Modelling Queues Where Customers Randomly Change Priority Classes While Waiting

Geraint I. Palmer, Michalis Panayidis, Vincent Knight & Elizabeth Williams

Abstract

In this paper a generalised model of dynamically changing priorities within an M/M/c queue is presented. A simulation and Markov chain models are given that describes the behaviour of such systems, and their stationarity is explored. Bounded approximations of the Markov models are given, and measures of their accuracy in approximating the infinite versions given. Finally the models are used to model the waiting list for surgical endoscopies, with unknown service disciplines, fitting system parameters to reflect the queue behaviour. An exploration of behaviour under different class change parameters is given for a better understanding of the system.

1 Introduction

There are a number of situations in which a customer's priority in a queue might change during their time queueing, or equivalently where their priority depends on the amount of time they have already spent in the queue. Classic examples arise in healthcare systems, for example when a patient's medical urgency might increase the longer they spend waiting due to health degeneration. Another example would be a prioritisation scheme that attempts a trade-off between medical need and waiting times. These are both examples where a patient's priority has the chance to upgrade over time while in the queue. However there also might be situations in which a patient's priority can downgrade over time: consider a medical intervention that can improve a patient's outcome if caught early, if a patient has been waiting a long time already then they might be passed over for a newly referred patient who will gain more benefit from the intervention. In this case a patient's priority is downgraded the longer they wait.

In this paper a single M/M/c queue is modelled, with multiple classes of customer of different priorities. While waiting in the queue customers change their class to any other class at specific rates. Thus upgrades, downgrades, and 'skip-grades' (moving to a priority class not immediately above or below the current class) are modelled.

This is first modelled using simulation, where we describe generalisable logic. This was first implemented in version v2.3.0 of the Ciw library in Python [17] and is a contribution of this paper. Then two Markov chain models are defined for the system, which are used to find steady state distributions and expected sojourn times for each customer class. These Markov chains give some insights into the behaviour of the systems under different combinations of parameters; and numerical experiments give further behaviours.

This paper is structured as follows: Section 1.1 gives a motivating example from a healthcare setting demonstrating the need for this type of model. Section 1.2 highlights some previous and related work. Section 2 defines the system under consideration in detail. Section 3 discusses the contribution to the Ciw library and the logic required to simulate this system. Section 4 defines two Markov chain models of the system, one useful for considering system-wide statistics such as state probabilities, and one useful for considering customers' statistics such as average sojourn time. This includes exploring a bounded approximation for numerically tractable analysis, presents measures of accuracy for these bounded approximations, and the existence or otherwise of systems that can reach steady state. Section 5 experimentally justifies the use of

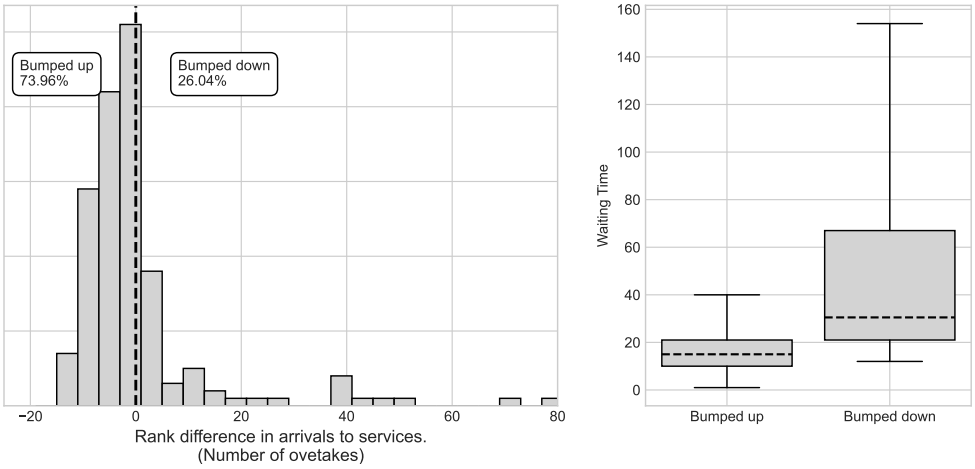


Figure 1: Distribution of overtakes in a CAVUHB endoscopy waiting list.

these models presented to model scenarios where prioritisation rules are unknown, using the aforementioned case study as an example; an exploration of the system behaviour under different parameters is given.

1.1 Motivating Case Study

We consider first attempt to model the queueing process for a particular surgical procedure; we consider and investigate the waiting list of a surgical endoscopy procedure carried out in the Cardiff and Value University Health Board (CAVUHB) over the period 1st January to 31st December 2021.

It is clear from examining the data that a first-in-first-out (FIFO) service discipline was not used. To show this, we assign each patient in the data set an arrival rank, the order in which they were referred for an endoscopy, and a service rank, the order in which they received their endoscopy. The difference in the two ranks corresponds to the net number of patients that they 'overtook' in the queue: a negative number indicates more customers overtook them, that is they were 'bumped down'the queue; while a positive number indicates that they overtook more people, and where 'bumped up' the queue. During the observation period, 169 patients received an endoscopy, and Figure 1 shows the distribution of the rank differences, and the distribution of waiting times by those who were bumped up or down the queue. No patient had a waiting time of zero, indicating that no patient was referred to the queue in a prioritised state, hinting that patients may have priorities that change over time.

1.2 Related Work

Systems of this kind have been investigated previously:

- [9] (1960): Non preemptive M/M/1 where customers are served in order of the difference between their waiting time and urgency number (that is priorities increasing linearly over time). Solved by considering event probabilities at clock ticks.
- [10] (1964): Another formulation giving the same behaviour as [9], but now for both non preemptive and preemptive priorities, and multiple customer classes. Called 'delay dependent' or 'time dependant' priorities, and recently by 'accumulating' priorities.
- [8] (1971): Similar to [9], but treat each urgency number as a separate customer class, and not considering clock ticks. Upper and lower bounds on the waiting times, based on FIFO and static priorities.

- [15] (1979): Now considers the case where priorities increase non-linearly but concavely over time.
- [6] (1990): Non preemptive M/G/1 queue with two classes of customers, where priorities switch if the number from one class exceeds a given threshold. Lower priority customers have a finite waiting capacity, higher have infinite capacity.
- [23] (1995): Introduces the generalised $c\mu$-rule (first conceived in [21]), which applies a class- and waiting time-dependant cost to each customer. This acts as a scheduling rule, but can also model dynamic priorities amongst customers.
- [12] (2003): Similar to [6] but with Markovian services and infinite waiting capacities for both customers.
- [24] (2008): Preemptive n-priority-classes M/M/c with exponential upgrades. Customers only upgrade to the priority immediately higher than themselves. Stability considered.
- [4] (2010): Preemptive two-priority-classes M/M/c with exponential upgrades. Customers cannot upgrade if the number of lower priority customers is below a given threshold. Holding costs considered.
- [7] (2012): Extension of [4], allows batch arrivals, multiple classes, phase-type upgrades and services. Customers only upgrade to the priority immediately higher than themselves.
- [22] (2014): Furthers the work of [10] to look at the maximum priority of the waiting customers in a single server queue as a stochastic process. This is extended in [20] to multi-server queues.
- [11] (2020): Upgrades and downgrades after a random, exponentially distributed, amount of time. Models two priority classes in a single server, finite buffer system with batch arrivals. Extended in [5] to include unreliable services and impatient customers.
- [1] (2022): Analytical (truncated) expressions for a two class delayed accumulating priority M/G/1 queue. Customer priorities increase linearly over time, at different rates according to class, after an initial fixed delay.
- [3] (2019): Evidence that in Canada that decision makers often use their own discretion in deciding which patients to be seen, rather than FIFO within each triage category. Fits prioritisation rules to this discretionary behaviour.

