Queueing Models and
Service Management
Vol. 1, No. 1, page 141-176, 2018

QMSM
© PU 2018

A 2-Class Maintenance Model with a Finite Population and Competing Exponential Failure Rates

Kevin Granville and Steve Drekic*

Department of Statistics and Actuarial Science
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada N2L 3G1

(Received March 2017; accepted October 2017)

Abstract: We investigate a maintenance system represented as a single-server polling model. Within the model, we assume two classifications for the type of failure a machine may experience. There are C total machines in the system, which at any point in time are either working, in service, or waiting to be served in one of two queues. Working machines are subject to independent and identically distributed exponential failure rates. Machines are returned to working condition after eventually receiving service according to the class of their failure. Service and switch-in time distributions for each class are assumed to be phase-type. Multiple service policies are examined, including preemptive resume priority, non-preemptive priority, and exhaustive service. We model the system as a level-dependent quasi-birth-and-death process, and use matrix analytic techniques to compute the steady-state joint queue length distribution as well as the sojourn time distribution of a broken machine. We present several numerical examples which highlight the dependency of the expected number of working machines on factors such as the service policy and the probability of a non-zero switch-in time.

Keywords: Exhaustive service, maintenance model, phase-type distribution, polling model, priority service, quasi-birth-and-death process, switch-in times.

1. Introduction

A classic single-server polling model typically involves multiple queues of customers of one or more classifications (in terms of arrival rates and service time distributions), which are attended to by a lone server whose behavior in terms of determining which queue to visit, and for how long, is defined by a set service policy. When one considers modeling maintenance systems, polling model service policies may not be the first thing to come to mind. However, in the area of maintenance optimization, deciding what components or systems to repair, and when to repair them, are common queries. In fact, two of the first papers to model systems that we now identify as polling models were regarded as maintenance problems! Mack et al. [19] investigated the efficiency of a closed system of machines, which were serviced by a patrolling repairman who would visit each machine in a cyclic fashion. Mack [18] would go on to revisit and generalize this model, extending the constant repair times to discrete random variables.

Within this work, we aim to once again examine how to analyze a maintenance system through the use of a polling model framework. Specifically, we recognize that the closed nature of these systems (i.e., having a finite population of machines) permits a natural application of matrix analytic methods when treating the systems as level-dependent quasi-birth-and-death (QBD) models. In designing our particular model, we envision a production facility wherein every (identical) functioning machine is at risk of failure as long as it is working, and upon failure it is possible to immediately diagnose the

* Corresponding author
Email : sdrekic@uwaterloo.ca

classification of failure and group it with other machines requiring a similar amount or type of work to repair. For example, a server farm containing an inventory of identical computers which are able to immediately produce an error report after failing. We could divide these computers into groups that require little work to return to functionality, such as simply resetting the system or removing accumulated dust, or those that require more significant work, such as repairing or replacing hardware. Within such a system, the goal of a business is to maximize the number of functioning machines, and so knowing the optimal order to repair different classifications of failure is of great importance. We describe this ordering in terms of a polling model's service policy. For more information on the study of polling models, we direct the interested reader to the works of Takagi [27], Levy and Sidi [15], Vishnevskii and Semenova [28], Boon [3] and Boon et al. [4], as well as the references therein.

When we refer to a model as a maintenance system, it is immediately clear that repairs and/or replacements will be involved. There are, however, very distinct types of models that can claim this label. This depends on what, exactly, is being maintained over time. For instance, a model may concern itself with the condition of a central machine, rather than being directly connected to a queueing-related issue. For example, Alfa and Castro [2] found the steady-state probabilities for a discrete-time model of a single machine system that would fail after its natural lifespan, or have a chance to fail after each time increment, at which point it would be repaired or replaced. This work was similar to that of Neuts et al. [22], who considered a comparable continuous-time model where the failures occurred according to a Poisson process. Pérez-Ocón and Montoro-Cazorla [23] would later expand on the continuous-time model by providing a way to numerically solve for the transition probability function matrices (as functions of time) for each operation and repair state, among other contributions.

When considering maintenance in a queueing system, depending on whether the "server(s)" or the "customers" are the ones receiving repairs, the interpretation and analysis of the model will vary greatly. In the former, we may have a machine required to conduct service or perform some function that is at risk of breaking down over time, leaving customers to wait in their queue (or risk abandonment due to impatience) until the server is repaired. In the latter, the server may be a repairman who tends to a closed system of machines or components that "arrive" to the queue by failing, where they will wait to be repaired. This is the type of maintenance system that we analyze in this paper.

Within our customer-centric model, we apply a matrix analytic approach in our analysis. This is also a convenient tool for server-centric maintenance models. For instance, Yang et al. [29] used matrix methods in their investigation of a queueing system where the server would break down over time (reducing their rate of service), according to random shocks modeled by a Poisson process. This model was further generalized by Chakravarthy [6], who introduced a probability of a shock not affecting the server, if the server was idle at the time, and replaced the assumption of a Poisson process customer flow by a more flexible Markovian arrival process. This paper also utilized phase-type distributions for several key system characteristics such as the effective service time and repair duration. Further examples of other server-centric maintenance models which do not employ matrix analytic methods include the works of Hsu [11], Perry and Posner [24], and Peschansky and Kovalenko [25].

Clearly, the patrolling repairman models of Mack et al. [19] and Mack [18] are both examples of customer-centric maintenance models. Kim and Koenigsberg [13] also considered a customer-centric maintenance model, applying some of the results from Mack et al. [19] to examine the server utilization and efficiency of machines in a system consisting of a single server repairing machines on two rotating carousel conveyors. Within customer-centric maintenance models, another feature not discussed by these papers that can be used is to maintain an inventory of spare (or reserve) machines (e.g., see Kim and Dshalalow [12] and Buyukkramikli et al. [5]). If it is not too costly to invest in more machines than can be used at capacity, an inventory of spares can help maintain productivity while one or more machines are shut down for repair.

Finally, we cite the closed queueing model of Gross et al. [9], who considered a closed system of $M+y$ machines, up to M of which could be turned on and working at any time (i.e., the y machines in the maintenance float were spares), having competing exponential failure times which may result in either a minor or major repair being required. Every failed machine would either be routed to the minor or major repair node, and those that receive minor repair may still be routed through the major repair node prior to being returned to operation. Each repair node was permitted to have multiple servers in parallel, and the optimal selection of y as well as the number of servers at each queue was investigated. Every distribution was assumed to be exponential, so that the analysis of their system was in the style of Gordon and Newell [8] for queueing networks with exponential servers. The reason that we single out this paper is that the concept of a closed network of machines which suffer either minor or major failures according to competing exponential failure rates is, in a way, analogous to our model of interest. We divert, however, in that machines suffering minor failures are never routed through the major failures queue before becoming operational again, and we only have a single server who alternates between serving the two queues according to some specified service policy. Moreover, we do not assume the existence of a maintenance float of spare machines that can be functional, but not turned on. Madu [20] also considered a similar model to Gross et al. [9], differing in that only one machine could be turned on at a time, only a single server was at either repair node, and failed machines always had to initially go through the minor repair node prior to possibly being routed to the major repair node. Abboud [1] later developed an efficient iterative method to find the optimal number of servers and machines for the same model as Gross et al. [9].

While all being instances of customer-centric maintenance models, it is clear that through their connections to the analysis of Gordon and Newell [8], the models of Gross et al. [9], Madu [20], and Abboud [1] are also specific variations of closed queueing networks (see also Lin et al. [16] and Righter [26]). In fact, due to the closed nature of these maintenance systems, this connection is quite logical. If we consider a machine's time until failure as the duration of their "service" required before leaving their work node, then the machines may be routed between nodes (representing work or repair stations) after receiving service like customers within a closed queueing network.

In fact, this alternative framework could also be used to describe our model of interest when we apply a priority service policy, rather than interpreting it solely as a polling model, similar to the work of Morris [21] who considered a closed queueing network with two classes of customers having class-dependent priorities and service rates at each of two nodes. In this case, we would describe our model as having two nodes, one with an infinite number of servers (i.e., the exponentially distributed times until failure for working machines), and the other with a single server who repairs machines according to their priority level, which is randomly determined between two levels upon completion of service at the first node. Of course, machines being repaired by the lone server at the second node would have a service time distribution dependent on their priority class, and when alternating between serving different priority classes, the server may incur a switch-in time. However, as we are also interested in the exhaustive service discipline, we perform all of our analysis within this work in the context of a polling model.

The remainder of the paper is organized as follows. In Section 2, we present the mathematical details concerning our 2-class maintenance system of interest. In order to determine the steady-state joint queue length distribution as well as the sojourn time (i.e., waiting time plus time in service) distribution of a broken machine, we model our system as a level-dependent quasi-birth-and-death process and employ a matrix analytic solution procedure. The exhaustive and non-preemptive priority service policies are considered in Section 3, whereas the preemptive resume priority service policy is dedicated to Section 4. In Section 5, we apply our results and present a variety of numerical examples which investigate the effect that switch-in times and service policies have on important performance measures of interest, such as the mean sojourn time of a broken machine and the mean number of working machines. Finally, in Section 6, we summarize our results and indicate possible directions for future research.

2. The Maintenance Model

We introduce a maintenance system characterized as a polling model with two classes, each of which represents a different type of failure which may require differently distributed service times to repair by a lone mechanic. Let C be the total number of machines in the system, which are all simultaneously subject to exponential failure rates as long as they are working. Define α_i, $i=1,2$, to be the rate for class-i failures, so that each machine has a total failure rate of $\alpha=\alpha_1+\alpha_2$. Once a machine has failed (or arrived to class i), it waits in the i^{th} queue to be served on a first-come-first-served basis amongst other machines in that same queue. It is assumed that only one type of failure can happen to a machine at one time, and that the times until failure of each of the machines are independent. While not represented as being in a queue directly, we denote working machines as being of class 0. When the system is empty, the server will move to a location separate from either queue to idle. For notational convenience, we denote the event of the server being idle as the server visiting class 0. Figure 1 depicts our maintenance model, where solid black circles represent machines, m and n are the respective lengths of queues 1 and 2, and μ_i represents the rate of service completion present only while the server is working on machines with class-i failures.

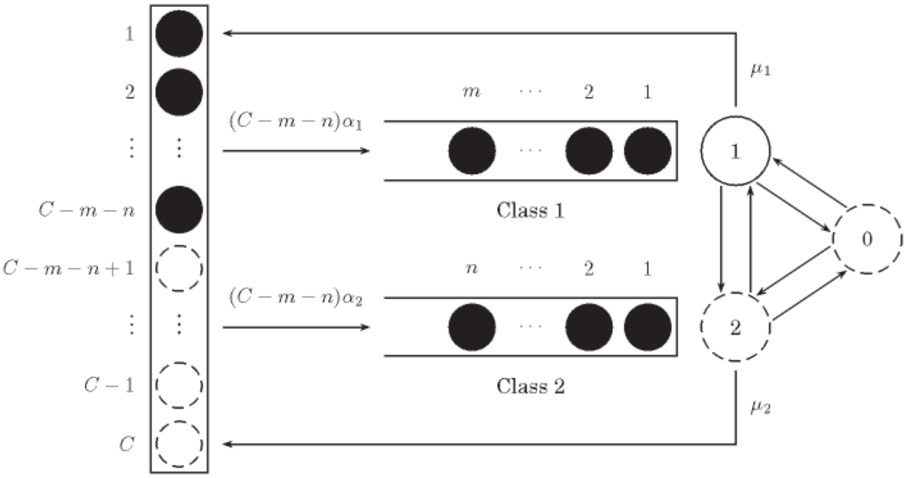

Figure 1: Depiction of the maintenance model.

The service policy that the mechanic (henceforth referred to as the server) uses to serve customers from either class may be *exhaustive*, where the server stays at one location and serves that class until its queue empties, or *priority-based*, preferring to serve one class (i.e., the high priority class) over the other (i.e., the low priority class). Among the priority policies, both *non-preemptive* and *preemptive resume* are considered. Under non-preemptive priority, the server switches to serve the high priority class if an arrival is observed while the server is idle or in a switch-in time, or after a service completion of the low priority class given that there are high priority customers waiting in their queue. Under preemptive resume priority, the server always switches to serve any high priority customers upon their arrival to the system. However, if the server happens to be serving a low priority customer at the time of switching, the partially rendered service of the interrupted customer is retained when the server eventually returns after emptying the high priority queue. Let $\mathcal{I}$ denote the type of service policy in place, such that

$$\mathcal{I}=\begin{cases}-2 & \text{, if class 2 has non-preemptive priority over class 1,}\\ -1 & \text{, if class 1 has non-preemptive priority over class 2,}\\ 0 & \text{, if the exhaustive service policy is in place,}\\ 1 & \text{, if class 1 has preemptive resume priority over class 2,}\\ 2 & \text{, if class 2 has preemptive resume priority over class 1.}\end{cases}$$

For $i=1,2$, class-i service times are assumed to be non-zero in duration, having a (continuous) phase-type distribution with representation $\mathrm{PH}(\underline{\beta}_i, B_i)$ of order b_i (e.g., see He [10], p. 10). Service times are assumed to be independent of each other and of the failure times. Similarly, class-i switch-in times are assumed to have a phase-type distribution with representation $\mathrm{PH}(\underline{\gamma}_{ji}, S_i)$ of order s_i. A class-i switch-in time can be understood as the period of time it takes the server to prepare before beginning work on the class-i queue, after previously attending to something else (e.g., serving customers in the queue of the other class or being idle). We allow the initial probability (row) vector $\underline{\gamma}_{ji}$ to depend on the class that the server is switching to (i.e., class i) and where the server is switching from (i.e., class j). We further assume that switch-in times are independent of the service and failure times, as well as the assumption that switching from a switch-in to class j is the same as switching from serving class j. For example, if class 1 has higher priority and the server is currently conducting a switch-in to class 2 when a class-1 failure is observed, the initial probability vector $\underline{\gamma}_{21}$ is used for the new switch-in to go serve class 1. In the same way, if the server switches after a class-2 service has completed (as in the case of class 2 emptying, or under non-preemptive priority), or during a class-2 service (as in the case of preemptive resume priority), $\underline{\gamma}_{21}$ is also used. Finally, for the switch-in times, we relax the non-zero duration assumption and let $\gamma_{ji}^{[0]} = 1 - \underline{\gamma}_{ji}\underline{e}'$ be the probability of a switch-in time from class j to class i being zero, where $\underline{e}'$ denotes an appropriately dimensioned column vector of ones (in general, the notation $'$ will be used to denote matrix transpose).

Due to the nature of this model, the analysis is conducted using matrix analytic methods. In particular, the system can be represented as a level-dependent QBD process, with the length of the class-1 queue serving as the *level* of the process. The associated infinitesimal generator is of the form

$$Q^{[C]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-2 \\ C-1 \\ C \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & C-2 & C-1 & C \end{array} \\ \left(\begin{array}{ccccccc} Q_{0,0}^{[C]} & Q_{0,1}^{[C]} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ Q_{1,0}^{[C]} & Q_{1,1}^{[C]} & Q_{1,2}^{[C]} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Q_{2,1}^{[C]} & Q_{2,2}^{[C]} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{C-2,C-2}^{[C]} & Q_{C-2,C-1}^{[C]} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{C-1,C-2}^{[C]} & Q_{C-1,C-1}^{[C]} & Q_{C-1,C}^{[C]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & Q_{C,C-1}^{[C]} & Q_{C,C}^{[C]} \end{array} \right) \end{array}, \tag{1}$$

where $\mathbf{0}$ represents an appropriately dimensioned zero matrix. Note that $Q^{[C]}$ is block-structured, in such a way that the sub-matrices (or blocks) $Q_{i,j}^{[C]}$ contain all transitions where the level changes from i to j. The particular forms of these blocks will be specified over the next two sections for each of the aforementioned service policies. In addition, the superscript $[C]$ of $Q^{[C]}$, as well as its associated blocks, corresponds to the number of machines in the system that is being modeled, and this choice of notation will be helpful in the upcoming sojourn time analysis.

