Appendices

EC.1. Matrix Blocks for Model M_2

All blocks are square matrices of order $m+1$.

$$\mathbf{B} = \begin{pmatrix} -\lambda & & & \\ s\gamma_{0,1} & -\lambda - s\gamma_{0,1} & & \\ & \ddots & \ddots & \\ & & s\gamma_{0,m} & -\lambda - s\gamma_{0,m} \end{pmatrix}, \mathbf{A}_0 = \begin{pmatrix} \lambda & & & \\ & \lambda & & \\ & & \ddots & \\ & & & \lambda \end{pmatrix},$$

$$\mathbf{A}_1^{(i)} = \begin{pmatrix} -\lambda - b\mu_{i,0} & & & \\ (s-b)\gamma_{i,1} & -\lambda - b\mu_{i,1} - (s-b)\gamma_{i,1} & & \\ & \ddots & \ddots & \\ & & (s-b)\gamma_{i,m} & -\lambda - b\mu_{i,m} - (s-b)\gamma_{i,m} \end{pmatrix}, \qquad i < k,$$

$$\mathbf{A}_1^{(i)} = \begin{pmatrix} -\lambda - b\mu_{i,0} & & & \\ & -\lambda - b\mu_{i,1} & & \\ & & \ddots & \\ & & & -\lambda - b\mu_{i,m} \end{pmatrix}, \qquad i \ge k,$$

$$\mathbf{A}_2^{(i)} = \begin{pmatrix} b\mu_{i,0} & & & \\ & b\mu_{i,1} & & \\ & & \ddots & \\ & & & b\mu_{i,m} \end{pmatrix}, i \le k, \mathbf{A}_2^{(i)} = \begin{pmatrix} 0 & b\mu_{i,0} & & \\ & \ddots & \ddots & \\ & & 0 & b\mu_{i,m-1} \\ & & & b\mu_{i,m} \end{pmatrix}, \qquad i \ge k+1.$$

EC.2. Computing $\overrightarrow{q}^{(i)}_{j,h}$ and $E(\overrightarrow{X}^{(i)}_h)$

In this section, we discuss the computation of the $\overrightarrow{q}^{(i)}_{j,h}$ probabilities (EC.2.1), we discuss the computation of $E(\overrightarrow{X}^{(i)}_h)$ (EC.2.2), and we analyze the computational complexity of computing $\overrightarrow{\mathbf{R}}^{(i)}$ (EC.2.3).

EC.2.1. Computing $\overrightarrow{q}^{(i)}_{j,h}$

The approach that we explained in Subsection 5.1.1 calculates the elements $\overrightarrow{q}^{(i)}_{j,h}, j = 0, \ldots, h$ and $h = 0, \ldots, m-1$, one at a time, but it is more efficient to calculate all entries in a column of $\overrightarrow{\mathbf{q}}^{(i)}$ simultaneously. If we label the columns of matrix $\overrightarrow{\mathbf{q}}^{(i)}$ from 0 to m, then the non-zero elements of Column h ($\overrightarrow{q}^{(i)}_{0,h}, \ldots, \overrightarrow{q}^{(i)}_{h,h}$), $h = 0, \ldots, m-1$, are the total probabilities of all acyclic paths from state (i,j) to state $(i+1,h)$, $j = 0, \ldots, h$, as illustrated in Fig. EC.1. The general equations to compute Column h, $h = 0, \ldots, m-1$, are:

$$\begin{cases} \overrightarrow{q}^{(i)}_{h,h} = \overrightarrow{\phi}_{i,h}, & \\ \text{If } h \ge 1\text{: } \overrightarrow{q}^{(i)}_{j,h} = \overrightarrow{\phi}_{i,j}\delta^{i+1,h}_{i+1,j}, & j = 0, \ldots, h-1, \\ \text{If } h \ge 1\text{: } \delta^{i+1,h}_{i+1,b} = \overrightarrow{\phi}_{i+1,b}\delta^{i+1,h}_{i+2,b}, & b = 0, \ldots, h-1, \\ \text{If } h \ge 1\text{: } \delta^{i+1,h}_{a,b} = \overrightarrow{\psi}_{a,b}\delta^{i+1,h}_{a-1,b+1}, & a = i+2, \ldots, i+h+1; b = i+h+1-a, \\ \text{If } h \ge 2\text{: } \delta^{i+1,h}_{a,b} = \overrightarrow{\phi}_{a,b}\delta^{i+1,h}_{a+1,b} + \overrightarrow{\psi}_{a,b}\delta^{i+1,h}_{a-1,b+1}, & a = i+2, \ldots, i+h; b = 0, \ldots, i+h-a. \end{cases} \quad \text{(EC.1)}$$

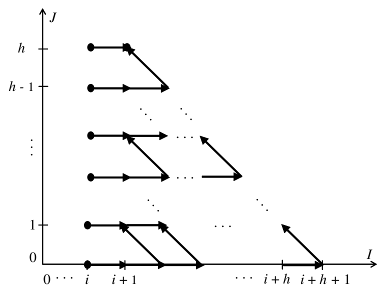


Figure EC.1 Paths from states (i,j) to state $(i+1,h)$, $j=0,\dots,h$.

These equations can be solved recursively, starting with the initial condition $\delta_{i+1,h}^{i+1,h}=1$.

We cannot use equation set (EC.1) for Column m of $\overrightarrow{\mathbf{q}}^{(i)}$, because there are infinitely many paths from state (i,j), $j=0,\dots,m$, to state $(i+1,m)$ due to the possible left and right transitions from states in Phase m without changing the phase. To calculate $\overrightarrow{q}_{j,m}^{(i)}$, $j=0,\dots,m-1$, we subtract the probability that a path enters $L(i+1)$ but returns to $L(i)$ before visiting state $(i+1,m)$ from the probability of a move from $L(i)$ to $L(i+1)$, that is:

$$\overrightarrow{q}_{j,m}^{(i)}=\begin{cases}\overrightarrow{\phi}_{i,j}-\sum_{l=j}^{m-1}\overrightarrow{\psi}_{i+1,l}\overrightarrow{q}_{j,l}^{(i)}, & j=0,\dots,m-1,\\ \overrightarrow{\phi}_{m,m}, & j=m.\end{cases}\tag{EC.2}$$

EC.2.2. Computing $E(\overrightarrow{X}_h^{(i)})$

We follow van Leeuwaarden and Winands (2006) for the computation of $E(\overrightarrow{X}_h^{(i)})$ with minor modifications to account for level-dependency. If $h<m$, then a right excursion from state (i,j), $i\geq k$, that visits state $(i+1,h)$ does so only once. At Phase m, however, given that state $(i+1,m)$ is visited once in a right excursion, the total number of visits to that state is geometrically distributed with parameter $\overrightarrow{\psi}_{i+1,m}$ (the probability that the next transition is a departure). Therefore,

$$E(\overrightarrow{X}_h^{(i)})=\begin{cases}1, & h=0,\dots,m-1,\\ 1/\overrightarrow{\psi}_{i+1,m}, & h=m.\end{cases}\tag{EC.3}$$