2 System Under Consideration

Consider an $M/M/c$ queue with K classes of customer labelled $0, 1, 2, \ldots, K-1$. Let:

- λ_k be the arrival rate of customers of class k,
- μ_k be the service rate of customers of class k,
- $\theta_{i,j}$ be the rate at which customers of class i change to customers of class j while they are waiting in line.

Customers of class i have priority over customers of class j if $i < j$. Customers of the same class are served in the order they arrived to that class. Figure 2 shows an example with two classes of customer.

The key feature here is the $K \times K$ class change matrix $\Theta = (\theta_{i,j})$. All elements $\theta_{i,j}$ where $i \neq j$ are rates, and so are non-negative real numbers, if customers of class i cannot change to customers of class j directly, then $\theta_{i,j} = 0$. The diagonal values $\theta_{i,i}$ are unused as customers cannot change to their own class. All elements $\theta_{i,i-1}$ represent the direct upgrade rates; all elements $\theta_{i,i+1}$ represent the direct downgrade rates, while all other elements can be thought of as 'skip-grades'. This is shown in Figure 3.

The priorities are pre-emptive, that is if a newly arriving customer has a higher priority than a customer in service, or if a waiting customer changes priority class to a higher priority than a customer is service, then

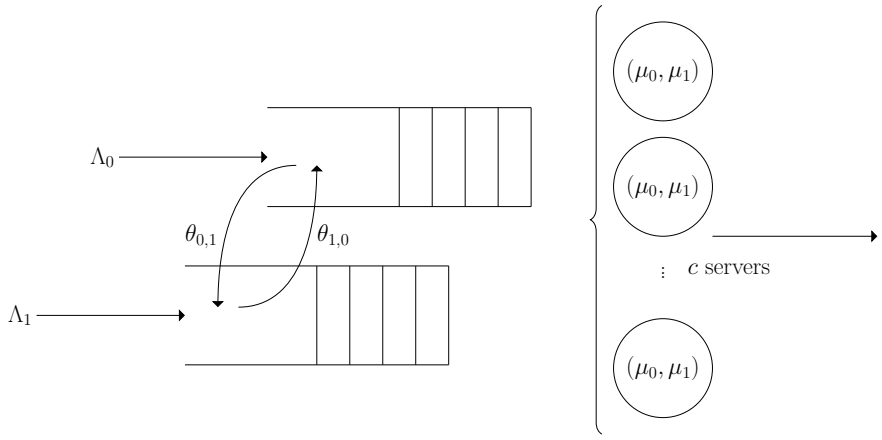


Figure 2: An example of a two-class priority queue.

$$\Theta = \begin{pmatrix} - & \theta_{01} & \theta_{02} & \theta_{03} & \theta_{04} \\ \theta_{10} & - & \theta_{12} & \theta_{13} & \theta_{14} \\ \theta_{20} & \theta_{21} & - & \theta_{23} & \theta_{24} \\ \theta_{30} & \theta_{31} & \theta_{32} & - & \theta_{34} \\ \theta_{40} & \theta_{41} & \theta_{42} & \theta_{43} & - \end{pmatrix}$$

—— downgrades
- - - upgrades
····· 'skip-grades'

Figure 3: Representations of parts of the matrix Θ. Example when $K = 5$.

that new customer displaces the lowest priority customer that is in service. The displaced customer rejoins the queue, before all other customers of their own or lower priority classes, but behind all other customers of higher priority classes. When that displaced customer eventually enters service again, their service time can either be resumed, re-started, or re-sampled.

The next two sections outline and compare a discrete devent simulation and Markov chain formulations for modelling this setup.

3 Simulation Model Logic

The Ciw library [17] is an open-source Python library for discrete-event simulation of open queueing networks. A key contribution of this work is the adaptation of the library's logic to facilitate the type of dynamically changing priority classes described in Section 2. This adaptation was first released in version Ciw v2.3.0, with usage documentation at `https://ciw.readthedocs.io/en/latest/Guides/CustomerClasses/change-class-while-queueing.html`. Figure 4 shows the Ciw code required to simulate the system with two classes of customer, $\lambda_1 = 1$, $\lambda_2 = 3$, $\mu_1 = 3$, $\mu_2 = 2$, $c = 1$, $\theta_{12} = 1.5$, and $\theta_{21} = 0.5$, for 365 time units. Here the key parameters are `priority_classes`, which takes a tuple containing a Python dictionary that maps customer class labels to priority rankings, and a list of pre-emption options for each node; and `class_change_time_distributions`, defining the time distributions used to describe the time it takes to transfer from one customer class to another. Note here that the simulation framework is general enough to define any probability distribution here, and is not restricted to Exponential distributions.

The core of Ciw's logic is the event scheduling approach, described in [19]. This is a variant of the three-phase approach, with an **A**-phase which advances the clock to the next scheduled event, a **B**-phase where

```
>>> import ciw
>>> N = ciw.create_network(
...     arrival_distributions={
...         'C0': [ciw.dists.Exponential(rate=1)],
...         'C1': [ciw.dists.Exponential(rate=3)]
...     },
...     service_distributions={
...         'C0': [ciw.dists.Exponential(rate=3)],
...         'C1': [ciw.dists.Exponential(rate=2)],
...     },
...     number_of_servers=[1],
...     priority_classes=({'C0': 0, 'C1': 1}, ["resample"]),
...     class_change_time_distributions={
...         'C0': {'C1': ciw.dists.Exponential(rate=1.5)},
...         'C1': {'C0': ciw.dists.Exponential(rate=0.5)}
...     }
... )
>>> Q = ciw.Simulation(network=N)
>>> Q.simulate_until_max_time(max_simulation_time=365)
```

Figure 4: Example Ciw code to simulate an M/M/1 queue with dynamic priorities.

scheduled events are carried out, and a **C**-phase where conditional events are carried out. Figure 5 shows a flow diagram of the logic of the event scheduling approach.

The primary scheduled, or **B**-events are customers arriving to a queue, and customers finishing service. The conditional, or **C**-events are those that happen immediately after, and because of, these **B**-events. The primary ones are customers beginning service, and customers leaving the queue.

All other features of queueing systems that can be simulated with the Ciw library involve increasing the range of **B**- and **C**-events that can happen during the simulation run. In the case of customers randomly changing priority classes while waiting, one additional **B**-event and one additional **C**-event are included:

- Upon arrival to the queue customers are assigned a date in which they will change customer class, determined by randomly sampling from a distribution. Therefore each customer's event of changing customer class is scheduled for the future, and are therefore **B**-events. If those customers begin service (which might not be scheduled yet) before that event has occurred, then their changing customer class event is cancelled.
- Upon changing class, they immediately schedule another changing class event for the future, again sampling a date from a given distribution. This happens immediately after the above, and so is a **C**-event.

Note that the particular distributions used to sample class change dates in these cases are generic, and any of Ciw's currently pre-programmed distributions can be chosen, or custom distributions can also be input. For the systems described in this paper, we choose Exponential distributions with rates determined by the class change matrix Θ. Ciw allows options letting pre-empted customers resume, re-start, or re-sample their service time.