The primary goal is to solve for the steady-state probability vector $\underline{\pi}$ of the process, and use it to derive the steady-state distribution of the amount of time it takes for a machine to be repaired and working again, after it has failed. We partition $\underline{\pi}$ as $\underline{\pi} = (\underline{\pi}_0, \underline{\pi}_1, \ldots, \underline{\pi}_C)$, where $\underline{\pi}_m$ denotes the vector of steady-state probabilities associated with the m^{th} level of the process, $m = 0,1,\ldots,C$. From Equation (1), we immediately obtain the following equilibrium equations expressed in terms of the blocks of $Q^{[C]}$:

$$\underline{0} = \underline{\pi}_0 Q_{0,0}^{[C]} + \underline{\pi}_1 Q_{1,0}^{[C]},$$

$$\underline{0} = \underline{\pi}_{m-1} Q_{m-1,m}^{[C]} + \underline{\pi}_m Q_{m,m}^{[C]} + \underline{\pi}_{m+1} Q_{m+1,m}^{[C]}, \ m = 1, 2, \ldots, C-1,$$

$$\underline{0} = \underline{\pi}_{C-1} Q_{C-1,C}^{[C]} + \underline{\pi}_C Q_{C,C}^{[C]},$$

where $\underline{0}$ denotes an appropriately dimensioned row vector of zeroes. Based on the procedure proposed by Gaver et al. [7], these equilibrium equations can be solved (in terms of $\underline{\pi}_0$) to obtain

$$\underline{\pi}_m = \underline{\pi}_0 \prod_{j=1}^{m} \mathcal{S}_j, \ m = 1, 2, \ldots, C, \tag{2}$$

where the set of matrices $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_C\}$ satisfy the recursive relation

$$\mathcal{S}_j = -Q_{j-1,j}^{[C]} (Q_{j,j}^{[C]} + \mathcal{S}_{j+1} Q_{j+1,j}^{[C]})^{-1}, \ j = 1, 2, \ldots, C-1,$$

with

$$\mathcal{S}_C = -Q_{C-1,C}^{[C]} (Q_{C,C}^{[C]})^{-1}.$$

If we define $\mathcal{S}_0 = Q_{0,0}^{[C]} + \mathcal{S}_1 Q_{1,0}^{[C]}$, then $\underline{\pi}_0$ clearly satisfies

$$\underline{\pi}_0 \mathcal{S}_0 = \underline{0}. \tag{3}$$

Moreover, since all steady-state probabilities must sum to 1, it follows that

$$1 = \underline{\pi}\underline{e}' = \sum_{m=0}^{C} \underline{\pi}_m \underline{e}' = \sum_{m=0}^{C} \underline{\pi}_0 \prod_{j=1}^{m} \mathcal{S}_j \underline{e}' = \underline{\pi}_0 \left(\sum_{m=0}^{C} \prod_{j=1}^{m} \mathcal{S}_j \underline{e}' \right), \tag{4}$$

where we adopt the convention that $\prod_{j=1}^{0} \mathcal{S}_j \underline{e}' = \underline{e}'$. Equations (3) and (4) provide a linear system of equations that may be solved for $\underline{\pi}_0$. Once $\underline{\pi}_0$ is obtained, we can recover each $\underline{\pi}_m$, $m = 1, 2, \ldots, C$, from Equation (2).

3. Exhaustive and Non-preemptive Priority Service Models

3.1. Steady-state probabilities

In this section, we focus solely on exhaustive and non-preemptive priority service policies (i.e., $\mathcal{I} \in \{-2, -1, 0\}$). As such, we need not consider server movements that interrupt the service of a customer from either class. To properly model the system, we must track four variables, namely (X_1, X_2, L, Y). Here, X_1 is the length of the class-1 queue (and is denoted as the level of the process), X_2 is the length of the class-2 queue, $L \in \{0, 1, 2, 3, 4, 5\}$ indicates the position of the server (0: server is idle; 1: switch-in to class 1; 2: serving class 1; 3: switch-in to class 2; 4: serving class 2; 5: switch-in to class 0), and Y denotes the phase of the service or switch-in time which has possible values depending on L in the following way:

$$Y \in \Omega_Y(L) = \begin{cases} \{0\} & \text{, if } L = 0, \\ \{1, 2, \ldots, s_1\} & \text{, if } L = 1, \\ \{1, 2, \ldots, b_1\} & \text{, if } L = 2, \\ \{1, 2, \ldots, s_2\} & \text{, if } L = 3, \\ \{1, 2, \ldots, b_2\} & \text{, if } L = 4, \\ \{1, 2, \ldots, s_0\} & \text{, if } L = 5. \end{cases}$$

Let $\pi_{m,n,l,y}$ be the steady-state probability that $X_1 = m$, $X_2 = n$, $L = l$, and $Y = y$, where $0 \le X_1 \le C$, $0 \le X_2 \le C - X_1$, and L and Y take values from their respective supports above. As a result, we have

$$\underline{\pi}_0 = (\pi_{0,0,0,0}, \pi_{0,0,5,1}, \ldots, \pi_{0,0,5,s_0}, \underline{\pi}_{0,1}, \ldots, \underline{\pi}_{0,C}),$$

where

$$\underline{\pi}_{0,n} = (\pi_{0,n,3,1}, \ldots, \pi_{0,n,3,s_2}, \pi_{0,n,4,1}, \ldots, \pi_{0,n,4,b_2})$$

is a row vector of length $s_2 + b_2$ for $n = 1, 2, \ldots, C$. For non-zero levels, the m^{th} steady-state probability row vector is given by

$$\underline{\pi}_m = (\underline{\pi}_{m,0}, \underline{\pi}_{m,1}, \ldots, \underline{\pi}_{m,C-m}), m \ge 1,$$

where

$$\underline{\pi}_{m,0} = (\pi_{m,0,1,1}, \ldots, \pi_{m,0,1,s_1}, \pi_{m,0,2,1}, \ldots, \pi_{m,0,2,b_1})$$

and (for $n = 1, 2, \ldots, C - m$)

$$\underline{\pi}_{m,n} = (\pi_{m,n,1,1}, \ldots, \pi_{m,n,1,s_1}, \pi_{m,n,2,1}, \ldots, \pi_{m,n,2,b_1}, \pi_{m,n,3,1}, \ldots, \pi_{m,n,3,s_2}, \pi_{m,n,4,1}, \ldots, \pi_{m,n,4,b_2})$$

are row vectors of length $s_1 + b_1$ and $s_1 + b_1 + s_2 + b_2$, respectively. Clearly, level 0 has $1 + s_0 + C(s_2 + b_2)$ states, whereas level m, $m \ge 1$, has $s_1 + b_1 + (C - m)(s_1 + b_1 + s_2 + b_2)$ states.

In order to determine $\underline{\pi}$ using the QBD procedure described in the previous section, we need only specify the blocks of $Q^{[C]}$ defined by Equation (1). In what follows, let $\delta_{i,j}$ be the standard Kronecker delta function which equals 1 if $i = j$ and 0 if $i \ne j$, and let I_i be an identity matrix of dimension i. Furthermore, let $\underline{B}_{0,i}' = -B_i \underline{e}'$ and $\underline{S}_{0,i}' = -S_i \underline{e}'$ be the absorption rate (column) vectors corresponding to phase-type representations $\text{PH}(\underline{\beta}_i, B_i)$ and $\text{PH}(\underline{\gamma}_{ji}, S_i)$, respectively. The diagonal blocks of $Q^{[C]}$ can be expressed as

$$Q_{0,0}^{[C]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-1 \\ C \end{array} \begin{array}{c} \begin{array}{cccccc} \quad 0 \quad & \quad 1 \quad & \quad 2 \quad & \cdots & C-1 & C \end{array} \\ \left(\begin{array}{cccccc} \Delta_0^{[C]} & C\alpha_2 \underline{e}' \left[\begin{array}{cc} \underline{\gamma}_{02} & \gamma_{02}^{[0]} \underline{\beta}_2 \end{array} \right] & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \left[\begin{array}{cc} \underline{0}' & \mathbf{0} \\ \gamma_{20}^{[0]} \underline{B}_{0,2}' & \underline{B}_{0,2}' \underline{\gamma}_{20} \end{array} \right] & \Delta_1^{[C]} & (C-1)\alpha_2 I_{s_2+b_2} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Gamma & \Delta_2^{[C]} & \ddots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \Delta_{C-1}^{[C]} & \alpha_2 I_{s_2+b_2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \Gamma & \Delta_C^{[C]} \end{array} \right) \end{array},$$

where

$$\Delta_n^{[C]} = \begin{cases} -C\alpha I_{1+s_0} + \left[\begin{array}{cc} 0 & \underline{0} \\ \underline{S}_{0,0}' & S_0 \end{array} \right] & \text{, if } n = 0, \\ -(C-n)\alpha I_{s_2+b_2} + \left[\begin{array}{cc} S_2 & \underline{S}_{0,2}' \underline{\beta}_2 \\ 0 & B_2 \end{array} \right] & \text{, if } n = 1, 2, \ldots, C, \end{cases}$$

and

$$\Gamma = \left[\begin{array}{cc} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{B}_{0,2}' \underline{\beta}_2 \end{array} \right],$$

while for $i = 1, 2, \ldots, C$,

$$Q_{i,i}^{[C]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-i-1 \\ C-i \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & \cdots & C-i-1 & C-i \end{array} \\ \begin{pmatrix} Q_{i,i,0}^{[C]} & (UD)_{i,0}^{[C]} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ (LD)_{i,1}^{[C]} & Q_{i,i,1}^{[C]} & (UD)_{i,1}^{[C]} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (LD)_{i,2}^{[C]} & Q_{i,i,2}^{[C]} & \ddots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{i,i,C-i-1}^{[C]} & (UD)_{i,C-i-1}^{[C]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & (LD)_{i,C-i}^{[C]} & Q_{i,i,C-i}^{[C]} \end{pmatrix} \end{array},$$

where

$$Q_{i,i,n}^{[C]} = \begin{cases} -(C-i)\alpha I_{s_1+b_1} + \begin{bmatrix} S_1 & \underline{S}'_{0,1}\underline{\beta}_1 \\ \mathbf{0} & B_1 \end{bmatrix} & \text{, if } n=0, \\ -(C-i-n)\alpha I_{s_1+b_1+s_2+b_2} + \begin{bmatrix} S_1 & \underline{S}'_{0,1}\underline{\beta}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & S_2 & \underline{S}'_{0,2}\underline{\beta}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & B_2 \end{bmatrix} & \text{, if } n=1,2,\ldots,C-i, \end{cases}$$

$$(UD)_{i,n}^{[C]} = \begin{cases} (C-i)\alpha_2 \begin{bmatrix} (1-\delta_{\mathcal{I},-2})I_{s_1} & \mathbf{0} & \delta_{\mathcal{I},-2}\underline{e}'\underline{\gamma}_{12} & \delta_{\mathcal{I},-2}\gamma_{12}^{[0]}\underline{e}'\underline{\beta}_2 \\ \mathbf{0} & I_{b_1} & \mathbf{0} & \mathbf{0} \end{bmatrix} & \text{, if } n=0, \\ (C-i-n)\alpha_2 I_{s_1+b_1+s_2+b_2} & \text{, if } n=1,2,\ldots,C-i-1, \end{cases}$$

and

$$(LD)_{i,n}^{[C]} = \begin{cases} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \underline{B}'_{0,2}\underline{\gamma}_{21} & \gamma_{21}^{[0]}\underline{B}'_{0,2}\underline{\beta}_1 \end{bmatrix} & \text{, if } n=1, \\ \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \delta_{\mathcal{I},-1}\underline{B}'_{0,2}\underline{\gamma}_{21} & \delta_{\mathcal{I},-1}\gamma_{21}^{[0]}\underline{B}'_{0,2}\underline{\beta}_1 & \mathbf{0} & (1-\delta_{\mathcal{I},-1})\underline{B}'_{0,2}\underline{\beta}_2 \end{bmatrix} & \text{, if } n=2,3,\ldots,C-i. \end{cases}$$

With regard to the off-diagonal blocks of $Q^{[C]}$, we first have

$$Q_{i,i+1}^{[C]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-i-1 \\ C-i \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & \cdots & C-i-1 \end{array} \\ \begin{pmatrix} (C-i)\alpha_1 I_{s_1+b_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & Q_{i,i+1,1}^{[C]} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{i,i+1,2}^{[C]} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{i,i+1,C-i-1}^{[C]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} \end{array}$$

for $i = 1, 2, \ldots, C-1$, where

$$Q_{i,i+1,n}^{[C]} = (C-i-n)\alpha_1 I_{s_1+b_1+s_2+b_2}, n = 1, 2, \ldots, C-i-1.$$

Moreover,

$$Q_{0,1}^{[C]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-1 \\ C \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & \cdots & C-1 \end{array} \\ \left(\begin{array}{ccccc} C\alpha_1 \underline{e}' \left[\begin{array}{cc} \underline{\gamma}_{01} & \gamma_{01}^{[0]} \underline{\beta}_1 \end{array} \right] & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & Q_{0,1,1}^{[C]} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{0,1,2}^{[C]} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{0,1,C-1}^{[C]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{array} \right) \end{array},$$

where

$$Q_{0,1,n}^{[C]} = (C-n)\alpha_1 \begin{bmatrix} \delta_{\mathcal{I},-1} \underline{e}' \underline{\gamma}_{21} & \delta_{\mathcal{I},-1} \gamma_{21}^{[0]} \underline{e}' \underline{\beta}_1 & (1-\delta_{\mathcal{I},-1}) I_{s_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & I_{b_2} \end{bmatrix}, \; n = 1, 2, \ldots, C-1,$$

and

$$Q_{1,0}^{[C]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-2 \\ C-1 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & C-2 & C-1 & C \end{array} \\ \left(\begin{array}{ccccccc} \left[\begin{array}{cc} \underline{0}' & 0 \\ \gamma_{10}^{[0]} \underline{B}_{0,1}' & \underline{B}_{0,1}' \underline{\gamma}_{10} \end{array} \right] & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Q_{1,0}^{\star} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{1,0}^{\star} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{1,0}^{\star} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & Q_{1,0}^{\star} & \mathbf{0} \end{array} \right) \end{array},$$

where

$$Q_{1,0}^{\star} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \underline{B}_{0,1}' \underline{\gamma}_{12} & \gamma_{12}^{[0]} \underline{B}_{0,1}' \underline{\beta}_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Finally, for $i = 2, 3, \ldots, C$, the remaining blocks of $Q^{[C]}$ are of the form

$$Q_{i,i-1}^{[C]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-i-1 \\ C-i \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & C-i-1 & C-i & C-i+1 \end{array} \\ \left(\begin{array}{ccccccc} Q_{i,i-1,0}^{[C]} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Q_{i,i-1,1}^{[C]} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{i,i-1,2}^{[C]} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{i,i-1,C-i-1}^{[C]} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & Q_{i,i-1,C-i}^{[C]} & \mathbf{0} \end{array} \right) \end{array},$$

where

$$Q_{i,i-1,n}^{[C]} = \begin{cases} \begin{bmatrix} 0 & 0 \\ 0 & \underline{B}_{0,1}' \underline{\beta}_1 \end{bmatrix} & , \text{ if } n = 0, \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & (1-\delta_{\mathcal{I},-2}) \underline{B}_{0,1}' \underline{\beta}_1 & \delta_{\mathcal{I},-2} \underline{B}_{0,1}' \underline{\gamma}_{12} & \delta_{\mathcal{I},-2} \gamma_{12}^{[0]} \underline{B}_{0,1}' \underline{\beta}_2 \\ 0 & 0 & 0 & 0 \end{bmatrix} & , \text{ if } n = 1, 2, \ldots, C-i. \end{cases}$$

3.2. Sojourn time distribution

Next, we turn our attention to deriving the class-1 sojourn time distribution of a broken machine, representing the time between when a machine suffers a class-1 failure and when it is up and working again. To do so, we require the steady-state distribution of the system immediately prior to a class-1 customer arrival, which can be obtained as follows (e.g., see Lakatos et al. [14], Chapter 9):

$$\begin{aligned} q_{m,n,l,y} &= P\left((X_1,X_2,L,Y)=(m,n,l,y) \text{ immediately prior to a class-1 customer arrival}\right) \\ &= \frac{\text{steady-state class-1 arrival rate from state } (m,n,l,y)}{\text{steady-state class-1 customer arrival rate}} \\ &= \frac{(C-m-n)\alpha_1\pi_{m,n,l,y}}{\sum_{x_1}\sum_{x_2}\sum_{w}\sum_{z}(C-x_1-x_2)\alpha_1\pi_{x_1,x_2,w,z}} \\ &= \frac{(C-m-n)\pi_{m,n,l,y}}{\sum_{x_1}\sum_{x_2}\sum_{w}\sum_{z}(C-x_1-x_2)\pi_{x_1,x_2,w,z}} . \end{aligned} \tag{5}$$

Note that the right-hand side of Equation (5) equals zero for all l and y when $m+n=C$, as this corresponds to states where every machine has already suffered a failure (and so there are no working machines available to fail).