Substituting (EC.3) in (10) results in the following expressions for the non-zero entries of $\overrightarrow{\mathbf{R}}^{(i)}$:

$$\overrightarrow{R}_{j,h}^{(i)}=\begin{cases}\overrightarrow{q}_{j,h}^{(i)}\dfrac{\overrightarrow{\phi}_{i+1,h}}{\overrightarrow{\phi}_{i,j}}, & i=k,\dots,s';j=0,\dots,m;h=j,\dots,m-1,\\ \overrightarrow{q}_{j,m}^{(i)}\dfrac{\overrightarrow{\phi}_{i+1,m}}{\overrightarrow{\phi}_{i,j}\overrightarrow{\psi}_{i+1,m}}, & i=k,\dots,s';j=0,\dots,m,\end{cases}\tag{EC.4}$$

where $\overrightarrow{q}_{j,h}^{(i)}$, $h=0,\dots,m-1$, and $\overrightarrow{q}_{j,m}^{(i)}$ are obtained from (EC.1) and (EC.2).

EC.2.3. Computational Complexity of Computing $\overrightarrow{\mathbf{R}}^{(i)}$

We first derive the complexity of computing matrix $\overrightarrow{\mathbf{q}}^{(i)}$, which is derived by solving equation set (EC.1). Each equation in (EC.1) corresponds to a state in a right triangle in the state space

(Fig. EC.1), with $h+1$ states on the vertical leg, $h+1$ states on the hypotenuse, and a total of $(h+1)(h+2)/2$ states. Solving the equation for each of the $2h+1$ states on the vertical leg or on the hypotenuse requires one arithmetic operation and solving the equation for each of the other $h(h-1)/2$ states requires three operations, for a total of $(3h^2+h+2)/2$ operations to compute the entries in column h of $\overrightarrow{\mathbf{q}}^{(i)}$, for $h=0,\dots,m-1$. Computing column m using (EC.2) requires $m(m+1)$ operations. In total, the number of operations needed to compute all non-zero entries in the matrix $\overrightarrow{\mathbf{q}}^{(i)}$ is

$$m(m+1)+\sum_{h=1}^{m-1}\frac{3h^2+h+2}{2}=\frac{m^3+m^2+4m-1}{2}.$$

Once we have computed $\overrightarrow{\mathbf{q}}^{(i)}$, we perform two operations for $h=0,\dots,m-1$ and three operations for $h=m$ to compute each non-zero entry of the rate matrix $\overrightarrow{\mathbf{R}}^{(i)}$ using (EC.4). The total number of operations needed to compute $\overrightarrow{\mathbf{R}}^{(i)}$ is, therefore, $(m^3+4m^2+11m+4)/2$.

For comparison, Van Houdt and van Leeuwaarden (2011) developed an algorithm with computational complexity $2m^3$ to compute super-diagonals of matrix $\mathbf{G}$, which, in turn, is used to compute matrix $\mathbf{R}=\mathbf{A}_0\left(\mathbf{I}-\mathbf{A}_1-\mathbf{A}_0\mathbf{G}\right)^{-1}$, for $M/G/1$-type Markov chains with triangular $\mathbf{A}_0$, $\mathbf{A}_1$, and $\mathbf{A}_2$ matrices. Finding the main diagonal of $\mathbf{G}$ requires a version of Newton's method that converges quadratically. The Van Houdt and van Leeuwaarden (2011) algorithm appears to be the most efficient published algorithm that could be used (with some modifications to accommodate state-dependent rates) for our model, but our algorithm is more efficient and easier to implement.

Assumption 1 implies that the rate matrices for levels $s', s'+1,\dots$ are level-independent, that is, $\overrightarrow{\mathbf{R}}^{(s')}=\overrightarrow{\mathbf{R}}^{(s'+l)}, l\geq 1$. It follows that one needs to compute a total of $s'-k+1$ rate matrices $\overrightarrow{\mathbf{R}}^{(i)}$.

EC.3. Computing $\overleftarrow{q}^{(i)}_{j,h}$ and $E(\overleftarrow{X}^{(i)}_{h})$

In this section, we discuss the computation of the $\overleftarrow{q}^{(i)}_{j,h}$ probabilities (EC.3.1), we discuss the computation of $E(\overleftarrow{X}^{(i)}_{h})$ (EC.3.2), and we analyze the computational complexity of computing $\overleftarrow{\mathbf{R}}^{(i)}$ (EC.3.3).

EC.3.1. Computing $\overleftarrow{q}^{(i)}_{j,h}$

Let the lower-triangular matrix $\overleftarrow{\mathbf{q}}^{(i)}$, $i=1,\dots,k$, contain elements $\overleftarrow{q}^{(i)}_{j,h}$. Similar to $\overrightarrow{\mathbf{q}}^{(i)}$, we develop systems of linear equations to compute the entries of $\overleftarrow{\mathbf{q}}^{(i)}$. If we label the columns of matrix $\overleftarrow{\mathbf{q}}^{(i)}$ from $h=0$ to $h=m$, then the non-zero elements of Column h, $(\overleftarrow{q}^{(i)}_{h,h},\dots,\overleftarrow{q}^{(i)}_{m,h})$, are the total probabilities of all paths from states states $(i,h),\dots,(i,m)$ to state $(i-1,h)$.

As an example, Fig. EC.2 shows a directed graph that contains all possible paths from state (3,3) to state (2,2), assuming $k=3$ and $m=3$. Paths can include loops. For example, paths $(3,3)\rightarrow(2,3)\rightarrow(2,2)$ and $(3,3)\rightarrow(2,3)\rightarrow(1,3)\rightarrow(1,2)\rightarrow(2,2)$ do not have loops but path

$(3,3) \to (2,3) \to (1,3) \to (2,3) \to (2,2)$ loops between states $(2,3)$ and $(1,3)$ once. Denote the probabilities of an arrival, a departure, and a transition that decreases the overwork by one unit in state $(i,j) \in \Omega_{\text{Left}}$ by $\overleftarrow{\phi}_{i,j} = \lambda/(\lambda+(s-b)\gamma_{i,j}+b\mu_{i,j})$, $\overleftarrow{\psi}_{i,j} = b\mu_{i,j}/(\lambda+(s-b)\gamma_{i,j}+b\mu_{i,j})$, and $\varphi_{i,j} = (s-b)\gamma_{i,j}/(\lambda+(s-b)\gamma_{i,j}+b\mu_{i,j})$, respectively. Entry $\overleftarrow{q}^{(3)}_{3,2}$, the total probability of all paths from $(3,3)$ to $(2,2)$, is obtained by solving the following linear equations with variables $\delta^{i-1,h}_{a,b}, a = 0,1,2, b = 2,3$ that represent the total probability of all paths from (a,b) to $(i+1,h) = (2,2)$. The initial condition is $\delta^{2,2}_{2,2} = 1$:

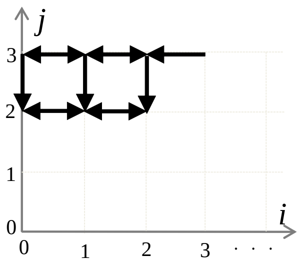


Figure EC.2 Paths from state $(3,3)$ to state $(2,2)$, $k \geq 3$.

$$\overleftarrow{q}^{(3)}_{3,2} = \overleftarrow{\psi}_{3,3}\delta^{2,2}_{2,3}, \qquad \delta^{2,2}_{0,3} = \varphi_{0,3}\delta^{2,2}_{0,2} + \overleftarrow{\phi}_{0,3}\delta^{2,2}_{1,3},$$
$$\delta^{2,2}_{2,3} = \overleftarrow{\psi}_{2,3}\delta^{2,2}_{1,3} + \varphi_{2,3}\delta^{2,2}_{2,2}, \qquad \delta^{2,2}_{1,2} = \overleftarrow{\psi}_{1,2}\delta^{2,2}_{0,2} + \overleftarrow{\phi}_{1,2}\delta^{2,2}_{2,2},$$
$$\delta^{2,2}_{1,3} = \overleftarrow{\psi}_{1,3}\delta^{2,2}_{0,3} + \varphi_{1,3}\delta^{2,2}_{1,2} + \overleftarrow{\phi}_{1,3}\delta^{2,2}_{2,3}, \qquad \delta^{2,2}_{0,2} = \overleftarrow{\phi}_{0,2}\delta^{2,2}_{1,2}.$$

The main diagonal entries of $\overleftarrow{\mathbf{q}}^{(i)}$ correspond to the probability of transitioning from (i,h) to $(i-1,h)$ through a service completion, so $\overleftarrow{q}^{(i)}_{h,h} = \overleftarrow{\psi}_{i,h}$, $h = 0,\ldots,m$. To compute the entries below the diagonal, for Column $h, h = 0,\ldots,m-1$, we begin by solving the following system of linear equations to obtain the first subdiagonal entry, $\overleftarrow{q}^{(i)}_{h+1,h}$, with the initial condition $\delta^{i-1,h}_{i-1,h} = 1$:

$$\begin{cases} \overleftarrow{q}^{(i)}_{h+1,h} = \overleftarrow{\psi}_{i,h+1}\delta^{i-1,h}_{i-1,h+1}, \\ \text{If } i \geq 2\text{: } \delta^{i-1,h}_{i-1,h+1} = \overleftarrow{\psi}_{i-1,h+1}\delta^{i-1,h}_{i-2,h+1} + \varphi_{i-1,h+1}\delta^{i-1,h}_{i-1,h}, \\ \text{If } i \geq 3\text{: } \delta^{i-1,h}_{a,h+1} = \overleftarrow{\psi}_{a,h+1}\delta^{i-1,h}_{a-1,h+1} + \varphi_{a,h+1}\delta^{i-1,h}_{a,h} + \overleftarrow{\phi}_{a,h+1}\delta^{i-1,h}_{a+1,h+1}, \qquad a = 1,\ldots,i-2, \\ \text{If } i \geq 3\text{: } \delta^{i-1,h}_{a,h} = \overleftarrow{\psi}_{a,h}\delta^{i-1,h}_{a-1,h} + \overleftarrow{\phi}_{a,h}\delta^{i-1,h}_{a+1,h}, \qquad a = 1,\ldots,i-2, \\ \delta^{i-1,h}_{0,h+1} = \varphi_{0,h+1}\delta^{i-1,h}_{0,h} + \overleftarrow{\phi}_{0,h+1}\delta^{i-1,h}_{1,h+1}, \\ \delta^{i-1,h}_{0,h} = \overleftarrow{\phi}_{0,h}\delta^{i-1,h}_{1,h}. \end{cases} \tag{EC.5}$$

After obtaining $\overleftarrow{q}^{(i)}_{h+1,h}$, the remaining off-diagonal entries of Column h, that is, $\overleftarrow{q}^{(i)}_{h+2,h}, \ldots, \overleftarrow{q}^{(i)}_{m,h}$, are computed by solving the system of linear equations (EC.6), separately for each entry. We improve computational efficiency by storing the values of the variables $\delta^{i-1,h}_{0,j}, \ldots, \delta^{i-1,h}_{i-1,j}$ found by solving for $\overleftarrow{q}^{(i)}_{j,h}$ and using those values when solving for $\overleftarrow{q}^{(i)}_{j+1,h}$. To obtain $\overleftarrow{q}^{(i)}_{j,h}$ for $j=h+2,\dots,m$ and $h=0,\dots,m-2$, we solve:

$$\begin{cases} \overleftarrow{q}^{(i)}_{j,h} = \overleftarrow{\psi}_{i,j}\delta^{i-1,h}_{i-1,j}, \\ \text{If } i\geq 2\text{: } \delta^{i-1,h}_{i-1,j} = \overleftarrow{\psi}_{i-1,j}\delta^{i-1,h}_{i-2,j} + \varphi_{i-1,j}\delta^{i-1,h}_{i-1,j-1}, \\ \text{If } i\geq 3\text{: } \delta^{i-1,h}_{a,j} = \overleftarrow{\psi}_{a,j}\delta^{i-1,h}_{a-1,j} + \varphi_{a,j}\delta^{i-1,h}_{a,j-1} + \overleftarrow{\phi}_{a,j}\delta^{i-1,h}_{a+1,j}, \qquad a=1,\dots,i-2, \\ \delta^{i-1,h}_{0,j} = \varphi_{0,j}\delta^{i-1,h}_{0,j-1} + \overleftarrow{\phi}_{0,j}\delta^{i-1,h}_{1,j}. \end{cases} \tag{EC.6}$$

EC.3.2. Computing $E(\overleftarrow{X}^{(i)}_h)$

The approach that we explained in EC.2.2 to compute $E(\overrightarrow{X}^{(i)}_h)$ does not apply for the computation of $E(\overleftarrow{X}^{(i)}_h)$ because of the structural differences between the transitions in Ω_{Right} and Ω_{Left}. Instead, we propose an efficient recursive equation for the computation of $E(\overleftarrow{X}^{(i)}_h)$ by exploiting the fact that once a left excursion has moved down from a phase, it will not return to that phase. A left excursion from state $(i,j), i\leq k$, might visit $(i-1,h)\in\Omega_{\text{Left}}$, one or more times, given that it visits $(i-1,h)$. It does so only once if a downward transition occurs in Phase h before returning to Level i and it does so more than once if the excursion loops in Phase h without a downward transition and then returns to Level i. Theorem EC.1 provides an efficient recursive equation to compute $E(\overleftarrow{X}^{(i)}_h)$. The efficiency of equation (EC.7) comes from the fact that the solutions for $i=k$ includes the solution for all Levels $i<k$. So, we need to solve (EC.7) only once, for $i=k$, in the whole process of computing all $\overleftarrow{\mathbf{R}}^{(i)}$ matrices.