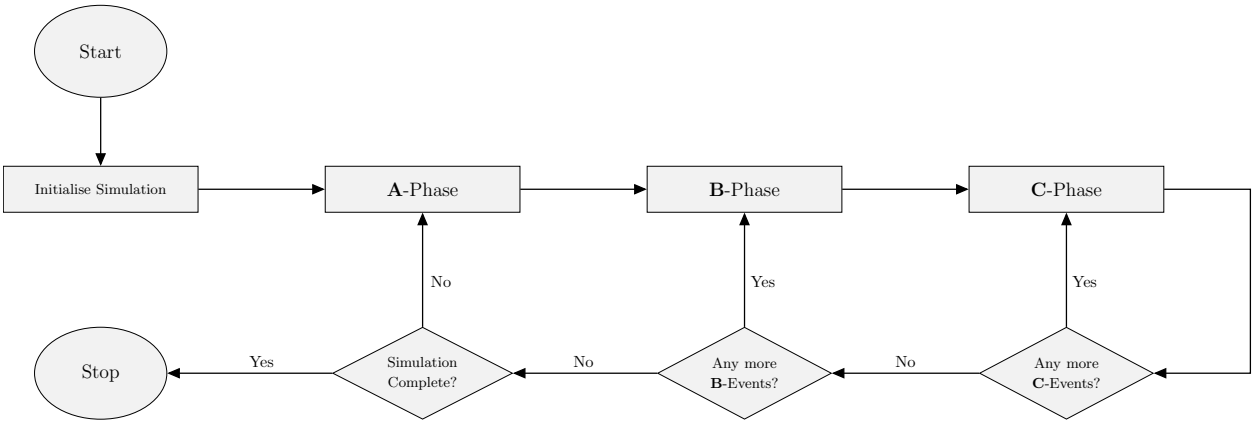


Figure 5: Flow diagram of the event scheduling approach used by Ciw, adapted from [16].

4 Markov Chain Models

The situation described in words in Section 2 can be described precisely as two different Markov chains. The first, described in Section 4.1, describes the overall changes in state, where a state records the number of customers of each class at the node. This is useful for analysing system-wide statistics such as average queue size. The second, described in Section 4.2, describes how an individual arriving customer experiences the system until their exit. This is useful for analysing individual customers' statistics such as average sojourn time. Note that as service times are Exponentially distributed, which is memoryless, the Markov chains are equivalent whether pre-empted customers resume, re-start, or re-sample their service.

4.1 Discrete State Markov Chain Formulation

Let $\underline{\mathbf{s}}_t = (s_{0,t}, s_{1,t}, \ldots, s_{K-1,t}) \in \mathbb{N}^K$ represent the state of the system at time step t, where $s_{k,t}$ represents the number of customers of class k present at time step t. Let S denote set of all states $\underline{\mathbf{s}}_t$.

The rates of change between $\underline{\mathbf{s}}_t$ and $\underline{\mathbf{s}}_{t+1}$ are given by Equation 1, where $\underline{\delta} = \underline{\mathbf{s}}_{t+1} - \underline{\mathbf{s}}_t$,

$$q_{\underline{\mathbf{s}}_t, \underline{\mathbf{s}}_{t+1}} = \begin{cases} \lambda_k & \text{if } \delta_k = 1 \text{ and } \delta_i = 0 \; \forall \; i \neq k \\ B_{k,t}\mu_k & \text{if } \delta_k = 1 \text{ and } \delta_i = 0 \; \forall \; i \neq k \text{ and } \sum_{i<k} s_{i,t} < c \\ (s_{k,t} - B_{k,t})\theta_{k_0,k_1} & \text{if } \delta_{k_0} = -1 \text{ and } \delta_{k_1} = 1 \text{ and } \delta_i = 0 \; \forall \; i \neq k_0, k_1 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

and $B_{k,t}$, representing the number of customers of class k currently in service at time step t, is given by Equation 2, where c is the number of servers.

$$B_{k,t} = \min\left(c - \min\left(\sum_{i<k} s_{i,t}, c\right), s_{k,t}\right) \tag{2}$$

Let $\pi_{\underline{\mathbf{s}}}$ denote the steady state probability of being in state $\underline{\mathbf{s}} \in S$.

4.2 Sojourn Time Markov Chain Formulation

Let $\underline{\mathbf{z}}_t = (z_{0,t}, z_{1,t}, \ldots, z_{n,t} \ldots, z_{K-1,t}, m_t, n_t) \in \mathbb{N}^{K+2} \times (1, \ldots, K-1)$ represent the state of a particular customer at time step t, where n_t represents that customer's class at time t; $z_{k,t}\ \forall\ k < n$ represents the number of customers of class k in front of the customer in the queue at time t; $z_{k,t}\ \forall\ n < k < K$ represents the number of customers of class k behind the customer in the queue at time t; and m_t represent the number of customers of class n_t behind the customer in the queue at time t. Also let $\star$ represent an absorbing state, representing the state where that customer has finished service and left the system. Let Z denote set of all states $\underline{\mathbf{z}}_t$ and $\star$.

Then the rates of change between $\underline{\mathbf{z}}_t$ and $\underline{\mathbf{z}}_{t+1}$ are given by Equation 3, where $\underline{\delta} = \underline{\mathbf{z}}_{t+1} - \underline{\mathbf{z}}_t$,

$$q_{\underline{\mathbf{z}}_t, \underline{\mathbf{z}}_{t+1}} = \begin{cases} \mu_n & \text{if } z_{t+1} = \star \text{ and } \sum_{k \le n} z_{k,t} < c \\ \lambda_n & \text{if } \delta_K = 1 \text{ and } \delta_i = 0\ \forall\ i \neq K \\ \lambda_k & \text{if } \delta_k = 1 \text{ and } \delta_i = 0\ \forall\ i \neq k \text{ and } k \neq n \\ A_{k,n,t}\mu_k & \text{if } \delta_k = -1 \text{ and } \delta_i = 0\ \forall\ i \neq k \text{ and } k < K \\ \tilde{A}_{n,t}\mu_n & \text{if } \delta_K = -1 \text{ and } \delta_i = 0\ \forall\ i \neq K \\ (z_{k_0,t} - A_{k_0,n,t})\theta_{k_0,k_1} & \text{if } \delta_{k_0} = -1 \text{ and } \delta_{k_1} = 1 \text{ and } \delta_i = 0\ \forall\ i \neq k_0, k_1 \text{ and } k_0 < K \text{ and } k_1 \neq n, K, K+1 \\ (z_{K,t} - \tilde{A}_{n,t})\theta_{n,k} & \text{if } \delta_K = -1 \text{ and } \delta_k = 1 \text{ and } \delta_i = 0\ \forall\ i \neq k, n \text{ and } k < K \\ (z_{k,t} - A_{k,n,t})\theta_{k,n} & \text{if } \delta_k = -1 \text{ and } \delta_K = 1 \text{ and } \delta_i = 0\ \forall\ i \neq k, K \\ \theta_{n,k} & \text{if } \delta_n = z_{K,t} \text{ and } \delta_K = -z_{K,t} \text{ and } \delta_{K+1} = n - k \text{ and } \delta_i = 0 \text{ otherwise, and } \sum_{k \le n} z_{k,t} < c \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

and $A_{k,n,t}$ and $\tilde{A}_{n,t}$, representing the number of customers of class k currently in service, are given by Equations 4 and 5.

$$A_{k,n,t} = \begin{cases} \min\left(c, \sum_{i \le k} z_{i,t}\right) - \min\left(c \sum_{i<k} z_{i,t}\right) & \text{if } k \le n \\ \min\left(c, \sum_{i \le k} z_{i,t} + 1 + z_{K,t}\right) - \min\left(c \sum_{i<k} z_{i,t} + 1 + z_{K,t}\right) & \text{if } n < k < K \end{cases} \tag{4}$$

$$\tilde{A}_{n,t} = \min\left(c, \sum_{i \le n} z_{i,t} + 1 + z_{K,t}\right) - \min\left(c, \sum_{i \le n} z_{i,t} + 1\right) \tag{5}$$

Let $a_{\underline{\mathbf{z}}}$ denote the expected time to absorption from state $\underline{\mathbf{z}} \in Z$.

