrix geometric techniques, such as highly efficient direct solvers [4] or spectral expansion [1]. We use Mathematica® [6] as a proofing tool for our methods, as it allows us to perform operations both symbolically and numerically. We include an example symbolic result for localized balance equations to demonstrate the relative simplicity of function in comparison with earlier work [5].

2 Queue formulation

We begin with the Kolmogorov balance equations for the simplest type of queue – $M/M/1/L$. For each new concept included in our queueing paradigm we successively augment this representation until the full balance equations for all the modelled concepts are synthesized.

In all the following, the balance equations are to be written in the form $r_j = 0$ for $0 \leq j \leq L$. The left-hand-side (LHS) of the j^{th} Kolmogorov balance equation r_j for the $M/M/1/L$ queue is given below:

$$r_j^{\text{M/M/1/L}} = \pi_{j-1}\lambda f_{j>0} - \pi_j\left[\lambda f_{j<L} + \mu f_{j>0}\right] + \pi_{j+1}\mu f_{j<L} \tag{1}$$

This expression gives the net incoming probability flux to level j, where π_j is the equilibrium probability of occupying state j, f is the indicator function defined by $f_{\text{true}} = 1$ and $f_{\text{false}} = 0$. The terms λ and μ are the arrival and service rates respectively.

2.1 Modulation

We allow the current state of an independent underlying continuous time, discrete state Markov process to select the parameters for the interval distributions of the arrivals and departures. This N-state process is fully defined by its instantaneous transition rate (generator) matrix Q:

$$Q = \begin{pmatrix} -\sum_{i\neq 1} q_{1i} & q_{12} & \cdots & q_{1N} \\ q_{21} & -\sum_{i\neq 2} q_{2i} & & q_{2N} \\ \vdots & & \ddots & \vdots \\ q_{N1} & \ldots\ldots\ldots\ldots\ldots & & -\sum_{i\neq N} q_{Ni} \end{pmatrix}$$

The joint queue length/modulation states of the resulting queue forms a 2D lattice strip. The dimension describing the state of the modulator is finite, and the other dimension – the queue-length – may be finite or infinite. The equilibrium probability of being in any particular column of the lattice is given by the steady state phase probability of the modulation process π defined by the two equations $\pi.Q = \mathbf{0}$, and $\|\pi\|_1 = 1.$[1]. Here $\mathbf{0}$ is a row vector of N zeros.

To allow for modulation in our model, we treat queue levels as single entities by representing the state occupation probability (SOP) at level j as the $N-$component row vector $\mathbf{v}_j$. This vector consists of the SOPs of the N modulation states at that level. In addition, the representation of arrivals and services is augmentedto give an analogous vector equation.

Consider a two state modulated system, in which an arrival stream of positive customers $\varLambda$ and a service stream M would be represented as follows:

$$\varLambda = \left(\begin{smallmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{smallmatrix}\right), M = \left(\begin{smallmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{smallmatrix}\right)$$

This notation allows us to associate SOPs on the lattice strip with rate parameter expressions by use of row vectors of SOPs pre-multiplying standard-looking rate terms, with the substitution of the identity matrix I for 1, and the appropriate diagonal matrix for a rate term.

$$\mathbf{r}_j^{\text{MM/MM/1/L}} = \mathbf{v}_{j-1}\varLambda f_{j>0} + \mathbf{v}_j\left[Q - \varLambda f_{j<L} - M f_{j>0}\right] + \mathbf{v}_{j+1} M f_{j<L}$$

Compare this expression with that for the unmodulated queue shown in expression 1 to see the substitution of vectors and matrices as appropriate for SOPs and rates respectively. The inclusion of Q provides the modulation transitions.

[1] We express the sum of elements in a vector $\mathbf{x}$ as its L_1 norm $\|\mathbf{x}\|_1 = \sum_i x_i$

2.2 Geometric batches and the GE distribution

The compound Poisson process (CPP) describes point arrivals of customers at exponentially distributed intervals in geometrically sized batches; this can model burstiness (*e.g.* [9]). The generalized exponential (GE) distribution of processing times describes a similar batched behaviour for service completions. The probability distribution function F_{GE} of the interarrival time random variable A of a CPP is a "generalized exponential" distribution [7] with rate parameter λ, and batch size parameter θ: $F_{GE}(a) = P(A \leq a) = 1 - (1-\theta)e^{-\lambda a}$.

The $(1-\theta)$ term gives an impulse at the origin ($F_{GE}(0) = \theta$) giving a non-zero probability to a null interarrival time. This allows a sequence of one or more zero interarrival times, and hence non-unit (geometric) batches. A deterministic unit batch size is given by setting the geometric distribution parameter to zero.

The queue can accomodate geometrically batched occurrences of both arrivals and processing completions. Formally, we have transitions within the queue whose vertical component (increasing or decreasing queue length) has size s with probability $(1-\theta)\theta^{s-1}$, and the horizontal component is zero. As with the rates $\mathit{\Lambda}$ and M, we write the batch parameters in matrix form, where $\mathit{\Theta}$ is the diagonal matrix of geometric arrival batch size parameters (θ) and $\mathit{\Phi}$ similarly for service batch sizes.