Theorem EC.1. *The following equations hold:*

$$\begin{aligned} &E(\overleftarrow{X}^{(1)}_h) = 1, \quad h=0,\dots,m, \\ &E(\overleftarrow{X}^{(i)}_h) = \frac{1}{1-\overleftarrow{\psi}_{i-1,h}\overleftarrow{\phi}_{i-2,h}E(\overleftarrow{X}^{(i-1)}_h)}, \quad i=1,\dots,k; h=0,\dots,m. \end{aligned} \tag{EC.7}$$

Proof for Theorem EC.1. When $i=1$, a left recursion from state $(1,j)$ visits state $(0,h)$, $h\leq j$, then it will do so only once before returning to Level 1 because the next transition after visiting $(0,h)$ will either be an arrival, which terminates the excursion, or an overwork reduction, with no possibility of returning to Phase h before returning to Level 1.

For $i=2,\dots,k$, if the first transition after first visiting $(i-1,h)$ is an arrival or a downward transition, then the number of visits to $(i-1,h)$ during the left excursion will be one, by the same argument as for the $i=1$ case. However, if a left transition (a service completion) occurs with probability $\overleftarrow{\psi}_{i-1,h}$, there is a chance that $(i-1,h)$ is visited again only if whenever $(i-2,h)$ is visited, a right transition (an arrival), which occurs with probability $\overleftarrow{\phi}_{i-2,h}$, takes the excursion back to $(i-1,h)$. The expected number of visits to $(i-2,h)$, given that it is visited once from

$(i-1,h)$, before returning back to $(i-1,h)$, is equal to $E(\overleftarrow{X}_h^{(i-1)})$, which is the expected number of visits to $(i-2,h)$ in a left excursion starting from Level $i-1$, given that $(i-2,h)$ is visited once. So, we can write the expression for $E(\overleftarrow{X}_h^{(i)})$ as

$$E(\overleftarrow{X}_h^{(i)}) = 1 + \overleftarrow{\psi}_{i-1,h} E(\overleftarrow{X}_h^{(i-1)}) \overleftarrow{\phi}_{i-2,h} + \left(\overleftarrow{\psi}_{i-1,h} E(\overleftarrow{X}_h^{(i-1)}) \overleftarrow{\phi}_{i-2,h}\right)^2 + \dots, \tag{EC.8}$$

Let $G = \overleftarrow{\psi}_{i-1,h} E(\overleftarrow{X}_h^{(i-1)}) \overleftarrow{\phi}_{i-2,h}$. The right side of (EC.8) is a geometric series, which converges to $1/(1-g)$ if $g<1$ and diverges if $g \geq 1$. Instead of showing directly that $g<1$, we show that the left side of (EC.8) is finite, which implies that $g<1$, and that the right side of (EC.8) reduces to the right side of (EC.7).

Consider a version of Model M_2, call it M_2', where the downward transitions with rate $(s-b)\gamma_{i,j}$ are removed. Using the same line of reasoning as we used to compute $E(\overrightarrow{X}_h^{(i)})$ for Phase m, the total number of visits to state $(i-1,h)$ in a left excursion in Model M_2' is geometrically distributed with expected value $1/\overleftarrow{\phi}'_{i-1,h}$. The downward transitions reduce the expected number of visits to state $(i-1,h)$ during a left excursion, because a downward transition to Phase $h-1$ prevents any further visits to state $(i-1,h)$ during the left excursion. Therefore, $E(\overleftarrow{X}_h^{(i)}) < 1/\overleftarrow{\phi}'_{i-1,h}$. The quantity $1/\overleftarrow{\phi}'_{i-1,h}$ is finite, because $\lambda > 0$. Therefore, $E(\overleftarrow{X}_h^{(i)}) < \infty, g<1$, $E(\overleftarrow{X}_h^{(i)}) = 1/(1-g)$ as in (EC.7), and the proof is complete. □

EC.3.3. Computational Complexity of Computing $\overleftarrow{\mathbf{R}}^{(i)}$

We first derive the computational complexity of computing matrix $\overleftarrow{\mathbf{q}}^{(i)}$ by solving the linear equation systems (EC.5) and (EC.6). Each diagonal element $\overleftarrow{q}_{0,0}^{(i)}, \dots, \overleftarrow{q}_{m,m}^{(i)}$, requires one operation for a total of $m+1$ operations.

Solving a system of n linear equations in n unknowns by Gaussian elimination requires approximately $(2/3)n^3$ operations. The system of linear equations (EC.5) for $\overleftarrow{q}_{h+1,h}^{(i)}$ has $2i$ unknowns, $\delta_{0,h+1}^{i-1,h}, \dots, \delta_{i-1,h+1}^{i-1,h}, \delta_{0,h}^{i-1,h}, \dots, \delta_{i-1,h}^{i-1,h}$. Each system of linear equations (EC.6) for $\overleftarrow{q}_{j,h}^{(i)}$, $j = h+2, \dots, m$, has only i unknowns, $\delta_{0,j}^{i-1,h}, \dots, \delta_{i-1,j}^{i-1,h}$, as the other $\delta_{a,b}^{i-1,h}$ values, $a = 0, \dots, i-1$ and $b = h, \dots, j-1$, required to calculate $\overleftarrow{q}_{j,h}^{(i)}$, $j = h+2, \dots, m$, were obtained when solving the equations for $\overleftarrow{q}_{j-1,h}^{(i)}$. Therefore, the total number of operations needed to compute all off-diagonal entries of Column h, $h = 0, \dots, m-1$, of $\overleftarrow{\mathbf{q}}^{(i)}$ is approximately:

$$\frac{2}{3}\left((2i)^3 + (m-h-1)i^3\right).$$

Therefore, the total number of operations to compute all non-zero entries of $\overleftarrow{\mathbf{q}}^{(i)}$ is approximately:

$$(m+1) + \sum_{h=0}^{m-1} \frac{2}{3}\left((2i)^3 + (m-h-1)i^3\right) = 5i^3 m + \frac{1}{3} i^3 m^2 + m + 1 \approx \frac{1}{3} i^3 m^2.$$

The number of additional operations needed to compute $\overleftarrow{\mathbf{R}}^{(i)}, i = 1, \dots, k$, using (11), are at most proportional to m^2, so the highest order term for the approximate total number of operations remains $i^3 m^2/3$.