4.2.1 Mean sojourn time calculation

The expected time to absorption can be calculated from each state. Customers arrive in all states where $z_K = 0$, and their class can be determined by n. First define $\tilde{Z} = \{\underline{\mathbf{z}} \in Z \setminus \{\star\} \mid z_K = m = 0\} \subset Z$ as the set of all states where the newly arriving customer can can arrive. Let $c : \tilde{Z} \to S$ be a map between states in $\tilde{Z}$ and S, given in Equation 6.

$$c\left(\underline{\mathbf{z}} = (z_0, z_1, \ldots, z_{K-1}, m, n)\right) = (z_0, z_1, \ldots, z_{K-1}) \tag{6}$$

Note that c is a surjective map, but not injective. In fact, for every element $\underline{\mathbf{s}} \in S$ exactly K states in $\tilde{Z}$ map to it. These correspond to states at which each of the K classes of customer can arrive. In each of

these states, the probability of a customer from class k arriving is $\frac{\lambda_k}{\sum_{i=0}^{K-1}\lambda_i}$. Therefore, we can combine this to get the overall mean sojourn time in Equation 7:

$$\overline{\Psi} = \sum_{\underline{\mathbf{z}}\in\tilde{Z}} \sum_{k=0}^{K-1} \frac{\lambda_k}{\sum_{i=0}^{K-1}\lambda_i} \pi_{c(\underline{\mathbf{z}})} a_{\underline{\mathbf{z}}} \tag{7}$$

Defining $\tilde{Z}_k = \{\underline{\mathbf{z}} \in \tilde{Z} \mid z_{K-1} = n = k\} \subset Z$ as the states that customers of class k arrive to, the mean sojourn times for customers who arrive as a given customer class k is given by Equation 8.

$$\Psi_k = \sum_{\underline{\mathbf{z}}\in\tilde{Z}_k} \pi_{c(\underline{\mathbf{z}})} a_{\underline{\mathbf{z}}} \tag{8}$$

4.3 Bounded Approximation

In order to analyse the above Markov chain models numerically, finite approximations can be made. Let $b \in \mathbb{N}$, and define the b-bounded version of the infinite queueing system described in Section 2, such that the maximum allowed number of customers of each priority class is b, and customer losses when that number is exceeded. The equivalent b-bounded Markov chains associated with this system are identical to those described in Sections 4.1 and 4.2 except with bounded state spaces $\underline{\mathbf{s}}_t = (s_{0,t}, s_{1,t}, \ldots, s_{K-1,t}) \in (0, 1, \ldots, b)^K$ and $\underline{\mathbf{z}}_t = (z_{0,t}, z_{1,t}, \ldots, z_{n,t} \ldots, z_{K-1,t}, m_t, n_t) \in (0, 1, \ldots, b)^{K+2}$ respectively. These Markov chains are finite and so stationary.

If the unbounded system is stationary, that is the system reaches steady state and has steady state probabilities $\underline{\pi}$, then the steady states of the b-bounded system, $\tilde{\underline{\pi}}$ is an approximation of $\underline{\pi}$. As b increases the probability of the number of customers of a particular customer class in the unbounded system exceeding b approaches zero as b increases. Therefore as b increases the b-bounded system becomes a better and better approximation of the unbounded system.

Choosing an appropriate value for b is a trade off between accuracy and model size, and so computational time. One way to choose b would be to sequentially build bounded models, increasing b each time, calculating the statistics of interest, and observing when the relationship between b and that statistic levels off. This is shown in Figures 6 and 7, which show that for a particular system (two customer classes, $\lambda_1 = \frac{2}{3}$, $\lambda_2 = \frac{1}{3}$, $\mu_1 = \frac{3}{2}$, $\mu_2 = \frac{5}{2}$, $\theta_{12} = 3$, $\theta_{21} = 1$, $c = 1$), the expected number of customers of each class in the simulation, and the expected sojourn time for each class, approaches that found using a long run simulation (400,000 simulated time units with a warmup and cooldown time of 3000 time units) as b increases.

This is an inefficient way of determining the accuracy of the bounded approximation. It would be more efficient to choose a b and be able to immediately measure if the accuracy is sufficient. We propose two measures, one for the ergodic Markov chains of Section 4.1, and one for the absorbing Markov chains of Section 4.2).

4.3.1 Accuracy measure for the ergodic Markov chain

Let $S_b = \{\underline{\mathbf{s}} \in S \mid b \in \underline{\mathbf{s}}\}$, the set of states that lie on the Markov chain boundary. We wish to choose b large enough that the boundary is irrelevant, that is that the Markov chain hardly ever reaches the boundary. Therefore we propose the *relative* probability of being at the boundary, $\mathcal{Q}(b)$, to be a measure of accuracy; if this is sufficiently small, then the bound b is large enough. This is given in Equation 9, it is the ratio of the probability of being at the boundary in the b-bounded system, and the probability of being at the boundary if every state was equally likely. This is necessary as the larger b is, the larger the state space is, meaning

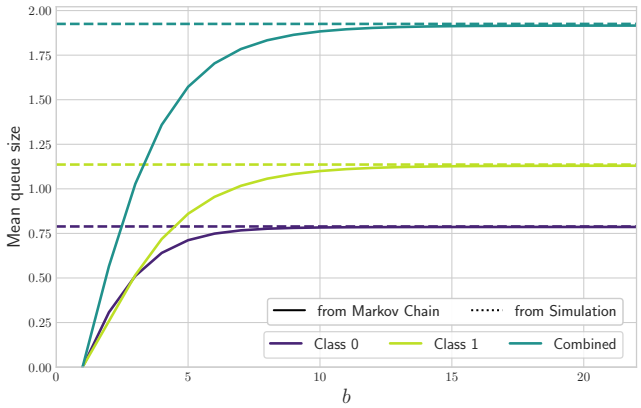


Figure 6: Demonstrating that as b increases, the expected number of customers of each class approaches that found using a long run simulation.

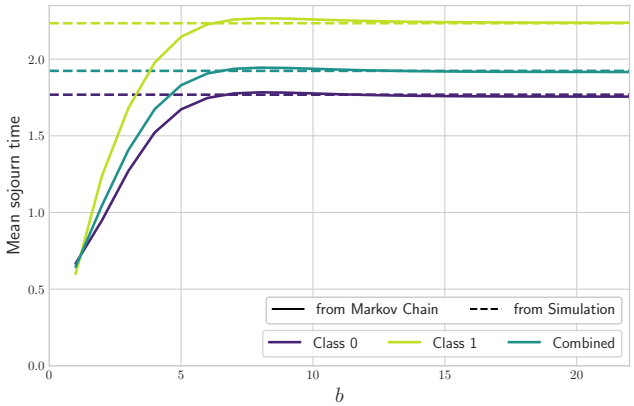


Figure 7: Demonstrating that as b increases, the expected sojourn time of customers of each class approaches that found using a long run simulation.

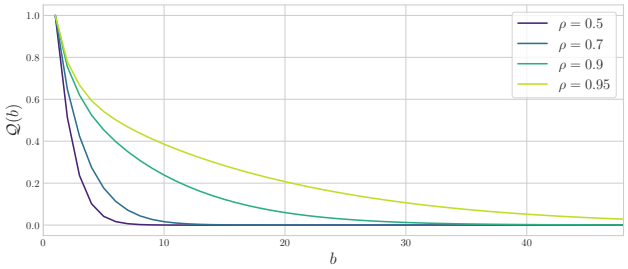


Figure 8: Demonstration of the effect of b on $\mathcal{Q}(b)$

that the steady state probabilities are spread over more states and so are not comparable alone, whereas the relative probability of being at the boundary is comparable over different sizes of b.

$$\mathcal{Q}(b) = \frac{|S|}{|S_b|} \sum_{s \in S_b} \tilde{\pi}_s \tag{9}$$

To demonstrate the effect of b on $\mathcal{Q}(b)$ under different systems, consider the dynamic priorities system with two customer classes, $\lambda_1 = \frac{1}{2}$, $\lambda_2 = \frac{1}{2}$, $c = 1$, $\mu_1 = \frac{1}{\rho}$, $\mu_2 = \frac{1}{\rho}$, $\theta_{12} = 1$, and $\theta_{21} = 1$; where $0 < \rho < 1$ is some given traffic intensity. Figure 8 shows the effect of b on $\mathcal{Q}(b)$ for this system, for different values of ρ. In all cases as b increases, $\mathcal{Q}(b)$ decreases, indicating greater accuracy of the bounded system. As expected, as ρ increases, we expect more customers in the queue, and so the boundary b needs to be much larger before it can be considered irrelevant.

