Journal of the Operational Research Society

ISSN: 0160-5682 (Print) 1476-9360 (Online) Journal homepage: www.tandfonline.com/journals/tjor20

Queues under stochastic priority switching

Geraint Ian Palmer, Michalis Panayidis, Vincent Knight & Elizabeth Williams

To cite this article: Geraint Ian Palmer, Michalis Panayidis, Vincent Knight & Elizabeth Williams (05 Jul 2025): Queues under stochastic priority switching, Journal of the Operational Research Society, DOI: 10.1080/01605682.2025.2525939

To link to this article: https://doi.org/10.1080/01605682.2025.2525939

Published online: 05 Jul 2025.

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
https://doi.org/10.1080/01605682.2025.2525939

RESEARCH ARTICLE

OPEN ACCESS Check for updates

Queues under stochastic priority switching

Geraint Ian Palmer, Michalis Panayidis, Vincent Knight and Elizabeth Williams

School of Mathematics, Cardiff University, Cardiff, UK

ABSTRACT

In this paper a generalised model of dynamic and stochastic changing priorities within an $M/M/c$ queue is presented. Simulation and Markov chain models are given that describe 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 a waiting list for surgical endoscopy 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.

ARTICLE HISTORY
Received 9 September 2024
Accepted 24 June 2025

KEYWORDS
Queueing; Markov chains; simulation

1. Introduction

There are a several situations in which a customer's level of urgency in a queue might change while they wait, 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 increases the longer they spend waiting due to health degeneration (Bradford Delong et al., 2008; Garbuz et al., 2006; Williams et al., 2020). Another example would be a prioritisation scheme that attempts a trade-off between medical need and waiting times (Powers et al., 2023). These are both examples where a patient's priority has the chance to upgrade over time while in the queue. Here we precisely define a customers' priority as an ordinal measure, with lower priority customers only being served once all higher priority customers in the system have been served. These ordinal priority measures are usually integers, and are usually attributed to whole classes of customers (Stewart, 2009).

There also might be situations in which a patient's priority can downgrade over time: consider waiting for some medical intervention that can improve a patient's outcome only 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 (D'Alessandro et al., 2017).

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 and downgrades 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 (Palmer et al., 2019) and is a contribution of this paper. An important question arises from this, when does a steady state distribution exist for such a queueing system? To answer this question, two Markov chain models are defined, 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 simulation logic required and the contribution to the Ciw library; then experimentally justifies the use of these models to model scenarios where prioritisation rules are unknown. 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, presenting measures of accuracy for these bounded approximations, and discussing the

CONTACT Geraint Ian Palmer palmergi1@cardiff.ac.uk School of Mathematics, Cardiff University, Cardiff, UK

existence or otherwise of systems that can reach steady state. An exploration of the system behaviour under different parameters is given.

1.1. Motivating case study: Surgical endoscopy

Consider the queueing process for a particular surgical procedure. In many cases the service discipline, that is the rules for how patients chosen from the waiting list to be scheduled for surgery, are not known to modellers, and may even be impossible to know as managers may be reluctant to disclose the information for political or other reasons. There have been efforts to understand such unknown service disciplines, for example in Ding et al. (2019), which shows 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. They attempt to fit prioritisation rules to this discretionary behaviour. Here we will do the same, utilising a stochastic priority switching configuration to emulate the observed service disciplines. We consider and investigate the waiting list of a surgical endoscopy procedure carried out in a Welsh health board over the period 1 January to 31 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 which will be referred to as being "bumped down" the queue. A positive number indicates that they overtook more people, and were "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, this indicates that patients have priorities that change over time. The figure also reports $\omega_{\uparrow}$ and $\omega_{\downarrow}$, the average waiting time for patients that were bumped up and down the list, respectively.

1.2. Related work

Queue where customers' priority changes as they wait or progress through the system have been studies previously. One of the first papers to consider such a system was Jackson (1960), which considers a customer's waiting time and their urgency number when deciding which customer to be served next from the queue; the difference between the waiting time and urgency number determines the customer's priority, which increases linearly over time. This is analysed by considering event probabilities at clock ticks. This is re-analysed and extended in Kleinrock (1964), a fundamental paper on dynamic priorities, also called "delay dependent," "time dependent" and "accumulating" priorities. Here both pre-emptive and non-pre-emptive priorities are considered (Ferrand et al., 2018). applies this to a simulation of an emergency department in a children's hospital, and shows that these accumulating priorities lead to more efficient resource use.