EC.4. Proof of Theorem 2

As mentioned in Section 5, one can view Model M_2 as a level-independent QBD process with rate matrix $\overrightarrow{\mathbf{R}}^{(s')}$ by considering all states from $(0,0)$ to $(s'-1,m)$ to be boundary states. Assumption 3 guarantees that M is irreducible. Therefore, Model M_2 is stable if and only if the following ergodicity condition for level-independent irreducible QBD processes is satisfied (Latouche and Ramaswami 1999, Theorem 7.2.4):

$$\boldsymbol{\nu}\mathbf{A}_0\mathbf{1} < \boldsymbol{\nu}\mathbf{A}_2^{(s')}\mathbf{1}, \tag{EC.9}$$

where the row vector $\boldsymbol{\nu} = (\nu_0, \ldots, \nu_m)$ contains the steady-state probabilities of the Markov chain that corresponds to the generator matrix $\mathbf{A}^{(s')} = \mathbf{A}_0 + \mathbf{A}_1^{(s')} + \mathbf{A}_2^{(s')}$. The Markov chain corresponding to $\mathbf{A}^{(s')}$ is a pure birth process with birth rates $s\mu_{s',0}, \ldots, s\mu_{s',m-1}$, all of which are positive by Assumption 3. Therefore, the steady-state probability vector ν equals $(0, \ldots, 0, 1)$. When we substitute ν in (EC.9), we obtain $\lambda < s\mu_{s',m}$.$\square$

EC.5. Wait Time Distribution When $s' = s$

The stationary probability $\overline{W}(x)$ that a user waits more than x time units before entering service in a queue that is modelled as a QBD process can be expressed as (Ramaswami and Lucantoni 1985, Theorem 4):

$$\overline{W}(x) = \sum_{n=0}^{\infty} d_n e^{-\theta x} \frac{(\theta x)^n}{n!}, \tag{EC.10}$$

where d_n is the probability that the user waits at least $n+1$ epochs of a uniformizing Poisson process with rate θ in the stochastically equivalent construction of the QBD process. For Model M_2,

$$\begin{aligned}
&\theta = \max_{j=0,\ldots,m} -\{\mathbf{A}_0 + \mathbf{A}_1^{(s)}\}_{j,j} = \max_j \{s\mu_{s,j}\}, \\
&d_n = \boldsymbol{\pi}_{s-1}\left(\mathbf{I} - \overrightarrow{\mathbf{R}}^{(s)}\right)^{-1} \overrightarrow{\mathbf{R}}^{(s)}\mathbf{H}_n\mathbf{e}, \qquad n = 0, 1, \ldots, \\
&\mathbf{H}_0 = \mathbf{I}, \qquad \mathbf{H}_{n+1} = \mathbf{H}_n\mathbf{P}_1 + \overrightarrow{\mathbf{R}}^{(s)}\mathbf{H}_n\mathbf{P}_2, \qquad n = 0, 1, \ldots, \\
&\mathbf{P}_1 = \frac{1}{\theta}(\mathbf{A}_0 + \mathbf{A}_1^{(s)}) + \mathbf{I}, \qquad \mathbf{P}_2 = \frac{1}{\theta}\mathbf{A}_2^{(s)}.
\end{aligned}$$

One can use two approaches to select an upper truncation limit UL for the infinite series (EC.10). The simpler approach is to set the upper limit equal to (Grassmann 1977, eq. (10))

$$UL = \theta x + 4\sqrt{\theta x} + 5, \tag{EC.11}$$

which guarantees that the truncation error $1 - \sum_{n=0}^{UL} e^{-\theta x}(\theta x)^n/n!$ is less than 10^{-4}.

The second approach requires more effort. Based on (EC.10), Ramaswami and Lucantoni (1985) derive the expected wait time in queue as,

$$W_q = \theta^{-1} \sum_{n=0}^{\infty} d_n. \tag{EC.12}$$

One can gradually increase the upper limit the upper limit UL for both (EC.10) and (EC.12) until the result of (EC.12) falls within an acceptable tolerance from the exact value for W_q obtained from the closed form solution in Table 1.

EC.6. Input Averaging and Output Averaging

Suppose we view the service rate μ in the Erlang C model as a random variable, with a distribution obtained from the steady-state probabilities for Model M_2, that is $\Pr\{\mu = \mu_{i,j}\} = \pi_{i,j}$ with $\overline{\mu} = E(\mu)$. Then Jensen's inequality implies that $\mathrm{E}[f(\mu)] \geq f(\bar{\mu})$ for the set of convex functions f, which implies that an output-averaging approximation is greater than or equal to an input-averaging approximation. We show that $C(.)$ and $D(.)$ are convex functions of μ, for sufficiently large values of μ to make the system stable.

The delay probability $C(.)$ is an increasing and convex function of $\rho = \lambda/(s\mu)$ when s is held constant (Harel 2010, Proposition 4). The utilization ρ is convex in μ. Therefore, $C(.)$ as a function of μ is a composition of a convex increasing function and a convex function, which implies that $C(.)$ is convex in μ. Grassmann (1983) shows that the average queue length in an Erlang C model is a convex function of μ. Combined with Little's Law, this result implies that the average delay $D(.)$ is convex with respect to μ.

EC.7. Additional Numerical Results

In the example of Section 6.3, we see that the net effect when the arrival rate increases is that the weighted average service rate decreases, that is, a slowdown effect (Fig. 13b). In this section, we perturb the example of Section 6.3 by varying μ_1, μ_2, μ_3, and μ_4 and the boundary lines that separate Regions 1-4. In both examples, we elevated the horizontal boundary line that separates Regions 1 and 2 from Regions 3 and 4 from $j = 1$ to $j = 35$, corresponding to one service completion per server. We show two experiments where the weighted average service rate increases with the arrival rate (net speedup) and where the weighted average service rate first increases and then decreases.

In the first experiment, we set $\mu_1 = 0.5$, $\mu_2 = 1$, $\mu_3 = 0.75$, and $\mu_4 = 0.45$. The state-dependent model is stable if $\lambda < 35 \times \mu_3 = 26.25$. The fixed-rate models have stability limits of $\lambda < 17.5, 35, 26.25$, and 15.75. Figs. EC.3-EC.4 compare the delay probability and the average delay (logarithmic scale) for Model M_2 and for the fixed-rate models. We fix the delay probabilities for fixed-rate models $\mu_1 = 0.5$ and $\mu_4 = 0.45$ to 1 for λ values beyond their stability limits in Fig. EC.3. Fig. EC.4 does not include the fixed-rate models for $\mu_1 = 0.5$ and $\mu_4 = 0.45$ as their stability limits are lower than the stability limit of Model M_2. As Fig. EC.6 shows the weighted average service rate increases with λ (net speedup). Figs. EC.7-EC.8 show input and output averaging approximations for this experiment. Our discussion about input and output averaging in Section EC.6 about delay probability does not hold for this experiment because when we extend the delay probability function $C(.)$ to equal 1 for $\mu < \lambda/s$, then $C(.)$ is not a convex function of μ, and μ_1 and μ_4 are smaller than λ/s for the values of λ shown in the figures for this example.