With geometric arrival and service batches included, our LHS balance equations become:

$$\begin{aligned}\mathbf{r}_j^{\text{MM CPP/GE/1/L}} =& \sum_{i=0}^{j-1} \mathbf{v}_i \mathit{\Lambda} (I - \mathit{\Theta})^{f_{j<L}} \mathit{\Theta}^{j-i-1} \\ &+ \mathbf{v}_j \left[Q - \mathit{\Lambda} f_{j<L} - M f_{j>0} \right] \\ &+ \sum_{i=j+1}^{L} \mathbf{v}_i M (I - \mathit{\Phi})^{f_{j>0}} \mathit{\Phi}^{i-j-1}\end{aligned}$$

2.3 Multiple Processors

So far our model has assumed the presence of exactly one processor ($c = 1$), with its processing rates given by the matrix M and batches described by the matrix $\mathit{\Phi}$. To introduce multiple homogeneous processors, the processing rate matrix M is replaced by $C_j = \min(j, c)M$ at queue length j, with the batch matrix $\mathit{\Phi}$ remaining unchanged. If all processors are busy, then service batches can clear jobs down to level $c-1$ inclusive; *i.e.* there is *unbounded* batch flow to levels c and above, and *bounded* (truncated) batch flow to $c-1$. If any processors are idle, then the processing batch size is exactly 1, as there are no jobs in the waiting room, and a processor can only clear its own job in service. The LHS is thus:

$$\begin{aligned}\mathbf{r}_j^{\text{MM CPP/GE/c/L}} =& \sum_{i=0}^{j-1} \mathbf{v}_i \mathit{\Lambda} (I - \mathit{\Theta})^{f_{j<L}} \mathit{\Theta}^{j-i-1} \\ &+ \mathbf{v}_j \left[Q - \mathit{\Lambda} f_{j<L} - C_j f_{j>0} \right] \\ &+ \sum_{i=j+1}^{L} \mathbf{v}_i C_i (I - \mathit{\Phi})^{f_{j>c-1}} \mathit{\Phi}^{i-j-1} f_{\substack{(i=j+1) \\ \vee (j \geq c-1)}}\end{aligned}$$

The switch $f_{j>c-1}$ bounds downward flow at level $c-1$. The term $f_{(i=j+1)\vee(j\geq c-1)}$ selects valid flows, which are batches from anywhere in the waiting room to just below c, or single jobs from a single processor.

2.4 Breakdowns and Repairs

We treat breakdowns and repairs (in the sense used by Mitrani and Chakka [1]) by allowing the number of processors to vary from 0 to c across the modulation phases. Thus the number of phases $N = c + 1$. This is introduced into the left-hand side expression by replacing references to c with the vector $(c_1, \ldots, c_N)$ of the numbers of operational processors in each modulation state. Our left-hand side expression thus becomes:

$$\begin{aligned}\mathbf{r}_j^{\text{MM CPP/GE/Mc/L}} = & \sum_{i=0}^{j-1} \mathbf{v}_i \mathit{\Lambda}(I - \mathit{\Theta})^{f_{j<L}} \mathit{\Theta}^{j-i-1} \\ & + \mathbf{v}_j \left[Q - \mathit{\Lambda} f_{j<L} - C_j f_{j>0} \right] \\ & + \sum_{i=j+1}^{L} \mathbf{v}_i C_i (I - \mathit{\Phi})^{F_{j>c_m-1}} \mathit{\Phi}^{i-j-1} F_{\substack{(i=j+1) \\ \vee(j \geq c_m - 1)}}\end{aligned}$$

where $F_{P(m)}$ is a diagonal matrix of values whose i^{th} diagonal element is $f_{P(i)}$. We define the result of raising a square matrix A to the power B with the same dimensions to be a simlar matrix of elements $a_{i,j}^{b_{i,j}}$. (In fact all matrices operated upon here are diagonal.) Also, the m^{th} element of C_j is now $\min(j, c_m)\mu$, where $c_m = (m-1)$. We combine breakdowns and repairs with modulated arrivals by the standard technique of taking the Kronecker product of the independent modulation matrices.

2.5 Negative customers

Negative customers [3] create additional downward flux which can be used to model network phenomena such as losses and load balancing. In addition to the diagonal rate and batch size matrices K and R, a killing mode has to be chosen. We consider three modes: t^v or "tail vulnerable" removes a job from the tail of the queue even if it is in service, t^s or "tail safe" removes a job from the tail of the queue but not when in service, and h^p or "head per" which removes a customer from the head of the queue (in service) at an independent but equal rate per processor, leading to a lower loss rate when some processors are inactive.