We must also consider the impact that the arrival may have on the server, should the arrival be to an empty class-1 queue. This distinction is important, since as we can see by contrasting the blocks $Q_{0,1}^{[C]}$ and $Q_{i,i+1}^{[C]}$, only an arrival to an empty queue may trigger the server to move (causing a change in L), as any additional arrivals to a non-empty queue simply increment X_1. For either possible service policy, if $L \in \{0,5\}$ (i.e., the server is idle or switching into the idle state), then the server will immediately begin a switch-in to serve the class-1 arrival. Let

$$q_{0,0,\bullet,\bullet} = q_{0,0,0,0} + \sum_{i=1}^{s_0} q_{0,0,5,i}$$

be the probability of the system being in any of these states immediately prior to the class-1 arrival. Furthermore, if $L=3$ (i.e., the server is conducting a class-2 switch-in), then the server will similarly initiate a switch to serve the class-1 arrival only when $\mathcal{I}=-1$. As such, let

$$q_{0,+,3,\bullet} = \delta_{\mathcal{I},-1}\sum_{n=1}^{C-1}\sum_{y=1}^{s_2} q_{0,n,3,y}$$

represent the desired probability that $L=3$ immediately before the class-1 arrival.

In order to construct the distribution of the waiting time (to reach service), we consider how long it takes for the queue in front of the target customer to empty, as well as the duration of time (if any) required for the server to switch to the target customer once at the head of their queue. Since we are considering an arrival to the system, the state of the process immediately prior to the arrival cannot possibly be one with $X_1+X_2=C$, as there would have had to be at least one machine working to fail. Thus, we construct initial probability vectors in the style of a queue featuring $C-1$ total machines.

We begin by considering the system with $X_1=0$ prior to the arrival. Let

$$\underline{q}_{0,n} = \left((1-\delta_{\mathcal{I},-1})q_{0,n,3,1},\ldots,(1-\delta_{\mathcal{I},-1})q_{0,n,3,s_2}, q_{0,n,4,1},\ldots,q_{0,n,4,b_2}\right) \tag{6}$$

be a row vector of length s_2+b_2 corresponding to the possible states when $X_1=0$ and $X_2=n$, $0 \le n \le C-1$. Since we have extracted the probability $q_{0,+,3,\bullet}$ when $\mathcal{I}=-1$, we must remove the probabilities of starting in the states where $L=3$ (in the class-1 non-preemptive priority case). When $X_1=X_2=0$, it follows that the only possible states immediately after the class-1 arrival correspond to a class-1 switch-in, which when finished (if not interrupted by a class-2 arrival, should $\mathcal{I}=-2$), leads to the completion of the waiting time. The initial probabilities for these states are contained in the row vector $q_{0,0,\bullet,\bullet}\underline{\gamma}_{01} + q_{0,+,3,\bullet}\underline{\gamma}_{21}$, which when combined with $\underline{q}_{0,n}$ in Equation (6), allow us to construct the full initial probability vector when $X_1 = 0$, namely

$$\underline{q}_0 = (q_{0,0,\bullet,\bullet}\underline{\gamma}_{01} + q_{0,+,3,\bullet}\underline{\gamma}_{21}, \underline{q}_{0,1}, \ldots, \underline{q}_{0,C-1}),$$

which has length $s_1 + (C-1)(s_2 + b_2)$.

When $X_1 = m \geq 1$ prior to the arrival, there is no shifting of probability mass required. We can simply construct $\underline{q}_m$ in a way which is analogous to how we originally defined $\underline{\pi}_m$ (although under the framework of a system with one less machine). Specifically, we have

$$\underline{q}_m = (\underline{q}_{m,0}, \underline{q}_{m,1}, \ldots, \underline{q}_{m,C-1-m}),$$

$$\underline{q}_{m,0} = (q_{m,0,1,1}, \ldots, q_{m,0,1,s_1}, q_{m,0,2,1}, \ldots, q_{m,0,2,b_1}),$$

$$\underline{q}_{m,n} = (q_{m,n,1,1}, \ldots, q_{m,n,1,s_1}, q_{m,n,2,1}, \ldots, q_{m,n,2,b_1}, q_{m,n,3,1}, \ldots, q_{m,n,3,s_2}, q_{m,n,4,1}, \ldots, q_{m,n,4,b_2}),$$

$$\underline{q} = (\underline{q}_{C-1}, \underline{q}_{C-2}, \ldots, \underline{q}_1, \underline{q}_0).$$

Note that $\underline{q}$ is a row vector of length

$$s_1 + (C-1)(s_2 + b_2) + \sum_{m=1}^{C-1}[s_1 + b_1 + (C-1-m)(s_1 + b_1 + s_2 + b_2)] = s_1 + \frac{C(C-1)}{2}(s_1 + b_1 + s_2 + b_2) \quad \text{and}$$

$\underline{q}\underline{e}' = 1 - q_{0,0,\bullet,\bullet}\gamma_{01}^{[0]} - q_{0,+,3,\bullet}\gamma_{21}^{[0]}$, where $q_{0,0,\bullet,\bullet}\gamma_{01}^{[0]} + q_{0,+,3,\bullet}\gamma_{21}^{[0]}$ is the probability that the machine immediately begins service after suffering a class-1 failure.

If we simply consider how the queue length ahead of the target class-1 customer changes, we can define, for a system with m total machines,

$$\tilde{Q}^{[m]} = \begin{array}{c|ccccccc|} & m & m-1 & m-2 & \cdots & 2 & 1 & 0 \\ m & Q_{m,m}^{[m]} & Q_{m,m-1}^{[m]} & 0 & \cdots & 0 & 0 & 0 \\ m-1 & 0 & Q_{m-1,m-1}^{[m]} & Q_{m-1,m-2}^{[m]} & \ddots & 0 & 0 & 0 \\ m-2 & 0 & 0 & Q_{m-2,m-2}^{[m]} & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 2 & 0 & 0 & 0 & \cdots & Q_{2,2}^{[m]} & Q_{2,1}^{[m]} & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 & Q_{1,1}^{[m]} & \tilde{Q}_{1,0}^{[m]} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \tilde{Q}_{0,0}^{[m]} \end{array},$$

which can serve as the rate matrix for a phase-type representation of the target customer's waiting time distribution, where the level of the process decreases until it is eventually absorbed out of level 1 or level 0. Note that we have retained the contributions from class-1 arrivals on the main diagonal terms of $Q_{i,i}^{[m]}$ and $\tilde{Q}_{0,0}^{[m]}$, as they are ultimately required for the final analysis. For the immediate discussion, however, we proceed as if these were not included, and hence would not cause non-zero row sums that would imply positive transition rates to absorption from unintended states. Moreover, the level of this rate matrix corresponds to the length of the queue in front of the target customer, which is clearly different than the total class-1 queue length. To adjust for this change relative to the original QBD process, and to the fact that the waiting time ends when the target customer is eligible to receive service, we make use of the modified blocks $\tilde{Q}_{1,0}^{[m]}$ and $\tilde{Q}_{0,0}^{[m]}$. Specifically,

$$\tilde{Q}_{0,0}^{[m]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ m-1 \\ m \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & \cdots & m-1 & m \end{array} \\ \left(\begin{array}{cccccc} -m\delta_{\mathcal{I},-2}\alpha I_{s_1}+S_1 & m\delta_{\mathcal{I},-2}\alpha_2\underline{e}'\left[\begin{array}{cc}\underline{\gamma}_{12} & \gamma_{12}^{[0]}\underline{\beta}_2\end{array}\right] & 0 & \cdots & 0 & 0 \\ \left[\begin{array}{c}0\\ \underline{B}'_{0,2}\underline{\gamma}_{21}\end{array}\right] & \Delta_1^{[m]} & (m-1)\alpha_2 I_{s_2+b_2} & \ddots & 0 & 0 \\ \delta_{\mathcal{I},-1}\left[\begin{array}{c}0\\ \underline{B}'_{0,2}\underline{\gamma}_{21}\end{array}\right] & (1-\delta_{\mathcal{I},-1})\Gamma & \Delta_2^{[m]} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \delta_{\mathcal{I},-1}\left[\begin{array}{c}0\\ \underline{B}'_{0,2}\underline{\gamma}_{21}\end{array}\right] & 0 & 0 & \cdots & \Delta_{m-1}^{[m]} & \alpha_2 I_{s_2+b_2} \\ \delta_{\mathcal{I},-1}\left[\begin{array}{c}0\\ \underline{B}'_{0,2}\underline{\gamma}_{21}\end{array}\right] & 0 & 0 & \cdots & (1-\delta_{\mathcal{I},-1})\Gamma & \Delta_m^{[m]} \end{array} \right) \end{array}$$

is structurally similar to $Q_{0,0}^{[m]}$, with the idle server state and class-0 switch-in states replaced with class-1 switch-in states which lead to absorption. Conditional on $\mathcal{I}=-1$, the transitions after a class-2 service completion are redirected towards these states. To achieve this, we multiply $(1-\delta_{\mathcal{I},-1})$ into Γ to remove those possible transitions, and redirect the system to $X_2=0$ with the transitions in column 0 of $\tilde{Q}_{0,0}^{[m]}$. If class 1 has non-preemptive priority, then the server will switch to serve class 1 after a service completion, and from the target class-1 customer's perspective, the class-2 queue length no longer matters. For this reason, we also multiply $\delta_{\mathcal{I},-2}$ into the failure rates of other machines, since once they have reached the front of their queue (and the server is switching to serve them), an arrival can only impact the target customer if it is a class-2 failure and class 2 has non-preemptive priority. This would result in the server leaving the target class-1 customer until the class-2 queue empties again. In addition, if the system would transition to these states following a class-2 service completion (with probability $\gamma_{21}^{[0]}$), then the process is directly absorbed without visiting the class-1 switch-in states.

Next, we have

$$\tilde{Q}_{1,0}^{[m]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ m-2 \\ m-1 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & m-2 & m-1 & m \end{array} \\ \left(\begin{array}{ccccccc} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \tilde{Q}_{1,0}^{\star} & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & \tilde{Q}_{1,0}^{\star} & \ddots & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \tilde{Q}_{1,0}^{\star} & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & \tilde{Q}_{1,0}^{\star} & 0 \end{array} \right) \end{array},$$

where

$$\tilde{Q}_{1,0}^{\star} = \delta_{\mathcal{I},-2}\left[\begin{array}{cc} 0 & 0 \\ \underline{B}'_{0,1}\underline{\gamma}_{12} & \underline{B}'_{0,1}\gamma_{12}^{[0]}\underline{\beta}_2 \\ 0 & 0 \end{array}\right] = \delta_{\mathcal{I},-2}Q_{1,0}^{\star}.$$

The definitions of $\tilde{Q}_{1,0}^{[m]}$ and $Q_{1,0}^{[m]}$ are almost identical, except that the block $\tilde{Q}_{1,0}^{[m]}$ leads the process to absorption automatically (instead of visiting level 0 of the process) when $X_2=0$ or when $X_2\geq 1$ and $\mathcal{I}\neq -2$, as there are no longer any customers ahead of the target customer and the server is already at the class-1 queue.

If the assumption that no class-1 customers could arrive behind the target customer held true, then we could claim that the waiting time is phase-type distributed with representation $\mathrm{PH}(\underline{q},\tilde{Q}^{[C-1]})$, as there are $C-1$ customers in the system which are not the target customer (and, in theory, could be queued ahead of it) and during this entire waiting time period, the target customer will never be at risk of failing again. However, this would obviously be an incorrect assumption to make since if a machine experiences a class-1 failure, while it does not add to the list of machines obtaining service ahead of the target customer, it does impact the rate of machines experiencing class-2 failures (due to the finite population assumption) which, depending on the service policy, may need to be serviced before the target customer. To address this issue, we propose the following rate matrix:

$$\mathcal{R} = \begin{array}{c|ccccccc} & C-1 & C-2 & C-3 & \cdots & 2 & 1 & 0 \\ \hline C-1 & \tilde{Q}^{[C-1]} & \tilde{Q}_{-}^{[C-1]} & 0 & \cdots & 0 & 0 & 0 \\ C-2 & 0 & \tilde{Q}^{[C-2]} & \tilde{Q}_{-}^{[C-2]} & \ddots & 0 & 0 & 0 \\ C-3 & 0 & 0 & \tilde{Q}^{[C-3]} & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 2 & 0 & 0 & 0 & \cdots & \tilde{Q}^{[2]} & \tilde{Q}_{-}^{[2]} & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 & \tilde{Q}^{[1]} & \tilde{Q}_{-}^{[1]} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \tilde{Q}^{[0]} \end{array},$$

where

$$\tilde{Q}_{-}^{[m]} = \begin{array}{c|cccccc} & m-1 & m-2 & \cdots & 2 & 1 & 0 \\ \hline m & 0 & 0 & \cdots & 0 & 0 & 0 \\ m-1 & Q_{m-1,m}^{[m]} & 0 & \ddots & 0 & 0 & 0 \\ m-2 & 0 & Q_{m-2,m-1}^{[m]} & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 2 & 0 & 0 & \cdots & Q_{2,3}^{[m]} & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & Q_{1,2}^{[m]} & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & \tilde{Q}_{0,1}^{[m]} \end{array},$$

$$\tilde{Q}_{0,1}^{[m]} = \begin{array}{c|ccccc} & 0 & 1 & 2 & \cdots & m-1 \\ \hline 0 & m\delta_{\mathcal{I},-2}\alpha_1 I_{s_1} & 0 & 0 & \cdots & 0 \\ 1 & 0 & \tilde{Q}_{0,1,1}^{[m]} & 0 & \ddots & 0 \\ 2 & 0 & 0 & \tilde{Q}_{0,1,2}^{[m]} & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ m-1 & 0 & 0 & 0 & \cdots & \tilde{Q}_{0,1,m-1}^{[m]} \\ m & 0 & 0 & 0 & \cdots & 0 \end{array},$$

and

$$\tilde{Q}_{0,1,j}^{[m]} = (m-j)\alpha_1 I_{s_2+b_2}.$$

Note that through the use of $\tilde{Q}_{-}^{[m]}$, the rate matrix $\mathcal{R}$ can reduce the system size by a single customer whenever a class-1 arrival would be observed. The blocks of $\tilde{Q}_{-}^{[m]}$ include the same $Q_{i,i+1}^{[m]}$ blocks defined previously, as well as a modified $\tilde{Q}_{0,1}^{[m]}$. When the queue length ahead of the target customer is zero, a class-1 arrival no longer increases the range of combinations of L and Y that the system must track from s_2+b_2 to $s_1+b_1+s_2+b_2$. Also, unless $\mathcal{I}=-2$, we do not consider the class-2 arrival rate when $X_2=0$. This follows since a class-1 switch-in time can only be interrupted if class 2 has non-preemptive priority.

To pair with the rate matrix $\mathcal{R}$, we define $\underline{\Phi}=(\underline{q},\underline{0},\underline{0},\ldots,\underline{0})$ to be the corresponding initial probability vector. The interpretation of $\underline{\Phi}$ is that the arrival of the target customer will always initiate the system in consideration of $C-1$ total other customers, which is only reduced further by future class-1 arrivals. As a result, the waiting time of our target class-1 customer is phase-type distributed with representation $\text{PH}(\underline{\Phi},\mathcal{R})$. Moreover, under exhaustive and non-preemptive priority service policies, a customer's service may not be interrupted, implying that the sojourn time is simply the sum of the waiting time and (independent) service time. Thus, it immediately follows that the class-1 sojourn time is phase-type distributed with representation $\text{PH}((\underline{\Phi},(q_{0,0,\bullet,\bullet}\gamma_{01}^{[0]}+q_{0,+,3,\bullet}\gamma_{21}^{[0]})\underline{\beta}_1),\mathcal{T})$, where

$$\mathcal{T}=\begin{bmatrix}\mathcal{R} & (-\mathcal{R}\underline{e}')\underline{\beta}_1\\ \mathbf{0} & B_1\end{bmatrix}.$$

Moments of the class-1 sojourn time distribution can easily be computed, as moments of a phase-type distribution are readily known (e.g., see He [10], p. 18). Finally, we remark that in order to obtain the corresponding sojourn time distribution for a machine that suffers a class-2 failure, one can simply switch all class-1 and class-2 parameters and distributions (the value of $\mathcal{I}$ will also need to be adjusted if the non-preemptive priority service policy is in place), recalculate the steady-state probabilities, and then repeat the above analysis.