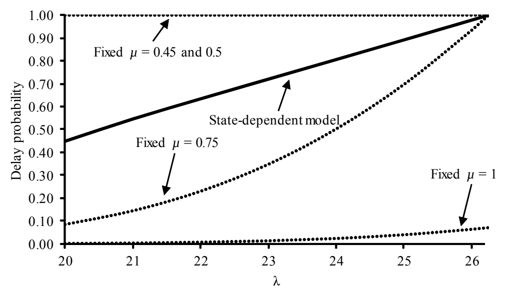


Figure EC.3 State-dependent vs. fixed-rate models: Delay probability.

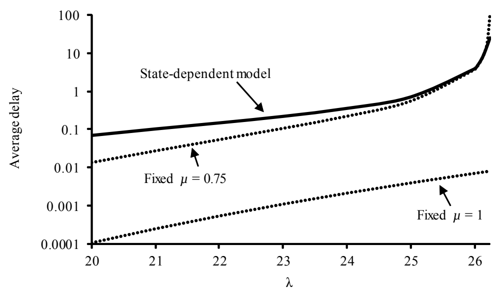


Figure EC.4 State-dependent vs. fixed-rate models: Average delay.

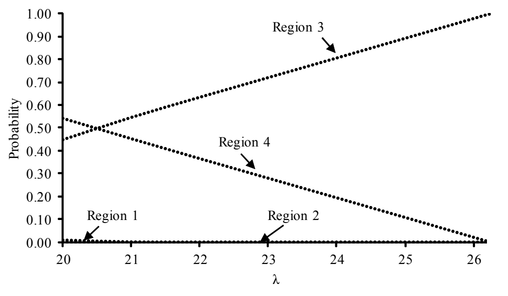


Figure EC.5 Total probabilities of service rate regions.

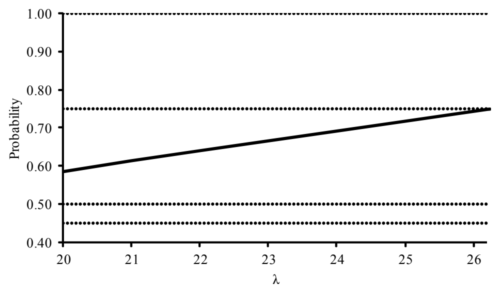


Figure EC.6 Weighted average service rate.

In the second experiment, we set $\mu_1 = 0.8$, $\mu_2 = 1$, $\mu_3 = 0.75$, and $\mu_4 = 0.8$. The state-dependent model is stable if $\lambda < 35 \times \mu_3 = 26.25$. Figs. EC.9 and EC.10 compare the delay probability and the average delay (logarithmic scale) for Model M_2 and for the fixed-rate models. As Fig. EC.12 shows, the average service rate goes first up and then down with λ. Figs. EC.13-EC.14 show input and output averaging approximations for this experiment.

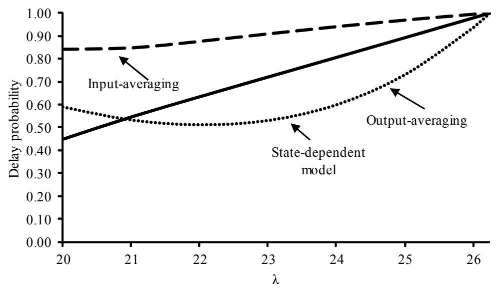


Figure EC.7 Delay prob. input and output averaging.

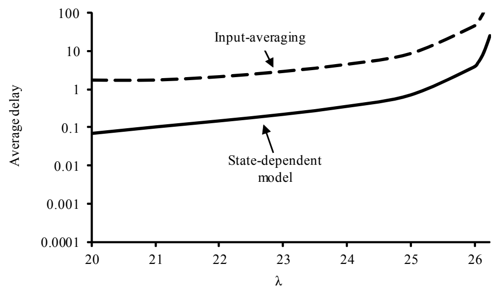


Figure EC.8 Mean delay input averaging.

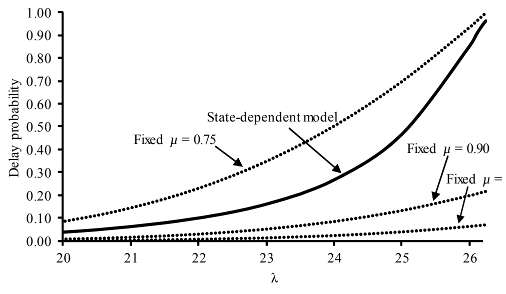


Figure EC.9 State-dependent vs. fixed-rate models: Delay probability.

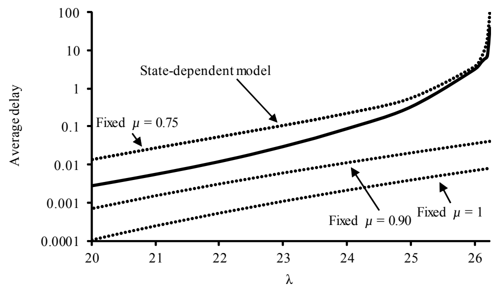


Figure EC.10 State-dependent vs. fixed-rate models: Average delay.

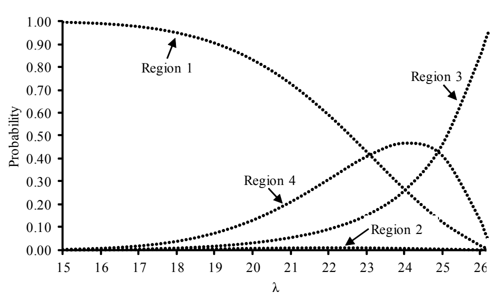


Figure EC.11 Total probabilities of service rate regions.

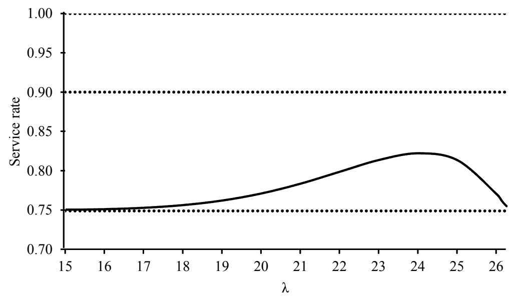


Figure EC.12 Weighted average service rate.