Killing mode t^v is the simplest, as it can kill any job in or out of service, and the batches are bounded only at level zero. Mode t^s is bound at level c, as it cannot kill any job in service. Mode h^p causes flux identical to processing completions.

$$\begin{aligned}\mathbf{r}_j = & \sum_{i=0}^{j-1} \mathbf{v}_i \big[\; \mathit{\Lambda}(1 - \mathit{\Theta})^{f_{j<L}} \mathit{\Theta}^{j-i-1} \big] && \Uparrow \\ & + \mathbf{v}_j \big[Q - \mathit{\Lambda} \, f_{j<L} - K \, f_{((j>\kappa) \vee \text{h}^p)} \beta_j - C_j \big] && \leftrightarrow \\ & + \sum_{i=j+1}^{L} \mathbf{v}_i \big[\; K(1 - R)^{f_{j>\kappa}} R^{i-j-1} \; f_{\substack{(j \geq \kappa) \\ \vee(\text{h}^p \wedge i=j+1)}} \; \beta_i && \downarrow \\ & \qquad + C_i (1 - \mathit{\Phi})^{f_{j>c-1}} \mathit{\Phi}^{i-j-1} f_{\substack{(i=j+1) \\ \vee(j \geq c-1)}} \big] && \Downarrow\end{aligned}$$

where κ is the lowest level reachable by killing, i.e. $\kappa_m = c_m f_{\text{t}^s} + (c_m - 1) f_{\text{h}^p}$, at which batch killing is truncated. We define the m^{th} diagonal element of the killing factor matrix at level (queue length) j, $\beta_j = \min(j, c_m) / \max_m(c_m)$ for h^p killing and $c_m / \max_m(c_m)$ otherwise.

2.6 Multiple streams

To add multiple streams, we augment the arrival term to be a sum of streams. For multiple arrivals this would take the form of

$$\mathbf{r}_j^{\text{MM CPP}_k \cdots} = \sum_{i=0}^{j-1} \mathbf{v}_i \sum_{k=1}^{n^{arr}} \mathit{\Lambda}_k (I - \mathit{\Theta}_k)^{f_{j<L}} \mathit{\Theta}_k^{j-i-1} + \ldots \text{(similarly to above)}$$

This introduction of sums of streams can also be performed for processing completions and negative customers to create a very general balance equation.

$$
\begin{aligned}
\mathbf{r}_j^{general} = & \sum_{i=0}^{j-1} \mathbf{v}_i \big[\ \textstyle\sum_{k=1}^{n^{arr}} \mathit{\Lambda}_k (1-\mathit{\Theta}_k)^{f_{j<L}} {\mathit{\Theta}_k}^{j-i-1} \big] \\
& + \mathbf{v}_j \big[Q - \textstyle\sum_{k=1}^{n^{arr}} \mathit{\Lambda}_k \, f_{j<L} - \sum_{k=1}^{n^{kill}} K \, f_{((j>\kappa)\vee \mathrm{h}^p)} \beta_j - \sum_{k=1}^{n^{serv}} C_{k,j} \big] \\
& + \sum_{i=j+1}^{L} \mathbf{v}_i \big[\ \textstyle\sum_{k=1}^{n^{kill}} K_k (1-R_k)^{f_{j>\kappa}} {R_k}^{i-j-1} \quad f_{\substack{(j\geq\kappa) \\ \vee(\mathrm{h}^p \wedge i=j+1)}} \quad \beta_i \\
& \qquad + \textstyle\sum_{k=1}^{n^{serv}} C_{k,i} (1-\mathit{\Phi}_k)^{f_{j>c-1}} {\mathit{\Phi}_k}^{i-j-1} \, f_{\substack{(i=j+1) \\ \vee(j\geq c-1)}} \ \big]
\end{aligned}
$$

3 Localized balance equation ensemble

We have a function $\mathbf{r}_j$ which, for a given level j, provides the (symbolic) left-hand side of a Kolmogorov equilibrium balance equation $\mathbf{r}_j = \mathbf{0}$. These balance equations generally include probability vectors from all levels of the queue due to the presence of batch arrivals and departures. We call these upward and downward streams respectively, referring to the direction of probability flux in the Kolmogorov balance equations. In order to efficiently solve for the steady-state solution of very large finite or infinite systems, we localize balance equations using a process we call *stream elimination*. Its aim shares a goal with Gaussian elimination - that of “diagonalizing” the system. A key difference is that stream elimination exploits the structure of the queue to enable it to solve infinite systems.

3.1 Localization approach

Each stream is eliminated by combining scalar multiples of two neighbouring left-hand side terms. The *target* level (usually called j) is the level to which the resulting localized left-hand side is deemed to pertain. The *eliminator* level provides the terms which will be subtracted to remove the stream.

3.2 Stream elimination example

We can eliminate any stream from a target level using any eliminator which has contributions of that same stream. For example, we can remove the summation terms associated with the stream of positive arrivals from any level using any other except level L (for a finite queue).

The closer the eliminator is to the target, the fewer terms for that stream remain. So, the choice is between using the level below, or the level above, and either would work for any given target level. There is in fact no choice, certainly for downward streams – and also for upward streams in a finite queue – if we consider the situation at the limits of the queue $j = 0$ and $j = L$, as we explain below.