4.3.2 Accuracy measure for the absorbing Markov chain

The above check isn't possible for absorbing Markov chains as they will not reach steady state, so another check is required. Define $h_{i,J}$ as the hitting probabilities of a set of states J from state i, that is, what is the probability of ever reaching any state in J when starting from state i. These are defined recursively by Equation 10, where $q_{i,k}$ is the transition rate from state i to state j defined in Equation 3 [18].

$$h_{i,J} = \begin{cases} \sum_k q_{i,k} h_{k,J} & \text{if } i \notin J \\ 1 & \text{if } i \in J \end{cases} \tag{10}$$

Relating this to the absorbing Markov chain described in Section 4.2, and letting $Z_b \subset Z$ be the set of boundary states such that $Z_b = \{\underline{\mathbf{z}} \in Z \mid b \in \underline{\mathbf{z}}\}$, then if a customer arrives to state i, the probability of that customer's state reaching the boundary is h_{i,S_b}. Therefore we propose the probability of an arriving customer experiencing the boundary, $\mathcal{P}(b)$, to be a measure of accuracy; if this is sufficiently small, then the bound b is large enough. This is calculated in a similar way to the mean sojourn time in Section 4.2.1, and given in Equation 11.

$$\mathcal{P}(b) = \sum_{\underline{\mathbf{z}} \in \tilde{Z}} \sum_{k=0}^{K-1} \frac{\lambda_k}{\sum_{i=0}^{K-1} \lambda_i} \pi_{c(\underline{\mathbf{z}})} h_{\underline{\mathbf{z}}, Z_b} \tag{11}$$

To demonstrate the effect of b on $\mathcal{P}(b)$ under different system, consider the same dynamic priorities system with two customer classes used in the previous demonstration. Figure 9 shows the effect of b on $\mathcal{P}(b)$ for this system, for different values of ρ. Again, in all cases as b increases, $\mathcal{P}(b)$ decreases, indicating greater

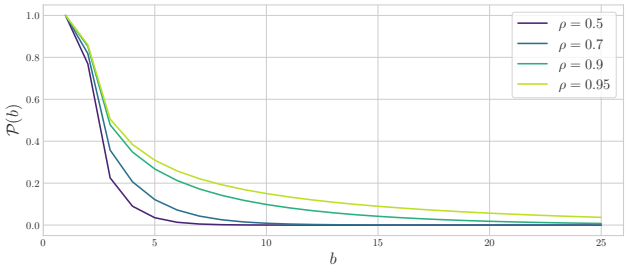


Figure 9: Demonstration of the effect of b on $\mathcal{P}(b)$

accuracy of the bounded system; and similarly as ρ increases the boundary b needs to be larger before it can be considered irrelevant.

4.4 Existence of Stationary Distributions

It is useful to know whether a dynamic priorities system is stationary, that is, will it reach steady state or not. Proposition 1 gives a naive check for the existence or non-existence of steady states, but does not cover all possibilities.

Proposition 1. *For an $M/M/c$ work conserving queue with K classes of customer, with arrival rate and service rate λ_k and μ_k for customers of class k, respectively; then*

1. *it will reach steady state if $\rho_{\max} = \frac{\sum_i \lambda_i}{c \min_i \mu_i} < 1$,*
2. *it will never reach steady state if $\rho_{\min} = \frac{\sum_i \lambda_i}{c \max_i \mu_i} \geq 1$.*

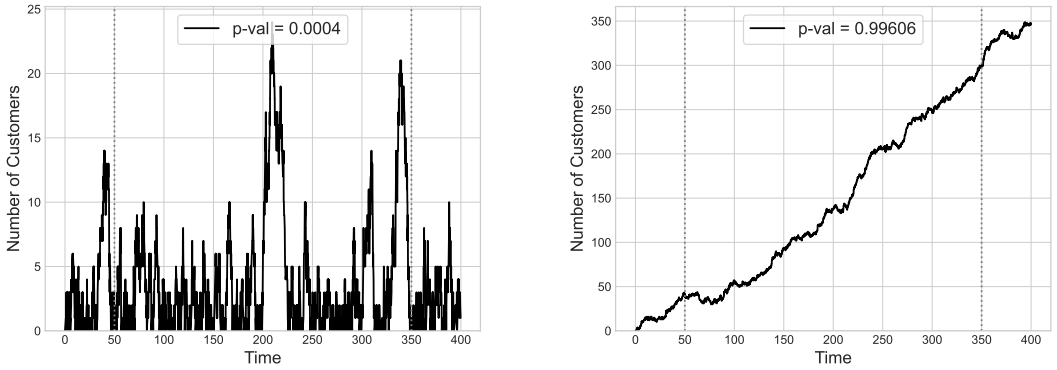
Note that this result does not assume any particular service discipline such as first-in-first-out or prioritised classes, but holds for any work conserving discipline.

Proof. The queue will reach steady state if the rate at which customers are added to the queue is less than the rate at which customers leave the queue. As arrivals are not state dependent, customers are added to the queue at a rate $\sum_i \lambda_i$ when in any state. The rate at which customers leave the queue is state dependent, depending on the service discipline.

We do not need to consider cases when there are less than c customers present, as here any new arrival will increase the rate at which customers leave the queue, as that arrival would enter service immediately. Considering the cases where there are c or more customers in the queue, there are two extreme cases, either:

1. all customers in service are of the class with the slowest service rate. In this case the rate at which customers leave the queue is $c \min_i \mu_i$, which is the slowest possible rate at which customers can leave the queue. If $\sum_i \lambda_i < c \min_i \mu_i$ then the rate at which customers enter the queue is smaller than the smallest possible rate at which customers leave the queue, and so will always be smaller than the rate at which customers leave the queue in all states. Therefore the system will reach steady state. Or:
2. all customers in service are of the class with the fastest service rate. In this case the rate at which customers leave the queue is $c \max_i \mu_i$, which is the fastest possible rate at which customers can leave the queue. If $\sum_i \lambda_i \geq c \max_i \mu_i$ then the rate at which customers enter the queue is greater than or equal to the largest possible rate at which customers leave the queue, and so will always be greater or equal to than the rate at which customers leave the queue in all states. Therefore the system cannot reach steady state.

□



(a) State time series for Example 1.

(b) State time series for Example 2.

Figure 10: Demonstration of the ADF test on states that do and do not reach steady state according to Proposition 1.

If $c \min_i \mu_i \leq \sum_i \lambda_i < c \max_i \mu_i$ then more investigation is needed. In the case of dynamic priority classes the class change matrix Θ may be significant. For example the service rate of customers of one class may be very slow, however if the rate at which customers leave that class is sufficiently large then that service rate may not have an effect. Alternatively if the rate at which customers of the other classes change to that class is large, then that slow service rate could be a bottleneck for the system.

We can however approximately test if a system is stationary or not using simulation. Consider the time series $x(t)$, representing the total number of customers in the system at time t. In Ciw, this can be empirically recorded using a state tracker object. If the system reaches steady state, then the $x(t)$ will be stochastic with non-increasing trend, therefore it would be a stationary time series. Conversely, if the system does not reach steady state, then $x(t)$ will be stochastic with increasing trend, therefore it would be a non-stationary time series. The Augmented Dicky-Fuller (ADF) test [2] tests for the non-stationarity of a stochastic time series, and so can be utilised here to test if a simulation has reached steady state or not. Note here that the time series $x(t)$ recorded by Ciw has irregular gaps (time stamps are the discrete time points where a customer arrives or leaves the system), and the ADF test requires regularly spaced time stamps; therefore the Traces library [13] is used to take regularly-spaced moving averages before the hypothesis test is undertaken.