EC.8. Formulating and Solving Model M_3

Let $b_i = \lambda/(\min(i+1,s)\mu_{i+1}), i = 0,1,\ldots$. Note that for $i > s''$, $b_i = b_{s''} = \lambda/(s\mu_{s''}) < 1$ because of the stability assumption that $\lambda < s\mu_{s''}$. The steady-state probabilities satisfy the following recursions:

$$\pi_{i+1} = b_i\pi_i, \qquad i = 0,\ldots,s''-1,$$
$$\pi_{s''+i} = b_{s''}^i \pi_{s''}, \qquad i = 1,2,\ldots$$

The steady-state delay probability C is:

$$C = \sum_{i=s}^{s''} \pi_i + \sum_{i=s''+1}^{\infty} \pi_i = \sum_{i=s}^{s''} \pi_i + \pi_{s''} \sum_{i=1}^{\infty} b_{s''}^i = \sum_{i=s}^{s''} \pi_i + \pi_{s''} \frac{b_{s''}}{1-b_{s''}}.$$

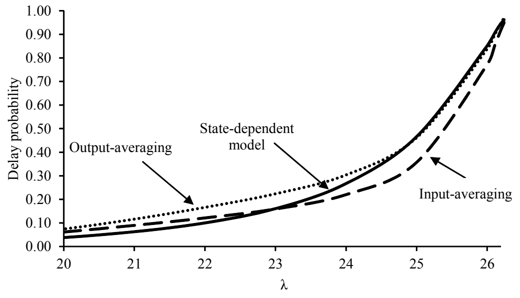


Figure EC.13 Delay prob. input and output averaging.

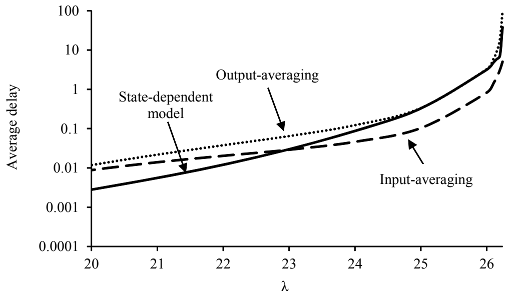


Figure EC.14 Mean delay input and output averaging.

```
Output: $C$
Input: $\lambda, \mu_1, \ldots, \mu_{s''}, s$
Initialization: $i \leftarrow 0, q \leftarrow 1, C \leftarrow 0, \text{converge} \leftarrow \text{FALSE}$
While not $\text{converge}$
    If $i \le s'' - 1$,
      $i \leftarrow i + 1$
      $b \leftarrow \lambda/(\min(i, s)\mu_i)$
      $q \leftarrow bq$
      If $i \ge s$ then $C \leftarrow C + q$
      Normalize: $\Sigma \leftarrow 1 + q, q \leftarrow q/\Sigma, C \leftarrow C/\Sigma$
    Else
      Geometric upper tail: $q \leftarrow qb/(1-b), C \leftarrow C + q$
      Normalize: $\Sigma \leftarrow 1 + q, q \leftarrow q/\Sigma, C \leftarrow C/\Sigma$
      $\text{converge} \leftarrow \text{TRUE}$
Return $C$.
```

Figure EC.15 Algorithm to compute the delay probability $C(\lambda, \mu_1, \ldots, \mu_{s''}, s)$.

The steady-state expected queue length is:

$$\begin{aligned}
Lq &= \sum_{i=s}^{s''}(i-s)\pi_i + \sum_{i=s''+1}^{\infty}(i-s)\pi_i = \sum_{i=s}^{s''}(i-s)\pi_i + \pi_{s''}\sum_{i=1}^{\infty}(s''-s+i)b_{s''}^i \\
&= \sum_{i=s}^{s''}(i-s)\pi_i + \pi_{s''}(s''-s)\sum_{i=1}^{\infty}b_{s''}^i + \pi_{s''}\sum_{i=1}^{\infty}ib_{s''}^i = \sum_{i=s}^{s''}(i-s)\pi_i + \pi_{s''}(s''-s)\frac{b_{s''}}{1-b_{s''}} + \pi_{s''}\frac{b_{s''}}{(1-b_{s''})^2} \\
&= \sum_{i=s}^{s''}(i-s)\pi_i + \pi_{s''}\frac{b_{s''}}{1-b_{s''}}\left(s''-s+\frac{1}{1-b_{s''}}\right).
\end{aligned}$$

Using Little's Law, the expected delay, D, is $D = Lq/\lambda$. Figs. EC.15 and EC.16 show pseudo code for algorithms to compute C and D, respectively.

We solved the nonlinear program for experiments 6.2 and 6.3 using Artelys Knitro (Byrd et al. 2015) on an iMac Retina 5K desktop computer with a 4 GHz Intel i7 processor, 16 GB 1600 MHz

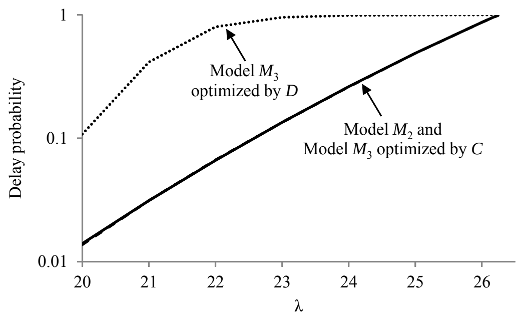
```
Output: $D$
Input: $\lambda, \mu_1, \ldots, \mu_{s''}, s$
Initialization: $i \leftarrow 0, q \leftarrow 1, D \leftarrow 0, \text{converge} \leftarrow \text{FALSE}$
While not $\text{converge}$
    If $i \le s'' - 1$,
      $i \leftarrow i + 1$
      $b \leftarrow \lambda / (\min(i, s)\mu_i)$
      $q \leftarrow bq$
      If  $i > s$  then  $D \leftarrow D + (i - s)q$
      Normalize: $\Sigma \leftarrow 1 + q, q \leftarrow q/\Sigma, D \leftarrow D/\Sigma$
    Else
      Geometric upper tail:  $q \leftarrow qb/(1-b), D \leftarrow D + q(s'' - s + 1/(1-b))$
      Normalize: $\Sigma \leftarrow 1 + q, q \leftarrow q/\Sigma, D \leftarrow D/\Sigma$
      $\text{converge} \leftarrow \text{TRUE}$
$D \leftarrow D/\lambda$
Return $D$.
```

Figure EC.16 Algorithm to compute the average delay $D(\lambda, \mu_1, \ldots, \mu_{s''}, s)$.