The simplest process to consider is the arrival of batches of positive customers. These arrive at every level in the queue. The stream arriving at level L must be cancelled out using the level below, as there are none above. To cancel the positive arrival stream for target level $L-1$, we could use either the level above or below. To see that we cannot use the level above to get an independent equation, we will show that the determinant of a matrix representation of the solution would have a zero determinant. The balance equations for our class of queues can be written as follows:

$$\mathbf{v}A = \mathbf{0}$$

Where $\mathbf{v}$ is the row vector $(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_L)$, and the columns $(j-1)N+1$ through jN of A give the matrix coefficients of the vectors $\mathbf{v}_j$ for the target level j. The matrix A is as follows for an MM CPP/GE/1/4 queue with N modulation states:

$$
A = \begin{pmatrix}
(Q-\mathit{\Lambda}) & \mathit{\Lambda}(I-\mathit{\Theta}) & \mathit{\Lambda}(I-\mathit{\Theta})\mathit{\Theta} & \mathit{\Lambda}(I-\mathit{\Theta})\mathit{\Theta}^2 & \mathit{\Lambda}(I-\mathit{\Theta})\mathit{\Theta}^3 \\
C & (Q-\mathit{\Lambda}-C) & \mathit{\Lambda}(I-\mathit{\Theta}) & \mathit{\Lambda}(I-\mathit{\Theta})\mathit{\Theta} & \mathit{\Lambda}(I-\mathit{\Theta})\mathit{\Theta}^2 \\
C\mathit{\Phi} & C(I-\mathit{\Phi}) & (Q-\mathit{\Lambda}-C) & \mathit{\Lambda}(I-\mathit{\Theta}) & \mathit{\Lambda}(I-\mathit{\Theta})\mathit{\Theta} \\
C\mathit{\Phi}^2 & C(I-\mathit{\Phi})\mathit{\Phi} & C(I-\mathit{\Phi}) & (Q-\mathit{\Lambda}-C) & \mathit{\Lambda}(I-\mathit{\Theta}) \\
C\mathit{\Phi}^3 & C(I-\mathit{\Phi})\mathit{\Phi}^2 & C(I-\mathit{\Phi})\mathit{\Phi} & C(I-\mathit{\Phi}) & (Q-C)
\end{pmatrix}
$$

Our transformation of the system involves column operations, which we can achieve by post-multiplication of A by an appropriate matrix, say B. The rank of A is $5N-1$, *i.e.* one less than the number of states. When a column is removed and replaced by an expression reflecting the normalization constraint, the system is correctly specified, so we can solve it. If we postmultiply by a singular matrix, the rank of the system will be reduced, and will no longer be soluble with the normalization constraint. Thus, the determinant of B must be non-zero. For example, to remove the arrival stream terms in column five, we could post-multiply by a matrix B:

$$B = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & -\Theta \\ 0 & 0 & 0 & 0 & I \end{pmatrix}$$

The determinant of B is 1, so our solution is safe. This yields a modified A matrix:

$$\begin{pmatrix} (Q-\Lambda) & \Lambda(I-\Theta) & \Lambda(I-\Theta)\Theta & \Lambda(I-\Theta)\Theta^2 & 0 \\ C & (Q-\Lambda-C) & \Lambda(I-\Theta) & \Lambda(I-\Theta)\Theta & 0 \\ C\Phi & C(I-\Phi) & (Q-\Lambda-C) & \Lambda(I-\Theta) & 0 \\ C\Phi^2 & C(I-\Phi)\Phi & C(I-\Phi) & (Q-\Lambda-C) & \Lambda(I-\Theta)-(Q-\Lambda-C)\Theta \\ C\Phi^3 & C(I-\Phi)\Phi^2 & C(I-\Phi)\Phi & C(I-\Phi) & (Q-C)-C(I-\Phi)\Theta \end{pmatrix}$$

Now consider the removal of the positive arrival components from column 4 (queue level 3) *in addition.* We can achieve this by post multiplying A by one of the two following B matrices:

$$B_{\text{bad}} = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & \Theta & -\Theta \\ 0 & 0 & 0 & -I & I \end{pmatrix}, B_{\text{ok}} = \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & -\Theta & 0 \\ 0 & 0 & 0 & I & -\Theta \\ 0 & 0 & 0 & 0 & I \end{pmatrix}$$

These remove upward streams of positive arrivals from the balance equations for levels 3 and 4. B_{ok} removes the arrivals using the level above, and B_{bad} uses the level below as eliminator. Since columns four and five are linearly dependent, the determinant of B_{bad} is 0, so use of this matrix in multiplication would render the equation degenerate. The determinant of B_{ok} is 1, so it is safe to use.

A matrix B_{all} to remove all upward streams could be either of the following (since removing the upward stream at level 1 is not necessary):

$$B_{\text{all}} \begin{pmatrix} I & -\Theta & 0 & 0 & 0 \\ 0 & I & -\Theta & 0 & 0 \\ 0 & 0 & I & -\Theta & 0 \\ 0 & 0 & 0 & I & -\Theta \\ 0 & 0 & 0 & 0 & I \end{pmatrix} \text{ or } \begin{pmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & -\Theta & 0 & 0 \\ 0 & 0 & I & -\Theta & 0 \\ 0 & 0 & 0 & I & -\Theta \\ 0 & 0 & 0 & 0 & I \end{pmatrix}$$

Thus, we are constrained to eliminating upward streams using the balance equation at one level lower than the target level. A similar argument demands removal of downward streams using the raw left-hand side from the level above.