To demonstrate this, consider two examples:

- Example 1, that is guaranteed to reach steady state by Proposition 1: $\lambda_1 = 2$, $\lambda_2 = 1$, $\mu_1 = 4$, $\mu_2 = 4$, $\theta_{12} = 1$, $\theta_{21} = 1$, $c = 1$;
- Example 2, that is guaranteed not to reach steady state by Proposition 1: $\lambda_1 = 2$, $\lambda_2 = 1$, $\mu_1 = 1$, $\mu_2 = 1$, $\theta_{12} = 1$, $\theta_{21} = 1$, $c = 2$.

Figures 10a and 10b shows their state time series' $x(t)$ respectively. It is clear that the state time series for Example 1 is stationary, and the state time series for Example 2 is non-stationary and increasing. When performing the ADF test on these, Example 1 gives a p-value of 0.0004, rejecting the null hypothesis that the time series is non-stationary, while Example 2 gives a p-value of 0.9961, and the null hypothesis cannot be rejected.

There is a gap in Proposition 1 for systems where $c \min_i \mu_i \leq \sum_i \lambda_i < c \max_i \mu_i$. This is where the dynamic priority classes can influence the stationarity of the system. Consider a two class system with $\lambda_1 = 2$, $\lambda_2 = 2$, $c = 1$. For the service rates of each customer class, consider two cases:

- $\mu_1 = 3$ and $\mu_2 = 5$: here $\rho_{\max} > 1$ and $\rho_{\min} < 1$, and the prioritised class receive a slower service rate;

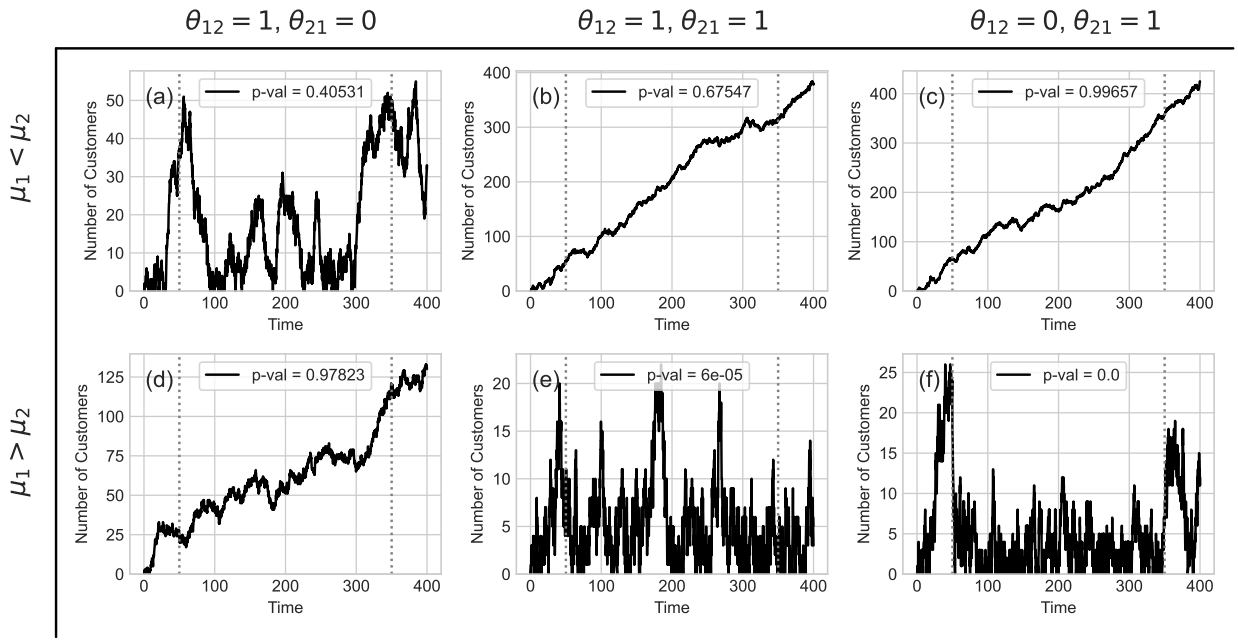


Figure 11: Investigating the stationarity under six cases not covered by Proposition 1.

- $\mu_1 = 5$ and $\mu_2 = 3$: here $\rho_{\max} > 1$ and $\rho_{\min} < 1$, and the prioritised class receive a faster service rate.

In each of these cases, we can consider three other cases pertaining to the class change rate matrix Θ:

- $\theta_{12} = 1$ and $\theta_{21} = 0$: downgrades but no upgrades;
- $\theta_{12} = 1$ and $\theta_{21} = 1$: bother downgrades and upgrades;
- $\theta_{12} = 0$ and $\theta_{21} = 1$: upgrades but no upgrades.

All these cases are not covered by Proposition 1, so we experimentally investigate their stationarity using the Ciw simulation and ADF test. Figure 11 shows the results. Here we see that three of the six cases are stationary, (a), (e), and (f), while the others are not. In all three we see that there is possibility of a customer from the class with the slower service rate transitioning to a class with the quicker service rate. In two of the non-stationary cases, (c) and (d), customers with the slower service rate have no possibility of transitioning out of their class, and so the queue builds up indefinitely. It is interesting to compare cases (b) and (e), in which both customer classes can transition to the other customer class. Here one case is stationary, and the other is non-stationary, with the only difference being whether the prioritised class has the quicker service rate or not. This evidences the interesting interplay between service rate, priority class, and class change rates.

5 Modelling Unknown Service Disciplines

The dynamic priority classes model presented in Section 2 may be used to model situations where FIFO is not appropriate, but specific service disciplines may not be known. Recall now the motivating example discussed in Section 1.1, where it was established that FIFO was not appropriate but service discipline unknown, here we attempt to use the dynamic priorities setup to model the endoscopies waiting list.

According to the observed data referrals roughly follow a Poisson distribution with rate 0.463 a day. Service rate data is estimated to be around 0.5 endoscopy procedures a day. To determine the appropriateness of the