4. Preemptive Resume Priority Models

We now turn our attention to the preemptive resume priority service policy. The primary way that preemptive resume priority differs from non-preemptive priority is that the arrival of a high priority customer to an empty queue (of its class) will trigger the server to begin a switch-in, independent of their current location. More precisely, the server is now able to interrupt the service of a low priority customer, whereas previously the server would only immediately change location (after observing an arrival) if they were idle or in the midst of a switch-in time. Eventually, once the high priority queue has been emptied again, the server resumes service with the interrupted customer in the low priority queue.

Unlike the previous section, whether class 1 or class 2 has preemptive resume priority will greatly impact the derivations needed to characterize the class-1 sojourn time distribution. As such, we consider each case separately in the following two subsections. In Section 4.1, we assume that class 1 has preemptive resume priority over class 2 and we determine the distribution of the time spent waiting and in service for a target class-1 customer. In Section 4.2, however, class 2 is assumed to have preemptive resume priority over class 1, and we seek to derive the sojourn time distribution of a target class-1 customer.

4.1. Case 1: $\mathcal{I}=1$

4.1.1. Steady-state probabilities

To model a system in which class 1 has preemptive resume priority, we keep track of five variables (X_1,X_2,L,Y,Y_2), where X_1, X_2, and L are as previously defined in Section 3. Moreover, Y denotes the phase of the service (if serving class 1) or switch-in time with possible values depending on L as follows:

$$Y\in\Omega_Y^{[1]}(L)=\begin{cases}\{0\} & \text{, if } L=0,\\ \{1,2\ldots,s_1\} & \text{, if } L=1,\\ \{1,2\ldots,b_1\} & \text{, if } L=2,\\ \{1,2\ldots,s_2\} & \text{, if } L=3,\\ \{0\} & \text{, if } L=4,\\ \{1,2\ldots,s_0\} & \text{, if } L=5.\end{cases}$$

The new variable Y_2 is intended to keep track of the phase of service of a preempted class-2 customer, taking on values (which depend on X_2) according to

$$Y_2 \in \Omega_{Y_2}^{[1]}(X_2) = \begin{cases} \{0\} & \text{, if } X_2 = 0, \\ \{1,2,\ldots,b_2\} & \text{, if } X_2 \geq 1. \end{cases}$$

Let $\pi_{m,n,l,y,y_2}^{[1]}$ be the steady-state probability that $X_1 = m$, $X_2 = n$, $L = l$, $Y = y$, and $Y_2 = y_2$, where $0 \leq X_1 \leq C$, $0 \leq X_2 \leq C - X_1$, and L, Y, and Y_2 take values from their respective supports above. Treating X_1 as the level of the process, define

$$\underline{\pi}_0^{[1]} = (\pi_{0,0,0,0,0}^{[1]}, \pi_{0,0,5,1,0}^{[1]}, \ldots, \pi_{0,0,5,s_0,0}^{[1]}, \underline{\pi}_{0,1}^{[1]}, \ldots, \underline{\pi}_{0,C}^{[1]})$$

to be the 0^{th} steady-state probability row vector, in which

$$\underline{\pi}_{0,n}^{[1]} = (\pi_{0,n,3,1,1}^{[1]}, \ldots, \pi_{0,n,3,1,b_2}^{[1]}, \pi_{0,n,3,2,1}^{[1]}, \ldots, \pi_{0,n,3,s_2,b_2}^{[1]}, \pi_{0,n,4,0,1}^{[1]}, \ldots, \pi_{0,n,4,0,b_2}^{[1]})$$

is a row vector of length $s_2 b_2 + b_2$ for $n = 1,2,\ldots,C$. Therefore, level 0 consists of $1 + s_0 + C(s_2 b_2 + b_2)$ states. For $m \geq 1$, the m^{th} steady-state probability row vector is

$$\underline{\pi}_m^{[1]} = (\underline{\pi}_{m,0}^{[1]}, \underline{\pi}_{m,1}^{[1]}, \ldots, \underline{\pi}_{m,C-m}^{[1]}),$$

where

$$\underline{\pi}_{m,0}^{[1]} = (\pi_{m,0,1,1,0}^{[1]}, \ldots, \pi_{m,0,1,s_1,0}^{[1]}, \pi_{m,0,2,1,0}^{[1]}, \ldots, \pi_{m,0,2,b_1,0}^{[1]})$$

and (for $n = 1,2,\ldots,C-m$)

$$\begin{aligned} \underline{\pi}_{m,n}^{[1]} = (&\pi_{m,n,1,1,1}^{[1]}, \ldots, \pi_{m,n,1,1,b_2}^{[1]}, \pi_{m,n,1,2,1}^{[1]}, \ldots, \pi_{m,n,1,s_1,b_2}^{[1]}, \\ &\pi_{m,n,2,1,1}^{[1]}, \ldots, \pi_{m,n,2,1,b_2}^{[1]}, \pi_{m,n,2,2,1}^{[1]}, \ldots, \pi_{m,n,2,b_1,b_2}^{[1]}) \end{aligned}$$

have respective lengths of $s_1 + b_1$ and $(s_1 + b_1)b_2$. Clearly, level m possesses $s_1 + b_1 + (C - m)(s_1 + b_1)b_2$ states for $m \geq 1$. Let $\underline{\pi}^{[1]} = (\underline{\pi}_0^{[1]}, \underline{\pi}_1^{[1]}, \ldots, \underline{\pi}_C^{[1]})$ be the steady-state probability row vector for the full process, which we can solve for using the QBD procedure described in Section 2. However, for notational convenience, let $Q^{[C,1]}$ now denote the corresponding infinitesimal generator for a system with C machines and class-1 preemptive priority, constructed in the manner of Equation (1), but with blocks denoted by $Q_{i,j}^{[C,1]}$ rather than $Q_{i,j}^{[C]}$. Letting $\otimes$ denote the Kronecker product operator, the diagonal blocks of $Q^{[C,1]}$ can be expressed as

$$Q_{0,0}^{[C,1]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-1 \\ C \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & \cdots & C-1 & C \end{array} \\ \left(\begin{array}{cccccc} \Delta_0^{[C,1]} & C\alpha_2 \underline{e}' \left[\begin{array}{cc} \underline{\gamma}_{02} \otimes \underline{\beta}_2 & \gamma_{02}^{[0]} \underline{\beta}_2 \end{array} \right] & 0 & \cdots & 0 & 0 \\ \left[\begin{array}{cc} \underline{0}' & 0 \\ \gamma_{20}^{[0]} \underline{B}'_{0,2} & \underline{B}'_{0,2} \underline{\gamma}_{20} \end{array} \right] & \Delta_1^{[C,1]} & (C-1)\alpha_2 I_{s_2 b_2 + b_2} & \ddots & 0 & 0 \\ 0 & \Gamma^{[1]} & \Delta_2^{[C,1]} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \Delta_{C-1}^{[C,1]} & \alpha_2 I_{s_2 b_2 + b_2} \\ 0 & 0 & 0 & \cdots & \Gamma^{[1]} & \Delta_C^{[C,1]} \end{array} \right) \end{array},$$

where

$$\Delta_n^{[C,1]} = \begin{cases} -C\alpha I_{1+s_0} + \left[\begin{array}{cc} 0 & \underline{0} \\ \underline{S}'_{0,0} & S_0 \end{array} \right] & \text{, if } n = 0, \\ -(C-n)\alpha I_{s_2 b_2 + b_2} + \left[\begin{array}{cc} S_2 \otimes I_{b_2} & \underline{S}'_{0,2} \otimes I_{b_2} \\ \mathbf{0} & B_2 \end{array} \right] & \text{, if } n = 1,2,\ldots,C, \end{cases}$$

and

$$\Gamma^{[1]} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{B}'_{0,2}\underline{\beta}_2 \end{bmatrix},$$

while for $i = 1, 2, \ldots, C$,

$$Q_{i,i}^{[C,1]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-i-1 \\ C-i \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & \cdots & C-i-1 & C-i \end{array} \\ \left(\begin{array}{cccccc} Q_{i,i,0}^{[C,1]} & (UD)_{i,0}^{[C,1]} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Q_{i,i,1}^{[C,1]} & (UD)_{i,1}^{[C,1]} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{i,i,2}^{[C,1]} & \ddots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{i,i,C-i-1}^{[C,1]} & (UD)_{i,C-i-1}^{[C,1]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & Q_{i,i,C-i}^{[C,1]} \end{array} \right) \end{array},$$

where

$$Q_{i,i,n}^{[C,1]} = \begin{cases} -(C-i)\alpha I_{s_1+b_1} + \begin{bmatrix} S_1 & \underline{S}'_{0,1}\underline{\beta}_1 \\ \mathbf{0} & B_1 \end{bmatrix} & \text{, if } n = 0, \\ \\ -(C-i-n)\alpha I_{(s_1+b_1)b_2} + \begin{bmatrix} S_1 & \underline{S}'_{0,1}\underline{\beta}_1 \\ \mathbf{0} & B_1 \end{bmatrix} \otimes I_{b_2} & \text{, if } n = 1, 2, \ldots, C-i, \end{cases}$$

and

$$(UD)_{i,n}^{[C,1]} = \begin{cases} (C-i)\alpha_2 I_{s_1+b_1} \otimes \underline{\beta}_2 & \text{, if } n = 0, \\ \\ (C-i-n)\alpha_2 I_{(s_1+b_1)b_2} & \text{, if } n = 1, 2, \ldots, C-i-1. \end{cases}$$

Moving to the off-diagonal blocks of $Q^{[C,1]}$, we first have

$$Q_{i,i+1}^{[C,1]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-i-1 \\ C-i \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & \cdots & C-i-1 \end{array} \\ \left(\begin{array}{ccccc} (C-i)\alpha_1 I_{s_1+b_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & Q_{i,i+1,1}^{[C,1]} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{i,i+1,2}^{[C,1]} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{i,i+1,C-i-1}^{[C,1]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{array} \right) \end{array}$$

for $i = 1, 2, \ldots, C-1$, where

$$Q_{i,i+1,n}^{[C,1]} = (C-i-n)\alpha_1 I_{(s_1+b_1)b_2},\ n = 1, 2, \ldots, C-i-1.$$

Furthermore,

$$Q_{0,1}^{[C,1]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-1 \\ C \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & \cdots & C-1 \end{array} \\ \left(\begin{array}{ccccc} C\alpha_1\underline{e}'\left[\begin{array}{cc} \underline{\gamma}_{01} & \gamma_{01}^{[0]}\underline{\beta}_1 \end{array} \right] & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & Q_{0,1,1}^{[C,1]} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{0,1,2}^{[C,1]} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{0,1,C-1}^{[C,1]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{array} \right) \end{array},$$

where

$$Q_{0,1,n}^{[C,1]} = (C-n)\alpha_1 \left[\begin{array}{cc} \underline{e}'\underline{\gamma}_{21} & \gamma_{21}^{[0]}\underline{e}'\underline{\beta}_1 \end{array} \right] \otimes I_{b_2},\ n = 1,2,\ldots,C-1,$$

and

$$Q_{1,0}^{[C,1]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-2 \\ C-1 \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & C-2 & C-1 & C \end{array} \\ \left(\begin{array}{ccccccc} \left[\begin{array}{cc} \underline{0}' & 0 \\ \gamma_{10}^{[0]}\underline{B}_{0,1}' & \underline{B}_{0,1}'\underline{\gamma}_{10} \end{array} \right] & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Q_{1,0}^{\star,[1]} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{1,0}^{\star,[1]} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{1,0}^{\star,[1]} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & Q_{1,0}^{\star,[1]} & \mathbf{0} \end{array} \right) \end{array},$$

where

$$Q_{1,0}^{\star,[1]} = \left[\begin{array}{cc} 0 & \underline{0}' \\ \underline{B}_{0,1}'\underline{\gamma}_{12} & \gamma_{12}^{[0]}\underline{B}_{0,1}' \end{array} \right] \otimes I_{b_2}.$$

Finally, for $i = 2,3,\ldots,C$, the remaining blocks of $Q^{[C,1]}$ are given by

$$Q_{i,i-1}^{[C,1]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-i-1 \\ C-i \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & C-i-1 & C-i & C-i+1 \end{array} \\ \left(\begin{array}{ccccccc} Q_{i,i-1,0}^{[C,1]} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & Q_{i,i-1,1}^{[C,1]} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{i,i-1,2}^{[C,1]} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{i,i-1,C-i-1}^{[C,1]} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & Q_{i,i-1,C-i}^{[C,1]} & \mathbf{0} \end{array} \right) \end{array},$$

where

$$Q_{i,i-1,n}^{[C,1]} = \begin{cases} \left[\begin{array}{cc} 0 & 0 \\ 0 & \underline{B}_{0,1}'\underline{\beta}_1 \end{array} \right] & ,\ \text{if } n = 0, \\ \left[\begin{array}{cc} 0 & 0 \\ 0 & \underline{B}_{0,1}'\underline{\beta}_1 \end{array} \right] \otimes I_{b_2} & ,\ \text{if } n = 1,2,\ldots,C-i. \end{cases}$$

4.1.2. Sojourn time distribution

When considering the time a machine spends offline after suffering a class-1 failure, we are again able to decompose the failed machine's sojourn time into its waiting time (to reach the server) and time in service, since a class-1 customer does not experience any service preemptions. We also note that unlike the exhaustive and non-preemptive priority service policies, when considering the class-1 waiting time in isolation, we do not need to track the class-2 queue at all. As a result, we can disregard all arrivals following the target class-1 customer and this greatly simplifies the subsequent analysis. We begin by modifying Equation (5) to determine the corresponding steady-state distribution of the system immediately prior to a class-1 customer arrival:

$$\begin{aligned} q^{[1]}_{m,n,l,y,y_2} &= P\left((X_1,X_2,L,Y,Y_2)=(m,n,l,y,y_2) \text{ immediately prior to a class-1 customer arrival}\right) \\ &= \frac{(C-m-n)\pi^{[1]}_{m,n,l,y,y_2}}{\sum_{x_1}\sum_{x_2}\sum_{w}\sum_{z}\sum_{z_2}(C-x_1-x_2)\pi^{[1]}_{x_1,x_2,w,z,z_2}}. \end{aligned} \tag{7}$$

As before, the right-hand side of Equation (7) is equal to zero for all l, y, and y_2 when $m+n=C$.

If a class-1 customer arrives to a non-empty queue, then this arrival does not affect the server and the waiting time is simply the time it takes to empty the queue of class-1 customers in front of this new arrival. On the other hand, if the target customer arrives to find an empty class-1 queue, then the corresponding waiting time is simply equal to the switch-in time, which will lead to an initial probability vector dependent on whether $X_2=0$ or not. Let

$$q^{[1]}_{0,0,\bullet,\bullet,\bullet} = q^{[1]}_{0,0,0,0,0} + \sum_{i=1}^{s_0} q^{[1]}_{0,0,5,i,0}$$

be the probability that the server is either idle or conducting a switch-in to the idle state immediately before the target class-1 customer arrives (since both queues were empty). Furthermore, let

$$q^{[1]}_{0,+,\bullet,\bullet,\bullet} = \sum_{n=1}^{C-1}\sum_{l=3}^{4}\sum_{y\in\Omega^{[1]}_{Y}(l)}\sum_{y_2\in\Omega^{[1]}_{Y_2}(n)} q^{[1]}_{0,n,l,y,y_2}$$

be the probability that the target customer arrives to an empty class-1 queue, while $X_2 \geq 1$. We separate these two events, despite both yielding a waiting time that only consists of a class-1 switch-in, because the initial probability vector may be different in either case. Taking this into consideration, we may now construct the initial probability vectors for the waiting time distribution. Letting the level of the process equal the number of class-1 customers ahead of the target customer, the initial probability vector corresponding to level 0 is given by

$$\underline{q}^{[1]}_{0} = q^{[1]}_{0,0,\bullet,\bullet,\bullet}\underline{\gamma}_{01} + q^{[1]}_{0,+,\bullet,\bullet,\bullet}\underline{\gamma}_{21},$$

which, as we can see, simply initializes the switch-in time, which has a phase-type distribution. For non-zero levels, it is possible for the server to be in the midst of a class-1 switch-in or service time. For each combination of m, l, and $y\in\{1,2\}$, we obtain the desired marginal distributions by summing the probability mass that was spread out over different states that were used to track X_2 or Y_2, namely

$$q^{[1]}_{m,\bullet,l,y,\bullet} = q^{[1]}_{m,0,l,y,0} + \sum_{n=1}^{C-m-1}\sum_{y_2\in\Omega^{[1]}_{Y_2}(n)} q^{[1]}_{m,n,l,y,y_2}. \tag{8}$$

Equation (8) may then be used to construct the initial probability vector corresponding to level m, $1\leq m\leq C-1$:

$$\underline{q}^{[1]}_{m,\bullet} = (q^{[1]}_{m,\bullet,1,1,\bullet},\ldots,q^{[1]}_{m,\bullet,1,s_1,\bullet},q^{[1]}_{m,\bullet,2,1,\bullet},\ldots,q^{[1]}_{m,\bullet,2,b_1,\bullet}).$$

These vectors may then be collected, including the probability vector for level 0, to construct the full initial probability vector for the process:

$$\underline{q}^{[1]} = (\underline{q}^{[1]}_{C-1,\bullet}, \underline{q}^{[1]}_{C-2,\bullet}, \dots, \underline{q}^{[1]}_{1,\bullet}, \underline{q}^{[1]}_{0}).$$

Note that $\underline{q}^{[1]}$ is a row vector of length $s_1 + (C-1)(s_1 + b_1)$ and $\underline{q}^{[1]}\underline{e}' = 1 - q^{[1]}_{0,0,\bullet,\bullet,\bullet}\gamma^{[0]}_{01} - q^{[1]}_{0,+,\bullet,\bullet,\bullet}\gamma^{[0]}_{21}$, where $q^{[1]}_{0,0,\bullet,\bullet,\bullet}\gamma^{[0]}_{01} + q^{[1]}_{0,+,\bullet,\bullet,\bullet}\gamma^{[0]}_{21}$ is the probability that the machine immediately begins service after suffering a class-1 failure.