The expression resulting from elimination of a stream includes terms corresponding to the union of the queue levels of the constituent left-hand sides. In each modulation state m, there are a number, say u_m, of distinct upward streams to cancel out, which requires the use of left-hand sides from levels $j - u_m \ldots j - 1$. Similarly in the general case[2], there are d_m distinct downward streams to this level, which requires eliminators from $j + 1 \ldots j + d_m$. The (vector) balance equations cover the maximum range over the modulation states upward and downward. We define $\hat{u}$ as the maximum over m of u_m, and $\hat{d}$ similarly. A localized equation for level j therefore comprises levels $j - \hat{u}$ through $j + \hat{d}$.

3.3 The recursive elimination operator

We define the transformation operator $E^{\nu,X}_{\tau,j}$, to be applied to a left-hand side vector-valued *function* $\mathbf{r}$ (mapping integers j to left-hand sides $\mathbf{r}_j$). This returns a left-hand side without batch sums of the stream given by the diagonal matrix of rates X, *e.g.* $X = \Lambda$:

$$E^{\nu,X}_{\tau,j}\mathbf{r} = \mathbf{r}_j \frac{\partial \mathbf{r}_{j+\nu}}{\partial(\mathbf{v}_\tau X)} - \mathbf{r}_{j+\nu} \frac{\partial \mathbf{r}_j}{\partial(\mathbf{v}_\tau X)} = \mathbf{r}_j C + \mathbf{r}_{j+\nu} D \tag{2}$$

[2] within the processing region, tail-safe negative customers cannot operate, resulting in fewer downward streams within this region.

The index j is the target level for which the returned function is a localized Kolmogorov equation. In it, batches of the stream with matrix rate term X are removed by linear combination of the balance equation left-hand sides given by $\mathbf{r}$ at j and at the elimination level $j+\nu$. The weightings are determined by the partial derivatives which serve only to select the appropriate coefficients. We therefore eliminate downward streams (processing completions and the action of negative customers) by using terms above the target, so $\nu = 1$ for these. Upward streams (positive customer arrivals) are eliminated using terms below the target so $\nu = -1$ for these.

The index τ is the queue level at which the coefficients of X are calculated, and we call this the *test level.* This must be outside the range of levels whose left-hand sides are used in constructing the localized left-hand side for level j. We call this range the *elimination bracket.* There may be a non-zero common factor between the two resulting elimination terms C and D in equation (1), and we can divide through by this because we ultimately equate the resulting left-hand side to zero. For example, eliminating the positive arrival stream in an MM CPP/MM/1 queue with only batched positive arrivals for level 2 involves the following terms:

$$\begin{aligned}\frac{\partial \mathbf{r}_2}{\partial(\mathbf{v}_0\Lambda)} &= \frac{\partial}{\partial(\mathbf{v}_0\Lambda)}\Big(\mathbf{v}_0\Lambda(I-\Theta)\Theta + \mathbf{v}_1\Lambda(I-\Theta) + \mathbf{v}_2[Q-\Lambda-C] + \mathbf{v}_3C\Big)\\ &= (I-\Theta)\Theta\\ \frac{\partial \mathbf{r}_1}{\partial(\mathbf{v}_0\Lambda)} &= \frac{\partial}{\partial(\mathbf{v}_0\Lambda)}\Big(\mathbf{v}_0\Lambda(I-\Theta) + \mathbf{v}_1[Q-\Lambda-C] + \mathbf{v}_2C\Big)\\ &= (I-\Theta)\end{aligned}$$

We use these in the elimination function as follows:

$$\begin{aligned}E^{-1,\Lambda}_{0,2}\mathbf{r} &= \mathbf{r}_2\frac{\partial \mathbf{r}_1}{\partial(\mathbf{v}_0\Lambda)} - \mathbf{r}_1\frac{\partial \mathbf{r}_j}{\partial(\mathbf{v}_0\Lambda)}\\ &= \mathbf{r}_2(I-\Theta) - \mathbf{r}_1(I-\Theta)\Theta\\ &= \mathbf{r}_2 - \mathbf{r}_1\Theta,\ \text{cancelling the non-zero common factor, all } \theta_m < 1\\ &= \mathbf{v}_1[\Lambda-(Q-C)\Theta] + \mathbf{v}_2[Q-\Lambda-C(I+\Theta)] + \mathbf{v}_3C\end{aligned}$$

3.4 Degenerate streams during elimination

It is possible to have a mixture of null (having zero rate, or a zero batch distribution parameter) and non-null streams across the modulating states. When performing the E-transformation, for a phase with zero rate or batch parameter, the corresponding component of the vector left-hand side returned is simply the argument's component at j. This is because there are no non-local terms to remove. For example, to eliminate a negative customer stream from the raw balance left-hand side for a finite t^v queue, if $\delta_1 = 1/2$ and $\delta_2 = 0$, we have (taking $\mathbf{r}_{j,m}$ to be the m^{th} element of the vector returned by $\mathbf{r}_j$):

$$E^{+1,R}_{L,j}\mathbf{r} = \left(\underbrace{\mathbf{r}_{j,1}\frac{\partial \mathbf{r}_{j+1,1}}{\partial(\mathbf{v}_L\delta_1)} - \mathbf{r}_{j+1,1}\frac{\partial \mathbf{r}_{j,1}}{\partial(\mathbf{v}_L\delta_1)}}_{\substack{\text{elimination was required}\\ \text{in this modulation state}}},\quad \underbrace{\mathbf{r}_{j,2}}_{\text{no elimination}}\right)$$