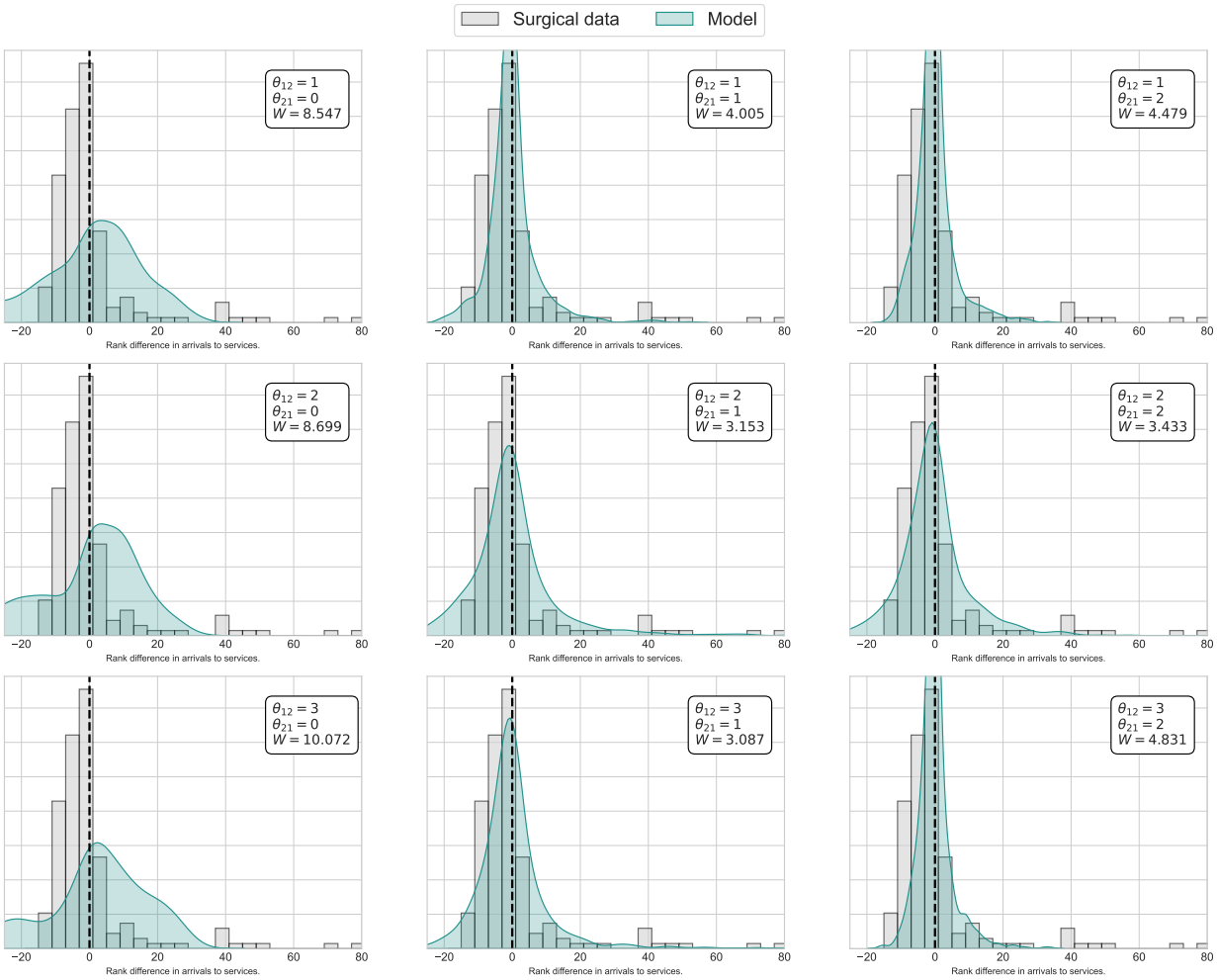


Figure 12: Comparison between simulated and observed overtakes for different values of θ_{12} and θ_{21}.

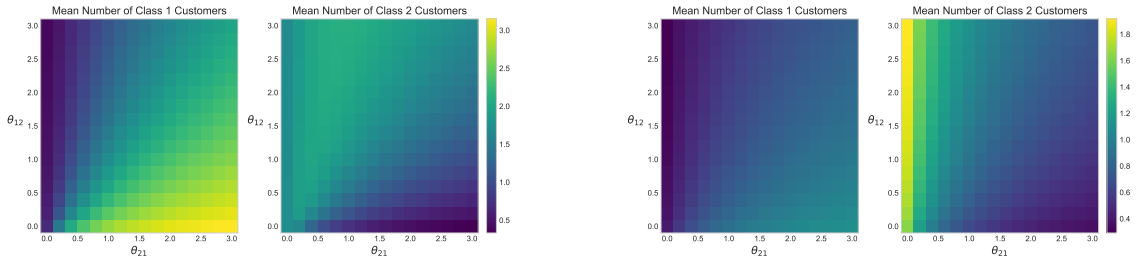
dynamic priority classes model here, we model the system as a two class system, with $\lambda_1 = 0.463$ and $\lambda_2 = 0$, and $\mu_1 = \mu_2 = 0.5$ and $c = 1$, that is all referrals are of the most urgent, with customers able to be downgraded and then de-downgraded during their waiting time. We simulate this system under different parameters of θ_{12} and θ_{21}, observe one year of referrals, over 5 trials, and compare the distribution of rank differences, or overtakes, with that of the original system with indeterminate service discipline. The Wasserstein distance metric [14] between the modelled and actual distributions is calculated to measure the models' accuracies in approximating the indeterminate service discipline. Figure 12 compares the modelled and actual distributions of net rank difference, along with the Wasserstein metric W, for all pairs $(\theta_{12}, \theta_{21}) \in \{1, 2, 3\} \times \{0, 1, 2\}$. From this it can be seen that a combination of downgrades and upgrades is required to fit a good distribution of overtakes, and of the parameters tested $\theta_{12} = 3$, $\theta_{21} = 1$ produces the best fit. This indicates that, with further parameter tuning, these dynamic priority classes models can be used to model unknown service disciplines.

5.1 Effect of Θ on Customer Experience

Now that it is established that some service disciplines can be modelled as dynamically changing priorities with a class change matrix Θ, we want to investigate the effect of this matrix on customer experience. This

Scenario	λ_1	λ_2	c	μ_1	μ_2
A	1	1	1	$5/2$	$7/2$
B	1	1	1	$7/2$	$5/2$

Table 1: Parameters used in experiments that investigate the effect of Θ.



(a) Scenario A.

(b) Scenario B.

Figure 13: Average number of customers of each class, as Θ changes.

would be important for the modelling process, and also for controllers of the system who might be able to influence the rates and improve customer experience or system outcomes.

We first define two scenarios that we will use to investigate the effect of Θ, defined in Table 1. Both have two classes of customer, in Scenario A prioritised customers have a slower service rate than unprioritised customer, while in Scenario B prioritised customers have a faster service rate than prioritised customers. The Markov chain models of Section 4 are build with these parameters, and all values of θ_{12} and θ_{21} from 0 to 3, in steps of 0.2, with a bound of 16 (all producing accuracy measures $\mathcal{Q}(16), \mathcal{P}(16) < 0.012$), and customer experience KPIs are found and compared.

Figures 13a and 13b give the steady-state average number of customers of each class, for all pairs of θ_{12} and θ_{21}, under Scenarios A and B respectively. In general it can be seen that as θ_{12} increases in comparison to θ_{21} then we expect less customers of class 1 and more of class 2, and the opposite is true as θ_{21} increases in comparison to θ_{12}. At first it seems that when $\theta_{12} \approx \theta_{21}$ the magnitude of the rate does not have a big effect of the number of customers of each class present, however further investigation shows this not to be the case. Figure 14 shows the mean number of class 1 and class 2 customers when $\theta_{12} = \theta_{21} = \tilde{\theta}$, under Scenarios A and B, as the magnitude $\tilde{\theta}$ changes. In Scenario A, where prioritised customers have a slower service rate, increasing the priority change rate increases the prioritised customers present, and decreases the number of unprioritised customers. In Scenario B the opposite trend it true. Looking at the effect of $\tilde{\theta}$ on the overall number of customers present, in Figure 15, we see that a higher rate of priority changes increases the number of customers in Scenario A, but decreases the number of customers in Scenario B. This may be because as the rate of priority changes increases, each customer is more likely to be served as a prioritised customer, and in Scenario B prioritised customers are processed quicker.

Figures 16a and 16b give the steady-state average sojourn time of customers beginning in each class, for all pairs of θ_{12} and θ_{21}, under Scenarios A and B respectively. The mean sojourn time for class 1 customers are hardly effected by the size of θ_{12}, but is more effected by the size of θ_{21}, likely due to an increased number of class 1 customers present when they join the queue. The effect of θ_{21} on the mean sojourn time of class 2 customers depends on the scenario: in Scenario A the higher θ_{21}, the more likely a class 2 customer is to be served as a class 1 customer, that with the slower service rate, and so a longer sojourn time; while in Scenario B they are more likely to be served as a class 1 customer with higher service rate, and so shorter sojourn time.

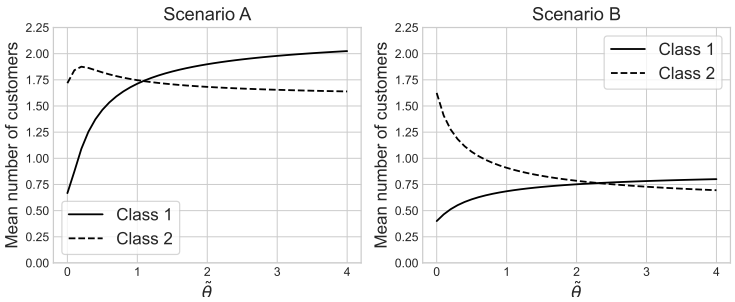


Figure 14: Average number of customers of each class, as $\tilde{\theta}$ changes.