We next focus on designing a rate matrix corresponding to this waiting time for a system that may have up to m customers waiting in front of the target class-1 customer. This ultimately results in

$$\tilde{Q}^{[m,1]} = \begin{array}{c} \\ m \\ m-1 \\ m-2 \\ \vdots \\ 2 \\ 1 \\ 0 \end{array} \begin{array}{c} \begin{array}{ccccccc} m & m-1 & m-2 & \cdots & 2 & 1 & 0 \end{array} \\ \left(\begin{array}{ccccccc} \tilde{Q}^{[m,1]}_{m,m} & \tilde{Q}^{[m,1]}_{m,m-1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{Q}^{[m,1]}_{m-1,m-1} & \tilde{Q}^{[m,1]}_{m-1,m-2} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{Q}^{[m,1]}_{m-2,m-2} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \tilde{Q}^{[m,1]}_{2,2} & \tilde{Q}^{[m,1]}_{2,1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \tilde{Q}^{[m,1]}_{1,1} & \tilde{Q}^{[m,1]}_{1,0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \tilde{Q}^{[m,1]}_{0,0} \end{array} \right) \end{array},$$

where

$$\tilde{Q}^{[m,1]}_{0,0} = S_1$$

is the class-1 switch-in time rate matrix,

$$\tilde{Q}^{[m,1]}_{1,0} = \mathbf{0}$$

is a zero matrix on account of the service completion of the lone customer queueing ahead of the target customer leading to absorption,

$$\tilde{Q}^{[m,1]}_{i,i} = \begin{bmatrix} S_1 & \underline{S}_{0,1}{}'\underline{\beta}_1 \\ \mathbf{0} & B_1 \end{bmatrix},\ i = 1, 2, \dots, m,$$

can track the s_1 switch-in time phases, of which a completion leads to the start of a class-1 service, and

$$\tilde{Q}^{[m,1]}_{i,i-1} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{B}_{0,1}{}'\underline{\beta}_1 \end{bmatrix},\ i = 2, 3, \dots, m,$$

since a class-1 service completion leads directly into the start of another class-1 service. As previously stated, we do not need to consider any arrivals following that of the target customer, since they do not impact the waiting time. Therefore, the rate matrix corresponding to the waiting time for a class-1 customer in a system with C total customers is simply $\mathcal{R}^{[1]} = \tilde{Q}^{[C-1,1]}$, and it subsequently follows that the waiting time of our target class-1 customer is phase-type distributed with representation $\text{PH}(\underline{q}^{[1]}, \mathcal{R}^{[1]})$. Finally, the class-1 sojourn time distribution of a broken machine, consisting of its waiting time plus an independent service time, can readily be represented as $\text{PH}((\underline{q}^{[1]}, (q^{[1]}_{0,0,\bullet,\bullet,\bullet}\gamma^{[0]}_{01} + q_{0,+,\bullet,\bullet,\bullet}\gamma^{[0]}_{21})\underline{\beta}_1), \mathcal{T}^{[1]})$, where

$$\mathcal{T}^{[1]} = \begin{bmatrix} \mathcal{R}^{[1]} & (-\mathcal{R}^{[1]}\underline{e}')\underline{\beta}_1 \\ \mathbf{0} & B_1 \end{bmatrix}.$$

4.2. Case 2: $\mathcal{I} = 2$

4.2.1. Steady-state probabilities

We now consider the situation in which class 2 has preemptive resume priority over class 1. We remark that while we can use the results for $\mathcal{I} = 1$, by swapping the relevant parameters and distributions, to solve for the steady-state probabilities of the process, as well as the time until repair for a machine that suffers a class-2 failure, we would be unable to characterize the sojourn time distribution for a class-1 failed machine. As such, the purpose of this subsection is to act as a compliment to the analysis of the previous subsection, so that the sojourn time distribution for the lower priority class of machine failures may be found when the server is employing a preemptive resume priority service policy.

First of all, the construction of the infinitesimal generator will involve many of the same techniques used previously, however this time tracking the service phase of the next class-1 customer in line (if any). Moreover, due to the preemptive priority of class-2 customers, the process does not need to consider states where the server is conducting a class-1 switch-in or service time whenever there are class-2 customers in the system. Thus, we track the variables (X_1, X_2, L, Y, Y_1), where X_1, X_2, and L are as previously defined, while Y denotes the phase of the service (if serving class 2) or switch-in time with values depending on L in the following way:

$$Y \in \Omega_Y^{[2]}(L) = \begin{cases} \{0\} & \text{, if } L = 0, \\ \{1, 2, \ldots, s_1\} & \text{, if } L = 1, \\ \{0\} & \text{, if } L = 2, \\ \{1, 2, \ldots, s_2\} & \text{, if } L = 3, \\ \{1, 2, \ldots, b_2\} & \text{, if } L = 4, \\ \{1, 2, \ldots, s_0\} & \text{, if } L = 5. \end{cases}$$

The variable Y_1 is used to track the phase of service of a class-1 customer and is determined at the arrival instant of a class-1 customer to an empty queue, as well as upon a service completion of a class-1 customer that segues into the next class-1 service time. Thus, the possible values of Y_1 are

$$Y_1 \in \Omega_{Y_1}^{[2]}(X_1) = \begin{cases} \{0\} & \text{, if } X_1 = 0, \\ \{1, 2, \ldots, b_1\} & \text{, if } X_1 \geq 1. \end{cases}$$

We define $\pi^{[2]}_{m,n,l,y,y_1}$ to be the steady-state probability that $X_1 = m$, $X_2 = n$, $L = l$, $Y = y$, and $Y_1 = y_1$, where $0 \leq X_1 \leq C$, $0 \leq X_2 \leq C - X_1$, and L, Y, and Y_1 take values from their supports above. Corresponding to the 0^{th} level of the process, let

$$\underline{\pi}_0^{[2]} = (\pi^{[2]}_{0,0,0,0,0}, \pi^{[2]}_{0,0,5,1,0}, \ldots, \pi^{[2]}_{0,0,5,s_0,0}, \underline{\pi}^{[2]}_{0,1}, \ldots, \underline{\pi}^{[2]}_{0,C}),$$

where

$$\underline{\pi}^{[2]}_{0,n} = (\pi^{[2]}_{0,n,3,1,0}, \ldots, \pi^{[2]}_{0,n,3,s_2,0}, \pi^{[2]}_{0,n,4,1,0}, \ldots, \pi^{[2]}_{0,n,4,b_2,0})$$

is a row vector of length $s_2 + b_2$ for $n = 1, 2, \ldots, C$, so that level 0 has $1 + s_0 + C(s_2 + b_2)$ states. For level $m \geq 1$, we define

$$\underline{\pi}_m^{[2]} = (\underline{\pi}^{[2]}_{m,0}, \underline{\pi}^{[2]}_{m,1}, \ldots, \underline{\pi}^{[2]}_{m,C-m}),$$

where

$$\underline{\pi}_{m,0}^{[2]} = (\pi_{m,0,1,1,1}^{[2]}, \ldots, \pi_{m,0,1,1,b_1}^{[2]}, \pi_{m,0,1,2,1}^{[2]}, \ldots, \pi_{m,0,1,s_1,b_1}^{[2]}, \pi_{m,0,2,0,1}^{[2]}, \ldots, \pi_{m,0,2,0,b_1}^{[2]})$$

and (for $n = 1, 2, \ldots, C-m$)

$$\begin{aligned}\underline{\pi}_{m,n}^{[2]} = (&\pi_{m,n,3,1,1}^{[2]}, \ldots, \pi_{m,n,3,1,b_1}^{[2]}, \pi_{m,n,3,2,1}^{[2]}, \ldots, \pi_{m,n,3,s_2,b_1}^{[2]},\\ &\pi_{m,n,4,1,1}^{[2]}, \ldots, \pi_{m,n,4,1,b_1}^{[2]}, \pi_{m,n,4,2,1}^{[2]}, \ldots, \pi_{m,n,4,b_2,b_1}^{[2]})\end{aligned}$$

are row vectors of length $s_1b_1 + b_1$ and $(s_2 + b_2)b_1$, respectively. In keeping with the same notational convention we adopted in the previous subsection, we denote the steady-state probability vector for the overall process by $\underline{\pi}^{[2]} = (\underline{\pi}_0^{[2]}, \underline{\pi}_1^{[2]}, \ldots, \underline{\pi}_C^{[2]})$, which may be obtained via the QBD procedure outlined in Section 2 (in which $Q^{[C,2]}$ denotes the infinitesimal generator for a system with C machines and class-2 preemptive resume priority, structured in the style of Equation (1), but with blocks $Q_{i,j}^{[C,2]}$). When considering the blocks of $Q^{[C,2]}$, we first remark that $Q_{0,0}^{[C,2]}$ is actually identical to $Q_{0,0}^{[C]}$ from the exhaustive and non-preemptive priority service models. This is because unlike when $\mathcal{I} = 1$, we must now track phases of class-1 service with our fifth state variable Y_1, not class-2 service phases. Since $X_1 = 0$ in this block, there are no class-1 service phases to keep track of (i.e., $Y_1 = 0$ for all states within this block), and the state space of the level 0 block reduces to that of the aforementioned service models. For $i = 1, 2, \ldots, C$, the other diagonal blocks can be expressed as

$$Q_{i,i}^{[C,2]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-i-1 \\ C-i \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & \cdots & C-i-1 & C-i \end{array} \\ \left(\begin{array}{cccccc} Q_{i,i,0}^{[C,2]} & (UD)_{i,0}^{[C,2]} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ (LD)_{i,1}^{[C,2]} & Q_{i,i,1}^{[C,2]} & (UD)_{i,1}^{[C,2]} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (LD)_{i,2}^{[C,2]} & Q_{i,i,2}^{[C,2]} & \ddots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{i,i,C-i-1}^{[C,2]} & (UD)_{i,C-i-1}^{[C,2]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & (LD)_{i,C-i}^{[C,2]} & Q_{i,i,C-i}^{[C,2]} \end{array} \right) \end{array},$$

where

$$Q_{i,i,n}^{[C,2]} = \begin{cases} -(C-i)\alpha I_{s_1b_1+b_1} + \begin{bmatrix} S_1 \otimes I_{b_1} & \underline{S}_{0,1}' \otimes I_{b_1} \\ \mathbf{0} & B_1 \end{bmatrix} & \text{, if } n = 0, \\ \\ -(C-i-n)\alpha I_{(s_2+b_2)b_1} + \begin{bmatrix} S_2 & \underline{S}_{0,2}'\underline{\beta}_2 \\ \mathbf{0} & B_2 \end{bmatrix} \otimes I_{b_1} & \text{, if } n = 1, 2, \ldots, C-i, \end{cases}$$

$$(UD)_{i,n}^{[C,2]} = \begin{cases} (C-i)\alpha_2 \begin{bmatrix} \underline{e}'\underline{\gamma}_{12} & \gamma_{12}^{[0]}\underline{e}'\underline{\beta}_2 \end{bmatrix} \otimes I_{b_1} & \text{, if } n = 0, \\ \\ (C-i-n)\alpha_2 I_{(s_2+b_2)b_1} & \text{, if } n = 1, 2, \ldots, C-i-1, \end{cases}$$

and

$$(LD)_{i,n}^{[C,2]} = \begin{cases} \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \underline{B}'_{0,2}\underline{\gamma}_{21} \otimes I_{b_1} & \gamma_{21}^{[0]}\underline{B}'_{0,2} \otimes I_{b_1} \end{bmatrix} & \text{, if } n=1, \\ \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \underline{B}'_{0,2}\underline{\beta}_2 \end{bmatrix} \otimes I_{b_1} & \text{, if } n=2,3,\ldots,C-i. \end{cases}$$

As for the off-diagonal blocks, we first have

$$Q_{i,i+1}^{[C,2]} = \begin{array}{cc} & \begin{array}{ccccc} 0 & 1 & 2 & \cdots & C-i-1 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ \vdots \\ C-i-1 \\ C-i \end{array} & \begin{pmatrix} (C-i)\alpha_1 I_{s_1 b_1 + b_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & Q_{i,i+1,1}^{[C,2]} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{i,i+1,2}^{[C,2]} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{i,i+1,C-i-1}^{[C,2]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} \end{array}$$

for $i=1,2,\ldots,C-1$, where

$$Q_{i,i+1,n}^{[C,2]} = (C-i-n)\alpha_1 I_{(s_2+b_2)b_1},\ n=1,2,\ldots,C-i-1.$$

In addition,

$$Q_{0,1}^{[C,2]} = \begin{array}{cc} & \begin{array}{ccccc} 0 & 1 & 2 & \cdots & C-1 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ \vdots \\ C-1 \\ C \end{array} & \begin{pmatrix} C\alpha_1\underline{e}'\begin{bmatrix} \underline{\gamma}_{01} \otimes \underline{\beta}_1 & \gamma_{01}^{[0]}\underline{\beta}_1 \end{bmatrix} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & Q_{0,1,1}^{[C,2]} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & Q_{0,1,2}^{[C,2]} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & Q_{0,1,C-1}^{[C,2]} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} \end{array},$$

where

$$Q_{0,1,n}^{[C,2]} = (C-n)\alpha_1 I_{s_2+b_2} \otimes \underline{\beta}_1,\ n=1,2,\ldots,C-1,$$

and

$$Q_{1,0}^{[C,2]} = \begin{array}{cc} & \begin{array}{cccccccc} 0 & 1 & 2 & \cdots & C-2 & C-1 & C \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ \vdots \\ C-2 \\ C-1 \end{array} & \begin{pmatrix} \begin{bmatrix} \mathbf{0}' & \mathbf{0} \\ \gamma_{10}^{[0]}\underline{B}'_{0,1} & \underline{B}'_{0,1}\underline{\gamma}_{10} \end{bmatrix} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \end{array}.$$

Finally, for $i=2,3,\ldots,C$, the remaining blocks of $Q^{[C,2]}$ are given by

$$Q_{i,i-1}^{[C,2]} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \\ C-i-1 \\ C-i \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & C-i-1 & C-i & C-i+1 \end{array} \\ \left(\begin{array}{ccccccc} \begin{bmatrix} 0 & 0 \\ 0 & \underline{B}'_{0,1}\underline{\beta}_1 \end{bmatrix} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{array}\right) \end{array}.$$

4.2.2. Sojourn time distribution

When considering the time between when a machine suffers a class-1 failure and when it is up and working again in this particular model, we come to realize that, for the first time, we are unable to uncouple the time spent waiting from the time spent in service. This is due to the unique situation that this service policy presents, in that the target customer's service time can potentially be interrupted due to the arrival of a high priority customer. Therefore, instead of only being concerned about the queue in front of the target class-1 customer emptying, we will model the total time it takes for those in front of the target customer, and the target customer itself, to complete service and leave the system.