Examining the elements of matrices A and B in equation 2, we note that whenever it is not necessary to eliminate a stream in modulation state m, the m^{th} elements of the diagonals of A and B are informative, as follows. Element $a_{m,m}$ is zero if the stream has already been removed. The value of $b_{m,m}$ is zero if either the stream has been removed, or the batch parameter of that stream is zero. The only case which requires care is when the stream has already been removed, wherein both $a_{m,m}$ and $b_{m,m}$ are zero, and we simply set $a_{m,m}$ to 1 if it is calculated in equation 2 as zero. This keeps the term from the source equation at the target level, this being $\mathbf{r}_{j,2}$ in the example above.

Equation 2 gives us a left-hand side *function of* j with stream (X,ν) removed. Elimination of further streams is simply the application of the same elimination procedure to the intermediate partially localized left-hand sides.

3.5 Worked example

We now peform a dry run of generating the localized balance equation for level 11 of a $c = 10$, t^v infinite waiting room MM CPP/GE/c/L G-queue. Equation 2 is used recursively. This creates branches which encompass r_{10} through r_{13}. Thus, the test level for upward streams must be lower than 10 and the test level for downward streams must be higher than 13. We choose $\tau^u = 14$ and $\tau^d = 9$ as these are the values used in normal operation of the system.

In our elimination terminology, this localized balance equation is $\mathbf{b}_{11} = E^{+1,K}_{14,11} E^{+1,C}_{14} E^{-1,\Lambda}_{10} \mathbf{r}$. The K (negative customer arrival stream) elimination uses equations at levels 11 and 12, which in turn involve application of the C (service completion) elimination term at, respectively, 11 and 12, and 12 and 13, and each of these involves an application of the Λ (positive arrivals) elimination term, which in total uses the raw balance equations for levels 10 through 13.

First, we calculate a leaf of this tree, eliminating the positive customer arrival summation terms by taking the raw balance equations (as given in equation 2.6) for levels 10 and 11 and combining them appropriately.

$$\begin{aligned}
\mathbf{r}_{10} = \ldots + \quad & \mathbf{v}_9[\Lambda \overbrace{(I-\Theta)}^{\frac{\partial \mathbf{r}_{10}}{\partial(\mathbf{v}_9\Lambda)}}] \\
+ \quad & \mathbf{v}_{10}[Q - \Lambda - K - C] \\
+ \quad & \mathbf{v}_{11}[K(I-R) + C(I-\Phi)] \\
+ \quad & \mathbf{v}_{12}[K(I-R)R + C(I-\Phi)\Phi] \\
+ \quad & \mathbf{v}_{13}[K(I-R)R^2 + C(I-\Phi)\Phi^2] \\
+ \quad & \mathbf{v}_{14}[K(I-R)R^2 + C(I-\Phi)\Phi^3] + \ldots
\end{aligned}$$

$$\begin{aligned}
\mathbf{r}_{11} = \ldots + \quad & \mathbf{v}_9[\Lambda \overbrace{(I-\Theta)\Theta}^{\frac{\partial \mathbf{r}_{11}}{\partial(\mathbf{v}_9\Lambda)}}] \\
+ \quad & \mathbf{v}_{10}[\Lambda(I-\Theta)] \\
+ \quad & \mathbf{v}_{11}[Q - \Lambda - K - C] \\
+ \quad & \mathbf{v}_{12}[K(I-R) + C(I-\Phi)] \\
+ \quad & \mathbf{v}_{13}[K(I-R)R + C(I-\Phi)\Phi] \\
+ \quad & \mathbf{v}_{14}[K(I-R)R + C(I-\Phi)\Phi^2] + \ldots
\end{aligned}$$

The coefficients of Λ for level 9 are $\frac{\partial \mathbf{r}_{10}}{\partial(\mathbf{v}_9\Lambda)} = (I-\Theta)$ and $\frac{\partial \mathbf{r}_{11}}{\partial(\mathbf{v}_9\Lambda)} = (I-\Theta)\Theta$. and hence in the ratio $I : \Theta$, so subtracting $\mathbf{r}_{10}\Theta$ from $\mathbf{r}_{11}$, we obtain $\mathbf{u}_{11} = E^{-1,\Lambda}_{9,11}\mathbf{r}$:

$$\begin{aligned}
\mathbf{u}_{11} = \quad & \mathbf{v}_{10}[\Lambda - (Q-K-C)\Theta] \\
+ \quad & \mathbf{v}_{11}[Q - \Lambda - K(I+\Theta(I-R)) - C(I+\Theta(I-\Phi))] \\
+ \quad & \mathbf{v}_{12}[K(I-R)(I-R\Theta) + C(I-\Phi)(I-\Phi\Theta)] \\
+ \quad & \mathbf{v}_{13}[K(I-R)R(I-R\Theta) + C(I-\Phi)\Phi(I-\Phi\Theta)] \\
+ \quad & \mathbf{v}_{14}[K(I-R)R^2(I-R\Theta) + C(I-\Phi)\Phi^2(I-\Phi\Theta)] \quad + \ldots
\end{aligned}$$