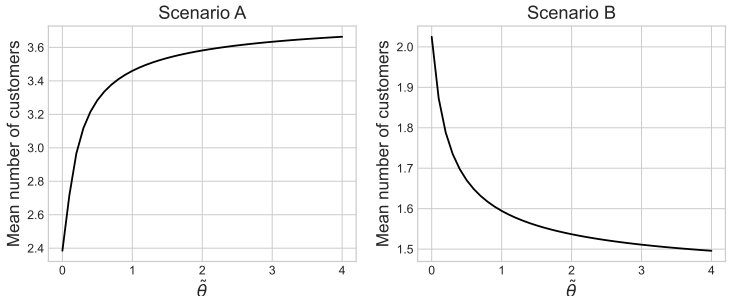
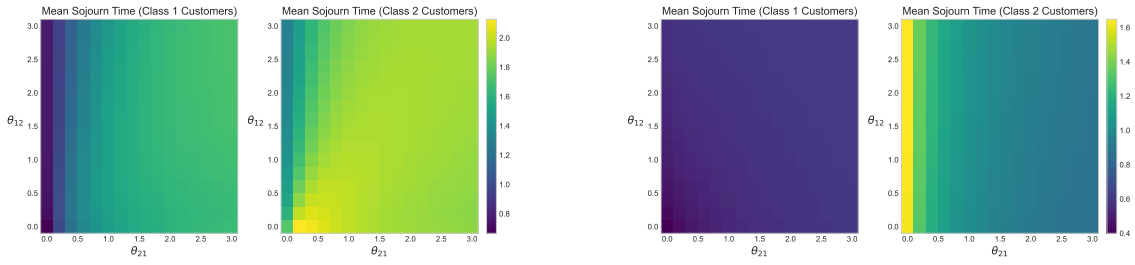


Figure 15: Average number of customers overall as $\tilde{\theta}$ changes.



(a) Scenario A.

(b) Scenario B.

Figure 16: Average sojourn time for customers beginning in each class, as Θ changes.

6 Conclusions

In this paper a generalised model of dynamically changing priorities within an M/M/c queue is presented. The general formulation was given in Section 2, while two methodologies for modelling it was given, first by simulation, with contributions to the Ciw library, and second by two separate Markov chains used in conjunction to find state probabilities and customer sojourn times. In order to use the Markov chains we present bounded approximations, and importantly we present two accuracy measures to immediately determine how well the bounded Markov chain approximates its infinite version: $\mathcal{Q}(b)$ the relative probability steady state probability of being at the boundary, and $\mathcal{P}(b)$ the probability of an arriving customer experiencing the boundary.

This work is motivated by modelling a healthcare situation, a surgical endoscopy waiting list, where modelling as FIFO is inappropriate. We show that modelling as dynamically changing priorities can approximate the queue behaviour somewhat, with a sufficient choice of class change rate matrix Θ. We then explore the effect of this matrix on customer experience, namely mean number of customers of each priority in the system, and mean sojourn time for each customer class. This exploration may be useful is queue controllers, that is waiting list managers, can influence or tweak the class change matrix - for example if prioritised customers have a faster service rate, then the queue is manages better when the priority change rate is higher, though at the expense of the prioritised customers themselves.

Although much of the work in this paper concentrated on two classes of customer and two prioritisation levels, the formulation is generalised to any number of customer classes, offering greater flexibility in modelling unknown service disciplines. Similarly, the simulation methodology, implemented and available out-of-the-box in an open-source Python package, is generalised and can model non-Markovian priority changes too. For example a deterministic distribution, that is one that samples the same number each time, is equivalent to a time cut-off for priority changes.

All code and computational work used to produce this paper is openly available at `https://github.com/geraintpalmer/DynamicClasses`.

References

[1] Blair Bilodeau and David A Stanford. High-priority expected waiting times in the delayed accumulating priority queue with applications to health care kpis. *INFOR: Information Systems and Operational Research*, pages 1–30, 2022.

[2] David A. Dickey and Wayne A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366):427–431, 1979.

[3] Yichuan Ding, Eric Park, Mahesh Nagarajan, and Eric Grafstein. Patient prioritization in emergency department triage systems: An empirical study of the canadian triage and acuity scale (ctas). *Manufacturing & Service Operations Management*, 21(4):723–741, 2019.

[4] Douglas G Down and Mark E Lewis. The n-network model with upgrades. *Probability in the Engineering and Informational Sciences*, 24(2):171–200, 2010.

[5] Alexander Dudin, Olga Dudina, Sergei Dudin, and Konstantin Samouylov. Analysis of single-server multi-class queue with unreliable service, batch correlated arrivals, customers impatience, and dynamical change of priorities. *Mathematics*, 9(11):1257, 2021.

[6] Stephen S Fratini. Analysis of a dynamic priority queue. *Stochastic Models*, 6(3):415–444, 1990.

[7] Qi-Ming He, Jingui Xie, and Xiaobo Zhao. Priority queue with customer upgrades. *Naval Research Logistics (NRL)*, 59(5):362–375, 2012.

[8] JM Holtzman. Bounds for a dynamic-priority queue. *Operations Research*, 19(2):461–468, 1971.

[9] James R Jackson. Some problems in queueing with dynamic priorities. *Naval Research Logistics Quarterly*, 7(3):235–249, 1960.

[10] Leonard Kleinrock. A delay dependent queue discipline. *Naval Research Logistics Quarterly*, 11(3-4):329–341, 1964.

[11] Valentina Klimenok, Alexander Dudin, Olga Dudina, and Irina Kochetkova. Queuing system with two types of customers and dynamic change of a priority. *Mathematics*, 8(5):824, 2020.

[12] Charles Knessl, Charles Tier, and Doo Il Choi. A dynamic priority queue model for simultaneous service of two traffic types. *SIAM Journal on Applied Mathematics*, 63(2):398–422, 2003.

[13] The Traces library developers. Traces. `https://traces.readthedocs.io/en/master/api_reference.html`, 2023.

[14] Hamidreza Mostafaei and Shaghayegh Kordnourie. Probability metrics and their applications. *Applied Mathematical Sciences*, 5(4):181–192, 2011.

[15] A Netterman and I Adiri. A dynamic priority queue with general concave priority functions. *Operations Research*, 27(6):1088–1100, 1979.

[16] Geraint Palmer. *Modelling deadlock in queueing systems*. PhD thesis, Cardiff University, 2018.

[17] Geraint I Palmer, Vincent A Knight, Paul R Harper, and Asyl L Hawa. Ciw: An open-source discrete event simulation library. *Journal of Simulation*, 13(1):68–82, 2019.

[18] Nicolas Privault. Understanding markov chains. *Examples and Applications, Publisher Springer-Verlag Singapore*, 357:358, 2013.

[19] Stewart Robinson. *Simulation: the practice of model development and use.* Palgrave Macmillan, 2014.

[20] Azaz Bin Sharif, David A Stanford, Peter Taylor, and Ilze Ziedins. A multi-class multi-server accumulating priority queue with application to health care. *Operations Research for Health Care*, 3(2):73–79, 2014.

[21] Wayne E Smith et al. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66, 1956.

[22] David A Stanford, Peter Taylor, and Ilze Ziedins. Waiting time distributions in the accumulating priority queue. *Queueing Systems*, 77:297–330, 2014.

[23] Jan A Van Mieghem. Dynamic scheduling with convex delay costs: The generalized c— mu rule. *The Annals of Applied Probability*, pages 809–833, 1995.

[24] Jingui Xie, Qi-Ming He, and Xiaobo Zhao. Stability of a priority queueing system with customer transfers. *Operations Research Letters*, 36(6):705–709, 2008.