Analogous to Equation (7), we find that the steady-state probabilities of the system immediately prior to a class-1 arrival can be obtained via

$$\begin{aligned} q_{m,n,l,y,y_1}^{[2]} &= P\left((X_1, X_2, L, Y, Y_1) = (m,n,l,y,y_1) \text{ immediately prior to a class-1 customer arrival}\right) \\ &= \frac{(C-m-n)\pi_{m,n,l,y,y_1}^{[2]}}{\sum_{x_1}\sum_{x_2}\sum_{w}\sum_{z}\sum_{z_1}(C-x_1-x_2)\pi_{x_1,x_2,w,z,z_1}^{[2]}}, \end{aligned}$$

which also yields a value of zero for all l, y, and y_1 when $m+n=C$. In anticipation of constructing the various probability vectors involved in characterizing the class-1 sojourn time distribution, we first define

$$q_{0,0,\bullet,\bullet,\bullet}^{[2]} = q_{0,0,0,0,0}^{[2]} + \sum_{i=1}^{s_0} q_{0,0,5,i,0}^{[2]} \tag{9}$$

to be the probability that a class-1 arrival finds the server idle or switching into the idle state. In addition, we group the other class-1 arrival instant probabilities into the following row vectors:

$$\begin{aligned} \underline{q}_{0,n}^{[2]} &= (q_{0,n,3,1,0}^{[2]},\ldots,q_{0,n,3,s_2,0}^{[2]}, q_{0,n,4,1,0}^{[2]},\ldots,q_{0,n,4,b_2,0}^{[2]}), \\ \underline{q}_{m,0}^{[2]} &= (q_{m,0,1,1,1}^{[2]},\ldots,q_{m,0,1,1,b_1}^{[2]}, q_{m,0,1,2,1}^{[2]},\ldots,q_{m,0,1,s_1,b_1}^{[2]}, q_{m,0,2,0,1}^{[2]},\ldots,q_{m,0,2,0,b_1}^{[2]}), \\ \underline{q}_{m,n}^{[2]} &= (q_{m,n,3,1,1}^{[2]},\ldots,q_{m,n,3,1,b_1}^{[2]}, q_{m,n,3,2,1}^{[2]},\ldots,q_{m,n,3,s_2,b_1}^{[2]}, \\ &\quad\; q_{m,n,4,1,1}^{[2]},\ldots,q_{m,n,4,1,b_1}^{[2]}, q_{m,n,4,2,1}^{[2]},\ldots,q_{m,n,4,b_2,b_1}^{[2]}). \end{aligned}$$

We note that if the target class-1 customer does not arrive to find an empty class-1 queue, then this arrival has no impact on any of the variables other than X_1. Therefore, letting $\underline{p}_{m,n}$ contain the ordered initial probability masses for states where $X_1 = m$ and $X_2 = n$, we have

$$\underline{p}_{m+1,n} = \underline{q}_{m,n}^{[2]},\ m = 1,2,\ldots,C-1.$$

However, if the target class-1 customer does arrive to find no other class-1 customers present (but with $X_2 \geq 1$), the characterization is not as straightforward. Even though the server will not be prompted to move, the first arrival of a class-1 customer requires that the system now track their eventual service phase. Therefore, we let

$$\underline{p}_{1,n} = \underline{q}_{0,n}^{[2]} \otimes \underline{\beta}_1,\ n \geq 1.$$

The last possibility for the arriving target customer involves finding the system empty of customers of either class requiring service, which occurs with probability $q_{0,0,\bullet,\bullet,\bullet}^{[2]}$ given by Equation (9). This sees the server begin either a class-1 switch-in time (while the system determines the initial service phase of the target customer), or an immediate class-1 service with probability $\gamma_{0,1}^{[0]}$. Therefore, we define

$$\underline{p}_{1,0} = (q_{0,0,\bullet,\bullet,\bullet}^{[2]} \underline{\gamma}_{01} \otimes \underline{\beta}_1, q_{0,0,\bullet,\bullet,\bullet}^{[2]} \gamma_{01}^{[0]} \underline{\beta}_1).$$

With these pieces in place, we can now define the initial probability vector for the m^{th} level, $m \geq 1$, as $\underline{p}_m = (\underline{p}_{m,0}, \underline{p}_{m,1}, \ldots, \underline{p}_{m,C-m})$, from which we can construct the overall initial probability vector

$$\underline{p} = (\underline{p}_C, \underline{p}_{C-1}, \ldots, \underline{p}_1).$$

We note that the levels of this modified process span from 1 to C. This is a result of the actual system immediately prior to the arrival requiring $0 \leq X_1 \leq C-1$ in order for a class-1 arrival to be observed, and due to the inclusion of the target customer, the level is incremented by 1. We have no interest in a 0^{th} level, since the emptying of the class-1 queue signifies the departure of the target customer, and as we will see below, leads to absorption in a particular continuous-time Markov chain. Incidentally, the row vector $\underline{p}$ has length

$$\sum_{m=1}^{C} [s_1 b_1 + b_1 + (C-m)(s_2 + b_2) s_1] = C(s_1 b_1 + b_1) + \frac{C(C-1)}{2}(s_2 + b_2) s_1$$

and satisfies $\underline{p}\underline{e}' = 1$ (since sojourn times are certain to be positive).

As was the case for the exhaustive and non-preemptive priority service models, we must consider future class-1 arrivals behind the target class-1 customer since they will affect the future arrival rates of class-2 customers, who must all finish service before any class-1 customers may be served. For a model with m total machines, in which there were no class-1 arrivals after the target customer, we would simply have the rate matrix

$$\tilde{Q}^{[m,2]} = \begin{array}{c} \\ m \\ m-1 \\ m-2 \\ \vdots \\ 2 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccccc} m & m-1 & m-2 & \cdots & 2 & 1 \end{array} \\ \begin{pmatrix} Q_{m,m}^{[m,2]} & Q_{m,m-1}^{[m,2]} & 0 & \cdots & 0 & 0 \\ 0 & Q_{m-1,m-1}^{[m,2]} & Q_{m-1,m-2}^{[m,2]} & \ddots & 0 & 0 \\ 0 & 0 & Q_{m-2,m-2}^{[m,2]} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & Q_{2,2}^{[m,2]} & Q_{2,1}^{[m,2]} \\ 0 & 0 & 0 & \cdots & 0 & Q_{1,1}^{[m,2]} \end{pmatrix} \end{array}.$$

In this case, the process is absorbed with rates equal to the service completion rates from $Q_{1,0}^{[m,2]}$ when residing in class-1 service states in level 1. This, of course, cannot accurately describe the entire process. We gather the blocks of $Q^{[m,2]}$ which contain transition rates corresponding to increments of X_1 and construct

$$\tilde{Q}_-^{[m,2]} = \begin{array}{c} \\ m \\ m-1 \\ m-2 \\ \vdots \\ 3 \\ 2 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccccc} m-1 & m-2 & m-3 & \cdots & 2 & 1 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ Q_{m-1,m}^{[m,2]} & 0 & 0 & \ddots & 0 & 0 \\ 0 & Q_{m-2,m-1}^{[m,2]} & 0 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & Q_{2,3}^{[m,2]} & 0 \\ 0 & 0 & 0 & \cdots & 0 & Q_{1,2}^{[m,2]} \end{pmatrix} \end{array} .$$

Together, these matrices allow us to fully describe the process via the rate matrix

$$\mathcal{R}^{[2]} = \begin{array}{c} \\ C \\ C-1 \\ C-2 \\ \vdots \\ 2 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccccc} C & C-1 & C-2 & \cdots & 2 & 1 \end{array} \\ \begin{pmatrix} \tilde{Q}^{[C,2]} & \tilde{Q}_-^{[C,2]} & 0 & \cdots & 0 & 0 \\ 0 & \tilde{Q}^{[C-1,2]} & \tilde{Q}_-^{[C-1,2]} & \ddots & 0 & 0 \\ 0 & 0 & \tilde{Q}^{[C-2,2]} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \tilde{Q}^{[2,2]} & \tilde{Q}_-^{[2,2]} \\ 0 & 0 & 0 & \cdots & 0 & \tilde{Q}^{[1,2]} \end{pmatrix} \end{array} ,$$

in combination with the (further) modified initial probability vector $\underline{\Phi}^{[2]} = (\underline{p}, \underline{0}, \underline{0}, \ldots, \underline{0})$, constructed as such since the system will always start in consideration of the full inventory of machines. In conclusion, we deduce that the class-1 sojourn time distribution of a broken machine can be represented as $\mathrm{PH}(\underline{\Phi}^{[2]}, \mathcal{R}^{[2]})$.

5. Numerical Examples

In this section, we investigate the effect that switch-in times have on the optimality of the different service policies, and the sensitivity of the mean number of working machines on various factors, including the total number of machines as well as the choice of phase-type service time distributions. Let the mean sojourn time $\mathrm{E}[S]$ of a target machine be defined as the weighted average of the mean sojourn times of machines that suffered a class-1 or class-2 failure (with weights equal to the probability of a random failure being from either class, which we select to be 0.9 and 0.1, respectively). Let N_W denote the number of working machines. It immediately follows that

$$\mathrm{E}[N_W] = \begin{cases} C - \sum_m \sum_n \sum_l \sum_y (m+n)\pi_{m,n,l,y} & \text{, if } \mathcal{I} \in \{-2,-1,0\}, \\ C - \sum_m \sum_n \sum_l \sum_y \sum_{y_2} (m+n)\pi_{m,n,l,y,y_2}^{[1]} & \text{, if } \mathcal{I} = 1, \\ C - \sum_m \sum_n \sum_l \sum_y \sum_{y_1} (m+n)\pi_{m,n,l,y,y_1}^{[2]} & \text{, if } \mathcal{I} = 2. \end{cases}$$

In order to gain efficiency from the priority service policies, we assume that the stratification of jobs into two classes is done in a logical manner such that "small" jobs and "large" jobs are not grouped together. Without loss of generality, we select class 1 to hold the small jobs. The biggest disadvantage to using priority service policies is that they result in more frequent switching between queues by the server. When these switches require non-insignificant amounts of time to complete, the additional time spent not serving customers may reduce the overall system efficiency. Therefore, we begin by considering the effect of $p_{>0} = 1 - \gamma_{ji}^{[0]}$, the probability of a switch-in time from queue j to queue i being non-zero.

For now, we set $C=10$ and $\alpha=0.075$ (hence, $\alpha_1=0.0675$ and $\alpha_2=0.0075$). Let the corresponding initial probability vectors and rate matrices for the phase-type switch-in time distributions be given by

$$\begin{aligned} \underline{\gamma}_{10} &= (p_{>0},0), & \underline{\gamma}_{20} &= (0,p_{>0}), \\ \underline{\gamma}_{01} &= (0,p_{>0},0), & \underline{\gamma}_{21} &= (p_{>0},0,0), \\ \underline{\gamma}_{02} &= (0,p_{>0},0), & \underline{\gamma}_{12} &= (p_{>0},0,0), \end{aligned}$$

and

$$S_0=\frac{1}{M_S}\begin{pmatrix}-2 & 0\\ 0 & -1\end{pmatrix},\ S_1=\frac{1}{M_S}\begin{pmatrix}-1 & 1 & 0\\ 0 & -2 & 2\\ 0 & 0 & -2\end{pmatrix},\ S_2=\frac{1}{M_S}\begin{pmatrix}-2 & 2 & 0\\ 0 & -1 & 1\\ 0 & 0 & -1\end{pmatrix},$$

where M_S is a constant that allows us to scale the expected switch-in times. We may interpret the above as class-dependent Erlang-2 (E_2) set-up and exponential take-down times, with both being faster for class 1. If the server moves to class 0 instead of the opposite queue (due to it being empty), they may complete the take-down for their previous queue and only require a set-up following the next arrival.

For the service time distributions, we consider hyperexponential-2 (H_2), with initial probability vectors and rate matrices given by

$$\underline{\beta}_1=\underline{\beta}_2=(0.9,0.1),\ B_1=2\begin{pmatrix}-1 & 0\\ 0 & -\frac{1}{11}\end{pmatrix},\ \text{and}\ B_2=\frac{1}{10M_B}\begin{pmatrix}-1 & 0\\ 0 & -\frac{1}{11}\end{pmatrix},$$

where, in a similar fashion, M_B is a constant for scaling the mean class-2 service time. The mean class-1 service time is set equal to 1, whereas the mean class-2 service time is set equal to 20 (when $M_B=1$). Through the use of these distributions, we are, in effect, considering the mixtures of two exponential distributions, representing the grouping of more than one type of failure within each class.

Figure 2 contains plots of both $\mathrm{E}[S]$ and $\mathrm{E}[N_W]$ under the above set of parameters (including $M_S=1$), while varying $p_{>0}\in[0,1]$. Due to space constraints within our plots, we suppress the legend in all but the plot in Figure 2(a) (in particular, this same legend applies to all plots within this paper). Rounding to five decimal places, we observe that for $0\le p_{>0}<0.13351$, class-1 preemptive priority (i.e., $\mathcal{I}=1$) is optimal in terms of minimizing the mean sojourn time and maximizing the mean number of working machines, whereas class-1 non-preemptive priority (i.e., $\mathcal{I}=-1$) is optimal for $0.13351\le p_{>0}<0.68277$, and exhaustive (i.e., $\mathcal{I}=0$) is optimal otherwise. Based on our earlier intuition concerning switch-in times and priority service policies, this makes sense. It is optimal for the server to switch upon every class-1 failure when the probability of experiencing a non-zero switch-in time is minimal, but as this probability increases, it no longer becomes optimal to interrupt a class-2 service, eventually reaching the point where the server wishes to eliminate any unnecessary switches. An important observation here is that the optimality of $\mathcal{I}$ changes simultaneously for both the mean sojourn time and mean number of working machines. Therefore, with all else being equal, the choice of $\mathcal{I}$ that maximizes the number of working machines will also minimize the amount of time between a machine's failure and when it is up and working again.

We point out that there does not exist a perfectly linear relationship between $\mathrm{E}[S]$ and $\mathrm{E}[N_W]=C-\mathrm{E}[X_1]-\mathrm{E}[X_2]$. Little's Law [17] states that the expected number of customers present in a system is equal to the expected amount of time a customer spends in system, multiplied by the average arrival rate. For many models, said arrival rate is constant, and corresponds to one or more Poisson processes that are independent of the rest of the system. Within this model, however, customers "arrive" as machines fail at a rate directly proportional to the number of working machines. In that way, the average arrival rate satisfies $\bar{\alpha} = \alpha \mathrm{E}[N_W]$, leading to the relationship $\mathrm{E}[X_1] + \mathrm{E}[X_2] = \bar{\alpha}\mathrm{E}[S]$ $= \alpha \mathrm{E}[N_W]\mathrm{E}[S]$. Seeing as how $\mathrm{E}[N_W]$ is not a constant in Figure 2, it is clear that there would not be an exact linear relationship between it (being a linear function of $\mathrm{E}[X_1] + \mathrm{E}[X_2]$) and $\mathrm{E}[S]$. In fact, it is easy to show that

$$\mathrm{E}[N_W] = \frac{C}{1 + \alpha \mathrm{E}[S]},$$

which proves the claim that minimizing $\mathrm{E}[S]$ is equivalent to maximizing $\mathrm{E}[N_W]$, at least when C and α are constant.

We are able to make similar conclusions between the effect of switch-in times and priority service policy optimality from Figure 3, by setting $p_{>0} = 1$ and letting M_S range between 0 and 2. Even with a guaranteed positive switch-in time, class-1 preemptive priority is optimal for the smallest mean values. This is followed by a small range where class-1 non-preemptive priority is optimal, followed by exhaustive, which continues to be the best choice as M_S becomes large. In both Figures 2 and 3, we remark at how fast class-1 preemptive priority switches from being the best choice to being the worst, as the cost of the extra incurred switch-in times becomes too large. In these examples, class-1 non-preemptive priority at its worst is not too far from the class-2 priority models in Figure 2, but as the mean switch-in times themselves are increasing in Figure 3, the total amount of idle time we are "risking" is increasing and the higher rate of class-1 failures makes class-1 non-preemptive priority vastly underperform the class-2 priority service policies at large values of M_S.

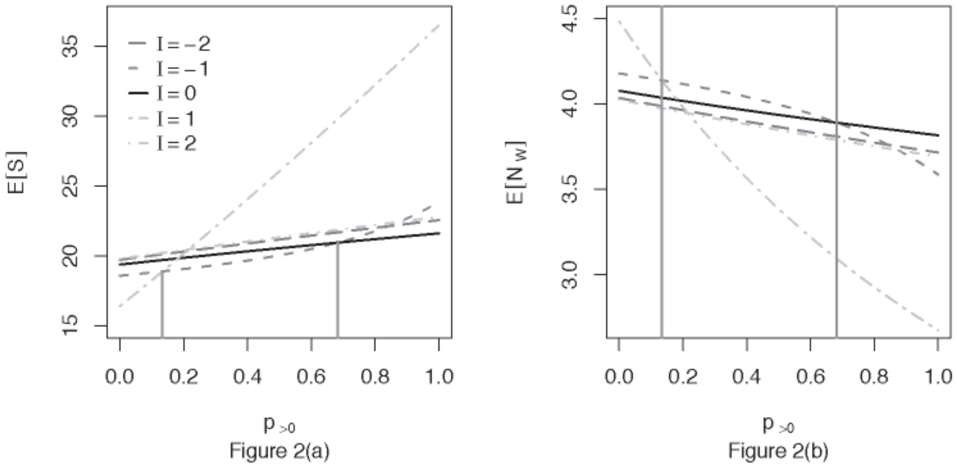

Figure 2: Plots of $\mathrm{E}[S]$ and $\mathrm{E}[N_W]$ versus $p_{>0}$ (along with vertical lines indicating values of $p_{>0}$ where the optimal choice of $\mathcal{I}$ changes), with $C = 10$, $\alpha = 0.075$, $M_S = 1$, and H_2 service with $M_B = 1$.