And we can write down the expressions for u_{12} and u_{13} directly from this, as the constituent behaviours are identical:

$$\begin{aligned}
\mathbf{u}_{12} = \quad & \mathbf{v}_{11}[\Lambda - (Q-K-C)\Theta] \\
+ \quad & \mathbf{v}_{12}[Q - \Lambda - K(I+\Theta(I-R)) - C(I+\Theta(I-\Phi))] \\
+ \quad & \mathbf{v}_{13}[K(I-R)(I-R\Theta) + C(I-\Phi)(I-\Phi\Theta)] \\
+ \quad & \mathbf{v}_{14}[K(I-R)R(I-R\Theta) + C\underbrace{(I-\Phi)\Phi(I-\Phi\Theta)]}_{\frac{\partial \mathbf{u}_{12}}{\partial(\mathbf{v}_{14}C)}} \quad + \ldots
\end{aligned}$$

$$\begin{aligned}
\mathbf{u}_{13} = \quad & \mathbf{v}_{12}[\Lambda - (Q-K-C)\Theta] \\
+ \quad & \mathbf{v}_{13}[Q - \Lambda - K(I+\Theta(I-R)) - C(I+\Theta(I-\Phi))] \\
+ \quad & \mathbf{v}_{14}[K(I-R)(I-R\Theta) + C\underbrace{(I-\Phi)(I-\Phi\Theta)]}_{\frac{\partial \mathbf{u}_{13}}{\partial(\mathbf{v}_{14}C)}} \quad + \ldots
\end{aligned}$$

To eliminate the processing stream, with rate matrix C, we take pairs of these equations and provide them to the elimination operator $E^{+1,C}_{14,j}$ to be applied at levels $11, 12, 13$. To calculate the expression for

$E^{+1,C}_{14,11}\mathbf{u}$ we find the coefficients of C at the test level in $\mathbf{u}_{12}$ and $\mathbf{u}_{13}$, which are respectively $(I-\varPhi)\varPhi(I-\varPhi\varTheta)$ and $(I-\varPhi)(I-\varPhi\varTheta)$. The ratio between these is $\varPhi : I$, so we subtract $\mathbf{u}_{12}\varPhi$ from $\mathbf{u}_{11}$ to yield the left-hand side $\mathbf{d}$ with only K-batch terms:

$$\begin{aligned}
\mathbf{d}_{11} = \quad & \mathbf{v}_{10}[\varLambda - (Q-K-M)\varTheta] \\
+ \quad & \mathbf{v}_{11}[Q(I+\varTheta\varPhi) - \varLambda(I+\varPhi) - K(I+\varTheta(I-R+\varPhi)) - C(I+\varTheta)] \\
+ \quad & \mathbf{v}_{12}[-Q\varPhi + \varLambda\varPhi + K((I-R)(I-R\varPhi) + (I+\varTheta)\varPhi(I-R)) + C] \\
+ \quad & \mathbf{v}_{13}[K(I-R)(I-R\varTheta)(R-\varPhi)] \\
+ \quad & \mathbf{v}_{14}[K\underbrace{(I-R)R(I-R\varTheta)(R-\varPhi)}_{\frac{\partial \mathbf{d}_{13}}{\partial(\mathbf{v}_{14}K)}}] + \ldots \text{only terms in } K
\end{aligned}$$

Again, the behaviour for level 12 is identical, so we immediately have $\mathbf{d}_{12}$:

$$\begin{aligned}
\mathbf{d}_{12} = \quad & \mathbf{v}_{11}[\varLambda - (Q-K-M)\varTheta] \\
+ \quad & \mathbf{v}_{12}[Q(I+\varTheta\varPhi) - \varLambda(I+\varPhi) - K(I+\varTheta(I-R+\varPhi)) - C(I+\varTheta)] \\
+ \quad & \mathbf{v}_{13}[-Q\varPhi + \varLambda\varPhi + K((I-R)(I-R\varPhi) + (I+\varTheta)\varPhi(I-R)) + C] \\
+ \quad & \mathbf{v}_{14}[K\underbrace{(I-R)(I-R\varTheta)(R-\varPhi)}_{\frac{\partial \mathbf{d}_{12}}{\partial(\mathbf{v}_{14}K)}}] + \ldots \text{only terms in } K
\end{aligned}$$

In these expressions, only the level immediately above the target contains terms associated with processing completions, as would have been the case with unbatched processing.