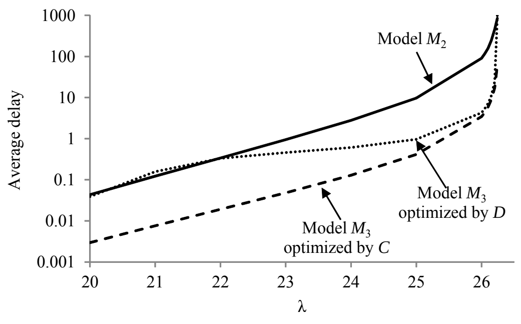
DDR3 ram, and Windows 7 x64. We set $\epsilon_1 = \epsilon_2 = \epsilon = 10^{-5}$. We called Knitro from Matlab and we programmed the algorithms in Figs. EC.15-EC.16 in Matlab. We ran each experiment for different values of s'' (40, 80, 150, 200, 300, and 600). For each s'', we used a multistart strategy, setting the initial service rates for all states in Model M_3 to 0.1, 0.2, 0.3, 0.5, 0.7, 0.9, 1, 1.5, 2, 3, 5, and 6. We also tried generating starting solutions with service rates randomly drawn from a uniform $(\epsilon, 10)$ distribution, but we found that starting solutions with fixed service rates consistently resulted in better final solutions. For each experiment, we report the best solution found. The time needed to solve the nonlinear program with Knitro's default settings increased from about 1 minute for $s'' = 40$, to 20 minutes for $s'' = 200$, to 2 hours for $s'' = 600$.

In Section 6.4, we presented the results of the nonlinear program for experiments 6.2 and 6.3. For experiment 6.2, we found a set of optimized service rates that provides good fits for C and D by setting $s'' = 200$. For experiment 6.3, the nonlinear program did not provide good fit for D even when we increased s'' to 600. Fig. EC.17 shows the nonlinear program results for $s'' = 40$, which is the smallest s'' value that we tried and the worst fit. The results for $s'' = 600$ are presented in Section 6.4. Fig. EC.18 shows the improvement in the root-mean-square error (RMSE) of the approximations to C and D produced by Model M_3 in experiment 6.3 for larger values of s''. At $s'' = 600$, the RMSE plots almost level off.

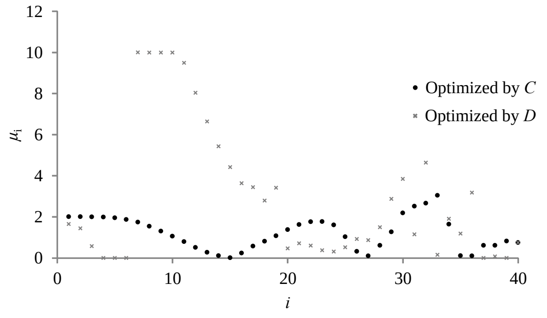
One issue that we noticed in our experiments is the wide fluctuations and sharp peaks in the optimized service rates that makes physical interpretation of the service rates implausible. This can be observed in Fig. EC.17c, for example. For experiment 6.3, we continued to see such fluctuations



(a) Delay probability (logarithmic scale)



(b) Average delay (logarithmic scale)



(c) Optimal service rates

Figure EC.17 Exp. 6.3: Optimizing the one-dimensional Model M_3, $s'' = 40$.

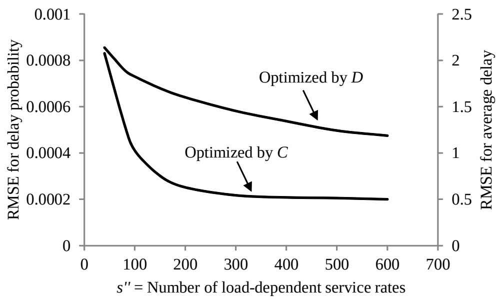


Figure EC.18 Exp. 6.3: RMSE improvement with s''.

for $s'' = 200$. To investigate whether we could obtain smoother service rates, we added the following constraints to the nonlinear program of the load and overwork scenario (experiment 6.3) with the objective function of fitting for D ($\alpha = 0$) and $s'' = 200$:

$$-\eta \le \mu_{i+1} - \mu_i \le \eta, \qquad i = 1, \ldots, s'' - 1.$$

These constraints prevent the service rates in two consecutive states from deviating by more than η. We experimented with different η values to investigate the sensitivity of the objective function and the service rate fluctuations to η in Figs. EC.19-EC.20. As expected, Fig. EC.19 shows that the fit for D improves as η increase. It also shows that we can reduce η from 5 to 1 to get smoother service rates without noticeably hurting the objective function. Fig. EC.20 compares service rates for three η values. For $\eta = 1$ and 5, we see large fluctuations in service rates ranging from almost 0 to 10, which makes physical interpretation of the service rates as representing server speed implausible. It is difficult to believe that servers speed up rapidly when the number of customers in the system increases to 15 customers and then slow down sharply to the service rates of almost 0 with 25 customers in the system. The plot for $\eta = 0.1$ shaves off the highest service rate peaks but it comes at the price of hurting the fit and the objective function.

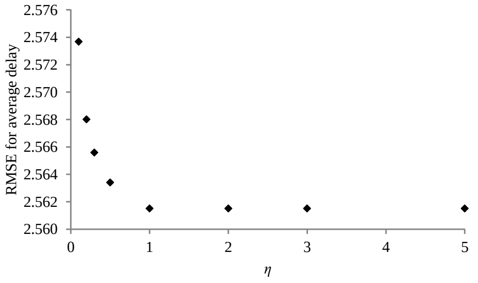


Figure EC.19 Exp. 6.3: Effect of η on RMSE.

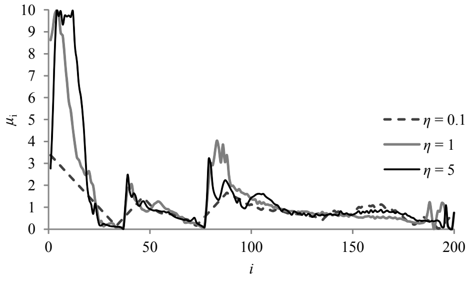


Figure EC.20 Exp. 6.3: Service rate fluctuations.

References

Byrd RH, Nocedal J, Waltz RA (2006) KNITRO: An integrated package for nonlinear optimization. In *Large-scale nonlinear optimization* (Springer US.): 35–59.

Grassmann WK (1977) Transient solutions in Markovian queueing systems. *Computers and Operations Research* 4(1): 47–54.

Grassmann W (1983) The convexity of the mean queue size of the $M/M/c$ queue with respect to the traffic intensity. *Journal of Applied Probability* 20(4): 916–919.

Harel A (2010) Sharp and simple bounds for the Erlang delay and loss formulae. *Queueing Systems* 64(2): 119–143.

Latouche G, Ramaswami V (1999) *Introduction to Matrix Analytic Methods in Stochastic Modeling* (ASA SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia).

Ramaswami V, Lucantoni DM (1985) Stationary waiting time distribution in queues with phase type service and in quasi-birth-and-death processes. *Stochastic Models* 1(2): 125–136.