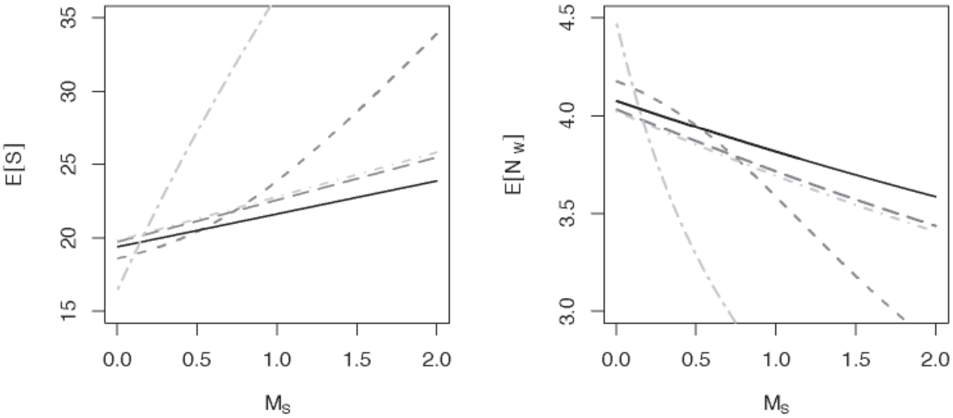

Figure 3(a)

Figure 3(b)

Figure 3: Plots of E$[S]$ and E$[N_W]$ versus M_S, with $C = 10$, $\alpha = 0.075$, $p_{>0} = 1$, and H_2 service with $M_B = 1$.

We now expand our scope and consider $\alpha \in \{0.05, 0.075, 0.1\}$, $C \in \{2, 3, \ldots, 18\}$, $p_{>0} \in \{0, 0.5, 1\}$, and Erlang-3 (E_3) service time distributions with phase-type components

$$\underline{\beta}_1 = \underline{\beta}_2 = (1,0,0),\ B_1 = \begin{pmatrix} -3 & 3 & 0 \\ 0 & -3 & 3 \\ 0 & 0 & -3 \end{pmatrix},\ \text{and}\ B_2 = \frac{1}{20M_B}\begin{pmatrix} -3 & 3 & 0 \\ 0 & -3 & 3 \\ 0 & 0 & -3 \end{pmatrix}.$$

Within the preemptive priority service policies, the E_3 model enables us to represent the possibility of having a partially completed service to return to (since we allow preemptive resume). In comparison to the H_2 service time distribution, the E_3 distribution will have smaller variance but equal mean at a given value of M_B.

Using the E_3 service time distributions, Figures 4 and 5 plot E$[N_W]$ versus C for different combinations of $p_{>0}$, α, M_B, and M_S. Clearly, as we increase C, E$[N_W]$ will converge to some constant value, potentially depending on $\mathcal{I}$. This is a result of the existence of a tipping point where the server's rate of fixing machines balances out with the total failure rate of working machines. Any further machines introduced into the system after this limit is reached will effectively increase the average number of broken machines by 1.

When $p_{>0} = 0$, the additional switches that a server experiences from a priority service policy do not result in any idle time, and so each policy converges to the same value of E$[N_W]$, albeit at different rates. When $p_{>0} = 0.5$, we observe that each service policy now converges to a different value of E$[N_W]$. This is due to the fact that different priority service policies introduce different amounts of extra switch-ins, which result in different percentages of time that the server is idle. The higher percentage of time that the server is idle, the smaller the net rate of fixing machines per unit time. As the probability of a failure coming from class 1 is much higher than that of class 2, class-1 preemptive priority results in the highest amount of extra switch-ins due to the long class-2 service times, followed by class-1 non-preemptive priority. The class-2 priority policies introduce similar amounts of extra switch-ins due to a combination of the lower frequency of class-2 failures and the faster class-1 service times. At $p_{>0} = 1$, this difference is further amplified and we see an increased amount of separation. A consequence of this is that the exhaustive service policy always converges to the highest value of E$[N_W]$ as $C \to \infty$, but as it does not necessarily do so at the fastest rate, other policies may yield a higher $\mathrm{E}[N_W]$ at a particular value of C.

Comparing Figures 4(a)-(c) against Figures 4(d)-(f), we can see that since increasing α results in a faster rate of occurrence for both failure classes, the spread of converged values of $\mathrm{E}[N_W]$ for each service policy is wider as the extra amount of idle time is increased. It is also notable to point out that the higher rate of failure causes a reduction in all converged values, given that the server's rate of repair is unchanged. Moreover, the increased rate of failure results in a faster rate of convergence, as each additional working machine contributes a larger amount to the total rate of failure.

We next compare Figures 4(a)-(c) against Figures 5(a)-(c) to ascertain the impact of increasing M_S. Similar to increasing $p_{>0}$, at positive values of $p_{>0}$, we remark that this penalizes the priority service policies proportional to their amount of extra incurred switch-ins. As the exhaustive service policy has minimal incurred switch-ins, its converged $\mathrm{E}[N_W]$ values are impacted the least.

Finally, observing Figures 4(a)-(c) and Figures 5(d)-(f), we note that the ratio of mean service times between the two classes is affected. In Figures 5(d)-(f), we have $M_B = 0.5$, which halves the mean class-2 service time while leaving the class-1 service time distribution unchanged. This increases the rate at which the server repairs machines, and so the rates of convergence are slower to higher final values. The quicker class-2 service times reduce the effectiveness of the class-1 priority policies (while marginally improving the class-2 priority policies), so this narrows the differences in $\mathrm{E}[N_W]$ between the priority service policies and the exhaustive service policy.

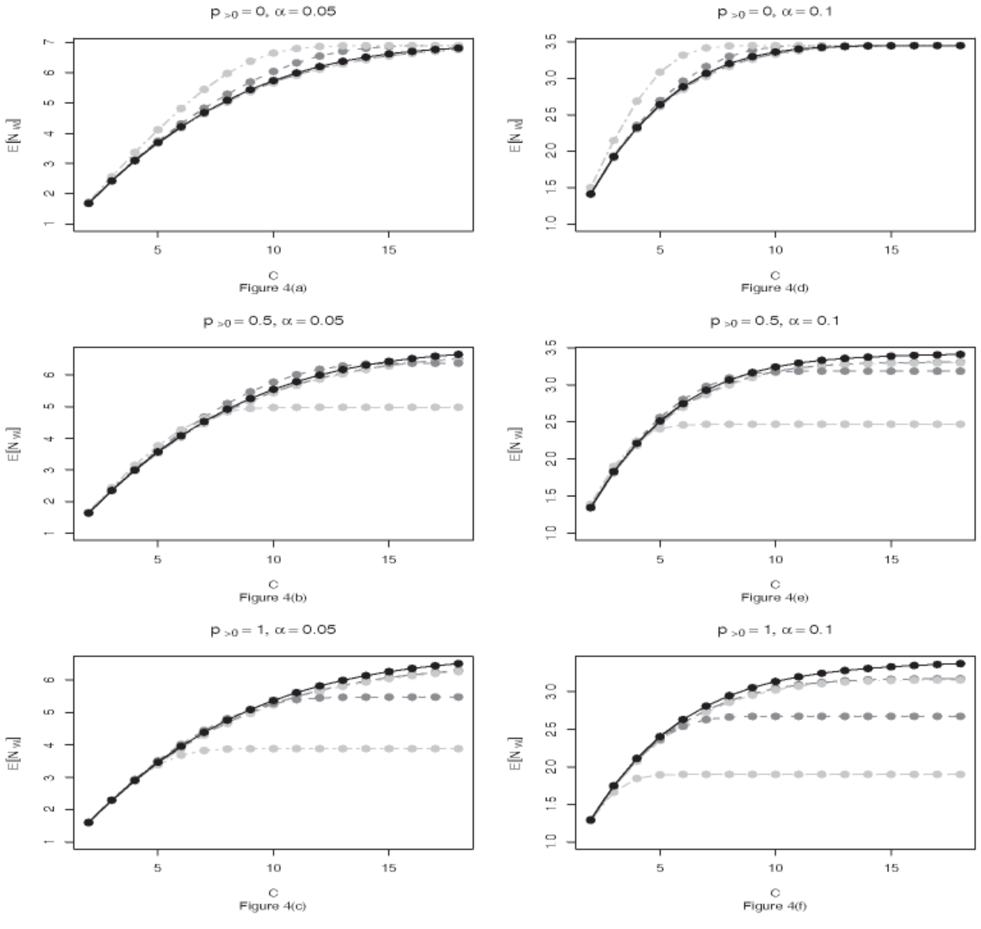

Figure 4: Plots of $\mathrm{E}[N_W]$ versus C under E_3 service and fixed $M_S = 1$ and $M_B = 1$, for varying $p_{>0}$ and α.

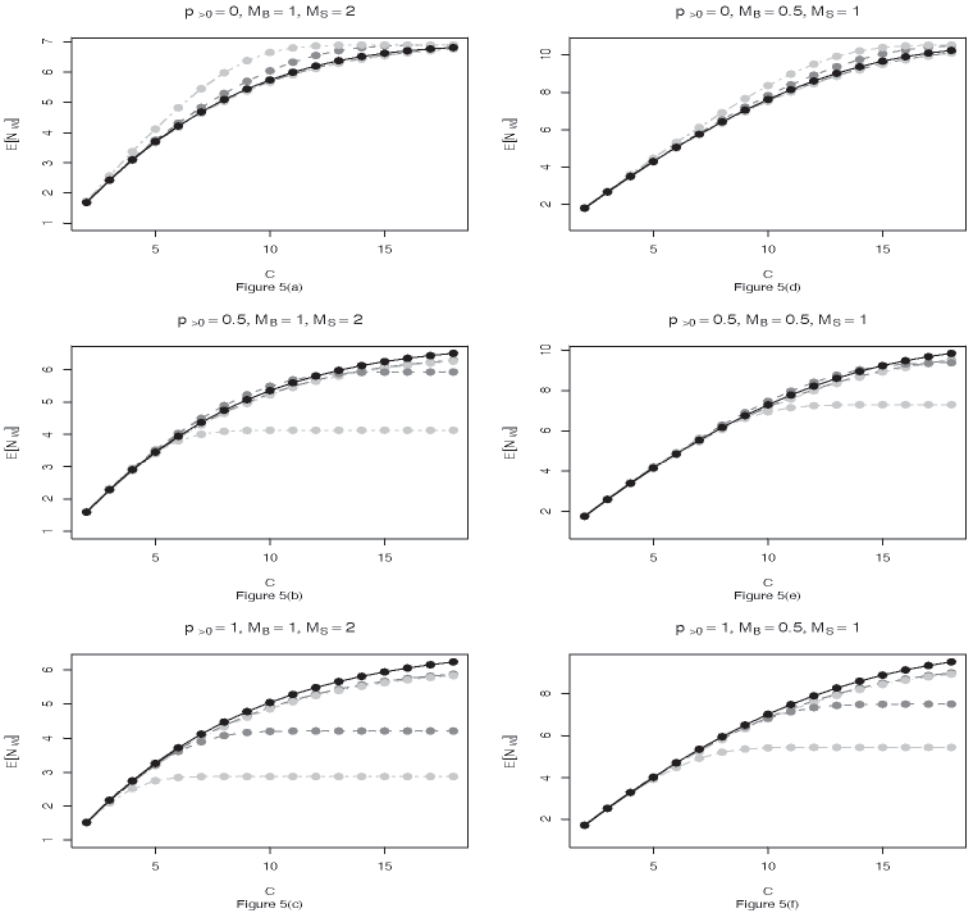

Figure 5(a)

Figure 5(d)

Figure 5(b)

Figure 5(e)

Figure 5(c)

Figure 5(f)

Figure 5: Plots of $\mathrm{E}[N_W]$ versus C under E_3 service and fixed $\alpha = 0.05$, for varying $p_{>0}$, M_S, and M_B.

If additional machines were cost-free, then a factory could achieve a maximum expected rate of production by selecting an exhaustive service policy and increasing C to an arbitrarily large value. However, in the real world, there are in fact restrictions on how many machines can be purchased, either due to capital or space restrictions. Due to the existence of costs, the correct decision may be to use a priority service policy at a value of C that results in a higher value of $\mathrm{E}[N_W]$ than the exhaustive service policy. To approximate this, we introduce the objective function $\mathrm{E}[N_W]-rC$, where r is the cost of possession for each machine in the system. This constant r can be interpreted as the cost per unit time as a fraction of the profit per unit time that a single working machine produces. In this case, the optimal choice of C and $\mathcal{I}$ will maximize our expected profit per unit time. Alternatively, r may be treated as the tolerance that we select to determine if $\mathrm{E}[N_W]$ has converged, such that the objective function will be locally maximized for a given $\mathcal{I}$ at the highest value of C before every additional machine added to the system results in an increase in $\mathrm{E}[N_W]$ of less than r units. Global maximization in this case tells us which service policy converges within the tolerance to the highest value, the fastest.

In Tables 1 and 2, we find the optimal C and $\mathcal{I}$ under the H_2 and E_3 service time distributions, respectively, over the aforementioned ranges of M_B, M_S, α, and $p_{>0}$. Additionally, we consider values of the cost parameter $r \in \{0.05, 0.1, 0.25\}$. Comparing these tables, it is clear that the smaller service time variance of the E_3 distributions causes the objective function to converge to higher values, often at smaller values of C. Here, a smaller service time variance reduces the probability of the server being stuck on one job for an unusually long period of time, resulting in machines being repaired at a more consistent rate. However, we do not observe a large impact on the optimal choices of $\mathcal{I}$, outside of the case when $M_B = 0.5$ and $M_S = 2$, where the optimal C values for the E_3 distributions are higher. Here, we see that exhaustive service is preferred over class-1 non-preemptive priority, which we would expect to observe at higher values of C.

When $p_{>0} = 0$, all service policies converge to the same value of $\mathrm{E}[N_W]$ (all else being equal), but the class-1 preemptive priority policy is universally preferred as it converges at the fastest rate. For moderate values of $p_{>0}$, either class-1 non-preemptive priority or exhaustive service is optimal, largely conditional on r, M_B, and M_S. For larger r, the cost per machine is higher, so that the objective function will maximize at a lower value of C. As the exhaustive service policy is best for large C, but not necessarily small C, it is possible for the optimal C to end up in the range where class-1 non-preemptive priority results in a higher value of $\mathrm{E}[N_W]$. Reducing the mean class-2 service time, as observed in Figures 4 and 5, causes the objective function to maximize at higher values of C, to a larger expected profit per unit time. For moderate values of $p_{>0}$, this may result in exhaustive service being preferred over class-1 non-preemptive priority. Finally, as M_S increases, the additional switch-in times that the non-preemptive priority service policy causes reduces the region where $\mathcal{I} = -1$ outperforms $\mathcal{I} = 0$ to potentially no values of C, so that exhaustive service becomes the best choice. Not surprisingly, exhaustive service performs the best over these ranges when $p_{>0} = 1$.

To close this section, we end with some remarks concerning the computations required to obtain the data used in this section. For Tables 1 and 2, we used the programming language R for all of our computations, and then analyzed our data in Excel. As we have made no approximations in any part of our analysis, the accuracy of our data is limited solely by the intrinsic functions built into R. For each combination of M_B, M_S, α, and $p_{>0}$, with either H_2 or E_3 service, the mean queue lengths for both classes were calculated from the steady-state probabilities for all $C \in \{2,3,\ldots,18\}$ and $\mathcal{I} \in \{-2,-1,0,1,2\}$, so that within Excel we were free to vary our cost parameter r and immediately obtain the corresponding optimal C and $\mathcal{I}$ for each case. Even if the total state space is large for a given C, since the QBD algorithm for deriving the steady-state probabilities outlined in Section 2 only involves operations on the smaller $Q_{i,j}^{[C]}$ blocks (i.e., not on the entire $Q^{[C]}$ matrix), it is possible to calculate $\mathrm{E}[N_W]$ for a given C and $\mathcal{I}$ from these ranges in under a second using a 4.00 GHz i7-6700K processor.