Assuming unequal batch terms (*i.e.* $R \neq \varPhi$), these can be combined using $E^{+1,K}_{14,j}\mathbf{d}$ to eliminate the final batch term in K by subtracting $\mathbf{d}_{12}R$ from $\mathbf{d}_{11}$ to give a localized balance $\mathbf{b}_{11}$:

$$\begin{aligned}
\mathbf{b}_{11} = \quad & \mathbf{v}_{10}[\varLambda - (Q-K-M)\varTheta] \\
+ \quad & \mathbf{v}_{11}[Q(I+\varTheta(\varPhi+R)) - \varLambda(I+\varPhi+R) \\
& \quad - K(I+\varTheta(I+\varPhi)) - C(I+\varTheta(I+R))] \\
+ \quad & \mathbf{v}_{12}[-Q(\varPhi+R(I+\varTheta\varPhi)) + \varLambda(\varPhi+R(I+\varTheta\varPhi)) \\
& \quad + K(I+\varPhi(I+\varTheta)) + C(I+R(I+\varTheta))] \\
+ \quad & \mathbf{v}_{13}[Q\varPhi R - \varLambda\varPhi R - K\varPhi - CR]
\end{aligned}$$

In this expression, we see two levels appear above the target level, one for each of negative customers and processing completions streams. This has the same form as the result found in [5].

Note that the coefficient of $\mathbf{v}_{13}$ is only strictly valid if it was necessary to remove all streams. The elimination ratio was determined from a term $K(I-R)(I-R\varTheta)(R-\varPhi)$, which contains zeros wherever the diagonal elements of R and $\varPhi$ are equal. In these circumstances, the information required to formulate an elimination ratio is not present. This is an example of stream degeneracy treated in section 3.4.

These coefficients in $\mathbf{b}_j$ pertain to balance equations for the repeating region of an MM CPP/GE/c/L G-queue described by a common balance equation for all levels, and describe a linear homogeneous matrix recurrence relation on the steady state SOP vectors. We can write this as $\mathbf{v}_{j-1}Q_0 + \mathbf{v}_jQ_1 + \mathbf{v}_{j+1}Q_2 + \mathbf{v}_{j+2}Q_3 = \mathbf{0}$, where

$$\begin{aligned}
Q_0 =& \varLambda - (Q-K-M)\varTheta \\
Q_1 =& Q(I+\varTheta)(\varPhi+R) - \varLambda(I+\varPhi+R) - K(I+\varTheta(I+\varPhi)) - C(I+\varTheta)(I+R) \\
Q_2 =& -Q(\varPhi+R(I+\varTheta\varPhi)) + \varLambda(\varPhi+R(I+\varTheta\varPhi)) + K(I+\varPhi(I+\varTheta)) \\
& + C(I+R(I+\varTheta)) \\
Q_3 =& Q\varPhi R - \varLambda\varPhi R - K\varPhi - CR
\end{aligned}$$

4 Conclusions

The problem of formulating these localized balance equations is entirely seperate to that of solving them. An example of their use can be found in [8], wherein queues with mutliple arrival processes are formulated. Treatment of such solution methods is beyond the scope of this paper.

Our methods facilitate routine implementation of new queueing paradigms based on geometrically batched processes. This is achieved by appropriate, simple modifications to the raw balance equations. Repeating regions, which map onto an eigensystem and can be solved using appropriate methods − are revealed when the localized equations in a contiguous region are of the form $\sum_{i=-u_k}^{d_k} \mathbf{v}_{j+i}.Q_{i+u_k} = \mathbf{0}$ for all levels j, ${\epsilon^b}_k + u_k \leq j \leq {\epsilon^t}_k - u_k$.

References

1. I. Mitrani and R. Chakka. Spectral expansion solution for a class of Markov models: Application and comparison with the matrix-geometric method, *Performance Evaluation* **23** pp. 241-260, 1995.
2. D.J. Thornley and Harf Zatschler. An automated formulation of queues with multiple modulated batches *submitted to* IFIP WG 7.3 International Symposium on Computer Performance Modeling, Measurement and Evaluation. (Performance 2002)
3. E. Gelenbe. Product form queueing networks with negative and positive customers, *Journal of Applied Probability* **28**, pp. 656-663, 1991.
4. Bini DA, Latouche G, Meini B. Solving matrix polynomial equations arising in queueing problems *Linear Algebra and its Applications* 340: 225-244 Jan 1 2002
5. R. Chakka and P.G. Harrison. A Markov modulated multi-server queue with negative customers - The MM CPP/GE/c/L G-queue. Acta Informatica **37**(11-12), pp. 881-919, 2001
6. Stephen Wolfram, *The Mathematica Book*, 4th ed., (Wolfram Media/Cambridge University Press, 1999) of solving them. An example of their use can be found in [8], wherein queues with mutliple arrival processes are formulated. Harrison. The MM CPP/GE/c/L G-queue:
7. Kouvatsos D, Awan I, *Entropy maximisation and open queueing networks with priorities and blocking* Performance Evaluation 51 (2003) 191-227.
8. P.G. Harrison, D.J. Thornley, H. Zatschler. *Geometrically batched networks.* in proceedings (ISCIS 17) Seventeenth International Symposium On Computer and Information Sciences October 28-30, 2002 University of Central Florida Orlando, Florida
9. M. Bhabuta and P.G. Harrison. Analysis of ATM traffic on the London MAN, Proceedings of the 4th International Conference on Performance Modelling and Evaluation of ATM Networks, Ilkely, Chapman and Hall, 1997.