To obtain mean sojourn times, we applied the formula for the first moment of a phase-type distribution. For example, to obtain the expected sojourn time of a class-1 customer when $\mathcal{I} \in \{-2,-1,0\}$, we computed

$$-(\underline{\Phi}, (q_{0,0,\bullet,\bullet}\gamma_{01}^{[0]} + q_{0,+,3,\bullet}\gamma_{21}^{[0]})\underline{\beta}_1)\mathcal{T}^{-1}\underline{e}'.$$

However, inverting the phase-type rate matrices proved computationally intense for large C, even when making use of the block upper-diagonal structure of each $\mathcal{T}$ during the inversion process. For instance, to compute the mean sojourn time for both classes for a given C (i.e., the total time to calculate the mean sojourn times for each $\mathcal{I} \in \{-2,-1,0,1,2\}$) when $M_B = 1$, $M_S = 1$, $\alpha = 0.075$, and $p_{>0} = 0.5$ under H_2 (E_3) service, the calculations take under a second for $C = 5$, approximately 11 (22) seconds for $C = 11$, and approximately 340 (950) seconds for $C = 18$.

While we are unable to avoid these (potentially lengthy) matrix inversions and multiplications when calculating higher moments or distributional values for the sojourn times, we can make use of Little's Law when calculating their first moments. Specifically, after calculating the mean queue lengths $\mathrm{E}[X_1]$ and $\mathrm{E}[X_2]$ (and hence the expected number of working machines,

$\mathrm{E}[N_W] = C - \mathrm{E}[X_1] - \mathrm{E}[X_2]$), it immediately follows that the expected sojourn time for a machine suffering a class-1 failure is simply

$$\frac{\mathrm{E}[X_1]}{\alpha_1 \mathrm{E}[N_W]},$$

and the expected sojourn time for an arbitrary failed machine is

$$\frac{\mathrm{E}[X_1] + \mathrm{E}[X_2]}{\alpha \mathrm{E}[N_W]}.$$

As we only require the steady-state probabilities to calculate the above quantities, in contrast to applying phase-type formulas, Little's Law only takes approximately 2 seconds or less to calculate the exact same values for the mean sojourn times of both classes for a given C (up to 18) when $M_B = 1$, $M_S = 1$, $\alpha = 0.075$, and $p_{>0} = 0.5$ under H_2 or E_3 service. Therefore, it is advisable to make use of Little's Law when calculating the first moment.

Table 1. Optimal combinations of C and $\mathcal{I}$ under H_2 service.

							$p_{>0}$				
$r = 0.05$				0			0.5			1	
M_B	M_S	α	C	$\mathcal{I}$	$E[N_W]$	C	$\mathcal{I}$	$E[N_W]$	C	$\mathcal{I}$	$E[N_W]$
1	1	0.05	15	1	6.8352	18	0	6.4476	18	0	6.3070
		0.075	11	1	4.5423	15	0	4.3593	15	0	4.2557
		0.10	9	1	3.4025	12	0	3.2409	12	0	3.1517
	2	0.05	15	1	6.8352	18	0	6.2903	18	0	6.0387
		0.075	11	1	4.5423	16	0	4.2961	17	0	4.1733
		0.10	9	1	3.4025	12	0	3.1421	13	0	3.0430
0.5	1	0.05	18	1	10.2860	18	0	9.2835	18	0	9.0040
		0.075	15	1	6.9608	18	0	6.6601	18	0	6.4709
		0.10	12	1	5.2079	16	0	5.0348	17	0	4.9405
	2	0.05	18	1	10.2860	18	0	8.9543	18	0	8.4692
		0.075	15	1	6.9608	18	0	6.4526	18	0	6.1226
		0.10	12	1	5.2079	17	0	4.9323	18	0	4.7578
$r = 0.1$											
1	1	0.05	13	1	6.6948	15	-1	6.2379	17	0	6.2189
		0.075	10	1	4.4827	11	-1	4.1039	12	0	4.0405
		0.10	8	1	3.3439	9	-1	3.0572	9	0	2.9313
	2	0.05	13	1	6.6948	17	0	6.2002	17	0	5.9419
		0.075	10	1	4.4827	12	0	4.0240	12	0	3.8182
		0.10	8	1	3.3439	9	0	2.9156	9	0	2.7410
0.5	1	0.05	18	1	10.2860	18	0	9.2835	18	0	9.0040
		0.075	13	1	6.8193	16	0	6.4965	16	0	6.2871
		0.10	11	1	5.1484	13	0	4.8391	13	0	4.6614
	2	0.05	18	1	10.2860	18	0	8.9543	18	0	8.4692
		0.075	13	1	6.8193	17	0	6.3640	18	0	6.1226
		0.10	11	1	5.1484	13	0	4.6380	14	0	4.4440
$r = 0.25$											
1	1	0.05	10	1	6.1367	11	-1	5.5854	10	0	5.0791
		0.075	7	1	3.9922	7	-1	3.4423	7	0	3.2798
		0.10	6	1	3.0645	5	-1	2.4134	5	0	2.2979
	2	0.05	10	1	6.1367	10	-1	5.1427	10	0	4.8054
		0.075	7	1	3.9922	7	-1	3.3006	6	0	2.8259
		0.10	6	1	3.0645	5	-1	2.3031	4	0	1.8741
0.5	1	0.05	15	1	9.7449	15	-1	8.6819	16	0	8.5677
		0.075	11	1	6.4953	10	-1	5.4877	10	0	5.2170
		0.10	8	1	4.6639	8	-1	4.0735	8	0	3.8568
	2	0.05	15	1	9.7449	16	0	8.5116	16	0	8.0301
		0.075	11	1	6.4953	10	-1	5.1787	10	0	4.8051
		0.10	8	1	4.6639	7	-1	3.5766	7	0	3.2658

Table 2. Optimal combinations of $\mathcal{C}$ and $\mathcal{I}$ under E_3 service.

							$p_{>0}$				
$r=0.05$				0			0.5			1	
M_B	M_S	α	$\mathcal{C}$	$\mathcal{I}$	$E[N_W]$	$\mathcal{C}$	$\mathcal{I}$	$E[N_W]$	$\mathcal{C}$	$\mathcal{I}$	$E[N_W]$
1	1	0.05	12	1	6.8660	18	0	6.6507	18	0	6.5040
		0.075	9	1	4.5771	13	0	4.3870	14	0	4.3240
		0.10	7	1	3.4162	11	0	3.2938	11	0	3.1979
	2	0.05	12	1	6.8660	18	0	6.4986	18	0	6.2331
		0.075	9	1	4.5771	14	0	4.3204	16	0	4.2377
		0.10	7	1	3.4162	11	0	3.1941	13	0	3.1412
0.5	1	0.05	16	1	10.4744	18	0	9.8396	18	0	9.5066
		0.075	12	1	6.9922	16	0	6.7514	18	0	6.6736
		0.10	9	1	5.2021	13	0	5.0391	15	0	4.9872
	2	0.05	16	1	10.4744	18	0	9.4846	18	0	8.9179
		0.075	12	1	6.9922	18	0	6.6718	18	0	6.3756
		0.10	9	1	5.2021	15	0	4.9859	17	0	4.8770
$r=0.1$											
1	1	0.05	11	1	6.7993	13	-1	6.2868	15	0	6.2558
		0.075	8	1	4.5144	9	-1	4.0746	11	0	4.1050
		0.10	6	1	3.3179	8	-1	3.0924	9	0	3.0543
	2	0.05	11	1	6.7993	16	0	6.3466	16	0	6.0524
		0.075	8	1	4.5144	11	0	4.0958	12	0	3.9632
		0.10	6	1	3.3179	9	0	3.0464	9	0	2.8522
0.5	1	0.05	15	1	10.3849	18	0	9.8396	18	0	9.5066
		0.075	11	1	6.9260	14	0	6.5952	15	0	6.4593
		0.10	9	1	5.2021	11	0	4.8806	12	0	4.7810
	2	0.05	15	1	10.3849	18	0	9.4846	18	0	8.9179
		0.075	11	1	6.9260	15	0	6.4505	17	0	6.2854
		0.10	9	1	5.2021	12	0	4.7729	14	0	4.6623
$r=0.25$											
1	1	0.05	10	1	6.6494	10	-1	5.7733	10	0	5.3707
		0.075	7	1	4.3540	7	-1	3.6871	7	0	3.4490
		0.10	5	1	3.0849	5	-1	2.5601	5	0	2.4048
	2	0.05	10	1	6.6494	10	-1	5.4871	10	0	5.0456
		0.075	7	1	4.3540	7	-1	3.4972	7	0	3.2007
		0.10	5	1	3.0849	5	-1	2.4237	5	0	2.2077
0.5	1	0.05	14	1	10.2076	14	-1	9.0266	15	0	8.8760
		0.075	10	1	6.7681	10	-1	5.9058	11	0	5.8309
		0.10	8	1	5.0649	8	-1	4.3761	8	0	4.1177
	2	0.05	14	1	10.2076	16	0	9.0887	16	0	8.4950
		0.075	10	1	6.7681	11	0	5.8022	11	0	5.3443
		0.10	8	1	5.0649	8	0	4.0903	8	0	3.7190

6. Concluding Remarks

We have examined a closed queueing maintenance model with two classes of machine failures which are repaired by a single server in a manner determined by a particular service policy. Exhaustive service as well as both non-preemptive and preemptive resume service policies were considered. Through the use of matrix analytic methods, the steady-state joint queue length distribution, as well as the distribution of the time between a target machine's failure and repair completion, have been found and expressed in phase-type form. We have also conducted a numerical analysis to investigate the influence of several model components on the expected number of working machines, and how these components affect the optimal choice of service policy when our goal is to maximize this expected number while incurring a cost per machine possessed in the system.

In future work, we anticipate extending this model by generalizing to hybrid priority policies, which will allow the server's decision to move to depend on the queue lengths of both classes. We also intend to introduce the idea of a maintenance float of spare machines into the model, by way of placing a cap on the number of machines that can work (and hence be at risk of failure), which may be less than the total number of machines in the system. Another interesting problem to investigate is the optimal manner of assigning different sized failures into "small" or "large" classes when the working machines are in fact susceptible to more than two distinct types of failure.

Another direction for a future extension of this model is to expand beyond the current two-level priority system to three or more priority levels. If we further divide the job types into more homogeneous classifications in terms of service distribution, we would expect to observe further gains in system optimization, dependent on the switch-in time distributions. For instance, if switch-in times were negligible, then a preemptive resume priority discipline assigning higher priority to classes with shorter expected repair times would bring us closer to the shortest-job-first service discipline, which is known to be optimal. However, under our current approach, the addition of another class of machine failures would necessitate an increase in the dimensionality of our model by one or two variables (depending on the service policy) when specifying the modified sub-matrices, which would correspondingly increase computation time and memory requirements for a system having the same total number of machines. Therefore, it would be advantageous to investigate the use of other priority queueing techniques that can reduce the effective state space for calculations.

Acknowledgment

Steve Drekic and Kevin Granville acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada through its Discovery Grants program (#RGPIN-2016-03685) and Postgraduate Scholarship-Doctoral program, respectively. The authors are also very grateful to the anonymous referees for their careful reading of the paper and their useful suggestions that have improved the overall presentation of the paper.

References

[1] Abboud, N. E. (1996). The Markovian two-echelon repairable item provisioning problem. *Journal of the Operational Research Society*, 47(2), 284-296.

[2] Alfa, A. S., & Castro, I. T. (2002). Discrete time analysis of a repairable machine. *Journal of Applied Probability*, 39(3), 503-516.

[3] Boon, M. A. A. (2011). Polling Models: From Theory to Traffic Intersections. Doctoral dissertation, Eindhoven: Technische Universiteit Eindhoven, 190 pages.

[4] Boon, M. A. A., van der Mei, R. D., & Winands, E. M. M. (2011). Applications of polling systems. *Surveys in Operations Research and Management Science*, 16(2), 67-82.

[5] Buyukkramikli, N. C., van Ooijen, H. P. G., & Bertrand, J. W. M. (2015). Integrating inventory control and capacity management at a maintenance service provider. *Annals of Operations Research*, 231(1), 185-206.

[6] Chakravarthy, S. R. (2012). Maintenance of a deteriorating single server system with Markovian arrivals and random shocks. *European Journal of Operational Research*, 222(3), 508-522.

[7] Gaver, D., Jacobs, P., & Latouche, G. (1984). Finite birth-and-death models in randomly changing environments. *Advances in Applied Probability*, 16(4), 715-731.

[8] Gordon, W. J., & Newell, G. F. (1967). Closed queuing systems with exponential servers. *Operations Research*, 15(2), 254-265.

[9] Gross, D., Miller, D. R., & Soland, R. M. (1983). A closed queueing network model for multi-echelon repairable item provisioning. *IIE Transactions*, 15(4), 344-352.

[10] He, Q.-M. (2014). Fundamentals of Matrix-Analytic Methods. Springer-Verlag, New York.

[11] Hsu, L. F. (1999). Simultaneous determination of preventive maintenance and replacement policies in a queue-like production system with minimal repair. *Reliability Engineering & System Safety*, 63(2), 161-167.

[12] Kim, S. K., & Dshalalow, J. H. (2003). A versatile stochastic maintenance model with reserve and super-reserve machines. *Methodology and Computing in Applied Probability*, 5(1), 59-84.

[13] Kim, W. B., & Koenigsberg, E. (1987). The efficiency of two groups of N machines served by a single robot. *Journal of the Operational Research Society*, 38(6), 523-538.

[14] Lakatos, L., Szeidl, L., & Telek, M. (2013). Introduction to Queueing Systems with Telecommunication Applications. Springer, Berlin.

[15] Levy, H., & Sidi, M. (1990). Polling systems: applications, modeling and optimization. *IEEE Transactions on Communications*, COM-38(10), 1750-1760.

[16] Lin, C., Madu, C. N., & Kuei, C. H. (1994). A closed queuing maintenance network for a flexible manufacturing system. *Microelectronics Reliability*, 34(11), 1733-1744.

[17] Little, J. D. (1961). A proof for the queuing formula: $L = \lambda W$. *Operations Research*, 9(3), 383-387.

[18] Mack, C. (1957). The efficiency of N machines uni-directionally patrolled by one operative when walking time is constant and repair times are variable. *Journal of the Royal Statistical Society. Series B (Methodological)*, 19(1), 173-178.

[19] Mack, C., Murphy, T., & Webb, N. (1957). The efficiency of N machines uni-directionally patrolled by one operative when walking time and repair times are constants. *Journal of the Royal Statistical Society. Series B (Methodological)*, 19(1), 166-172.

[20] Madu, C. N. (1988). A closed queueing maintenance network with two repair centres. *Journal of the Operational Research Society*, 39(10), 959-967.

[21] Morris, R. J. T. (1981). Priority queuing networks. *The Bell System Technical Journal*, 60(8), 1745-1769.

[22] Neuts, M. F., Pérez-Ocón, R., & Torres-Castro, I. (2000). Repairable models with operating and repair times governed by phase type distributions. *Advances in Applied Probability*, 32(2), 468-479.

[23] Pérez-Ocón, R., & Montoro-Cazorla, D. (2004). Transient analysis of a repairable system, using phase-type distributions and geometric processes. *IEEE Transactions on Reliability*, 53(2), 185-192.

[24] Perry, D., & Posner, M. J. M. (2000). A correlated M/G/1-type queue with randomized server repair and maintenance modes. *Operations Research Letters*, 26(3), 137-147.

[25] Peschansky, A. I., & Kovalenko, A. I. (2016). On a strategy for the maintenance of an unreliable channel of a one-server loss queue. *Automatic Control and Computer Sciences*, 50(6), 397-407.

[26] Righter, R. (2002). Optimal maintenance and operation of a system with backup components. *Probability in the Engineering and Informational Sciences*, 16(3), 339-349.

[27] Takagi, H. (1988). Queueing analysis of polling models. *ACM Computing Surveys*, 20(1), 5-28.

[28] Vishnevskii, V. M., & Semenova, O. V. (2006). Mathematical methods to study the polling systems. *Automation and Remote Control*, 67(2), 173-220.

[29] Yang, W. S., Lim, D. E., & Chae, K. C. (2009). Maintenance of deteriorating single server queues with random shocks. *Computers and Industrial Engineering*, 57(4), 1404-1406.