In Holtzman (1971) the original model by Jackson (1960) is extended by treating different urgency numbers as separate customer classes, with certain restrictions on the ordering of parameters on the linear priority functions. Bounds on the customers' waiting times are found. Later Bagchi and Sullivan (1985), extends the work of all of the above by considering cases where the multiple classes of customer have less restrictive orderings of the linear priority functions, and Sharma and Sharma (1994) derives its expected steady state waiting times (Stanford et al., 2014). furthers the work of Kleinrock (1964) to look at the maximum priority of the waiting customers in a single server queue,

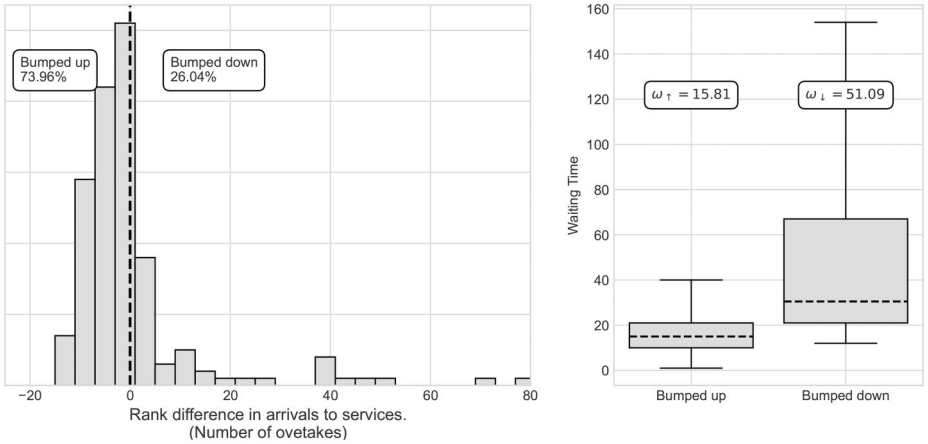


Figure 1. Distribution of overtakes in the endoscopy waiting list.

that is the next customer to be served, as a stochastic process termed an accumulating priority queue (APQ). This is extended in Sharif et al. (2014) to multi-server queues, and in Li and Stanford (2016) to account for heterogeneous servers, and in Kella and Ravner (2017) where waiting time distributions are derived. In Bilodeau and Stanford (2022) truncated analytical expressions are given for a two class delayed accumulating priority queue with general service time distributions, where customers experience an initial fixed delay before service.

Non-linear, concave accumulating priority functions are considered in Netterman and Adiri (1979) in place of the linearly accumulating priorities considered before. While in Li et al. (2017) equivalences between some families on non-linear accumulation functions and linearly increasing ones are given.

Another way in which modellers have considered dynamically changing classes is by static priorities within a customer class, but dynamically switching class. This is the model considered in this paper. In Fratini (1990) a non-pre-emptive $M/G/1$ closed loss queue with two classes of customers is considered, where priorities switch if the number from one class exceeds a given threshold. Similarly, Knessl et al. (2003) extends this to infinite waiting capacities for both customers in the case of Markovian services.

Stochastic class switching has also been studied. The logic for simulating a finite population queue with randomly increasing priority numbers is described in Panayiotopoulos (1980), where times between priority increases are randomly generated; while Grindlay (1965) simulation experiments are run where for series of queues in tandem where a customer's urgency number adapts according to the amount of time already spent in the system. In Xhafa and Tonguz (2001) calls in personal communication systems are modelled by considering a system with three priority classes, the lower priority class is lost to the system when servers are busy, while the other two classes experience non-pre-emptive priorities, and exponentially distributed upgrades from the middle priority to the highest priority. That is the time for a customer to upgrade is exponentially distributed. In Down and Lewis (2010) a pre-emptive $M/M/c$ priority queue with exponential upgrades for two classes of customer is considered with holding costs, with some state dependent restrictions on the upgrades. In Xie et al. (2008) the pre-emptive $M/M/c$ priority queue with exponential upgrades is studied for an arbitrary amount of customer classes; however customers can only upgrade to the priority immediately higher than themselves (He et al., 2012). extends this to allow batch arrivals and phase-type upgrades, but again, customers only upgrade to the priority immediately higher than themselves. Exponential downgrades are modelled in Klimenok et al. (2020) alongside upgrades, but limited to single server system with finite queueing capacity. This is extended in Lee et al. (2020) to include multiple priority classes but without downgrades, and in Dudin et al. (2021) to include unreliable services and impatient customers. We contribute to the literature by generalising these models to include upgrades and downgrades to any other priority class, introducing both a simulation and Markov chain models to describe these behaviours.

Other configurations of dynamically changing priority classes have also been considered. For example, Adiri (1971) introduces a model where customers are de-prioritised during their service if their service time exceeds a given minimum time interval, or quantum of time. Customers are downgraded and made to wait again for service, behind newly arriving and newly downgraded customers. Van Mieghem (1995) introduces the generalised $c\mu$-rule, first conceived in Smith (1956), which applies a cost to each customer which is dependent on both their class and their waiting time. This cost then acts as a scheduling rule, but can also model changing priorities amongst customers.

2. An $M/M/c$ queue with stochastic priority switching

In this section, we present the detailed description of a queueing model where customers can stochastically switch priority classes while waiting. 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" (moving to a priority

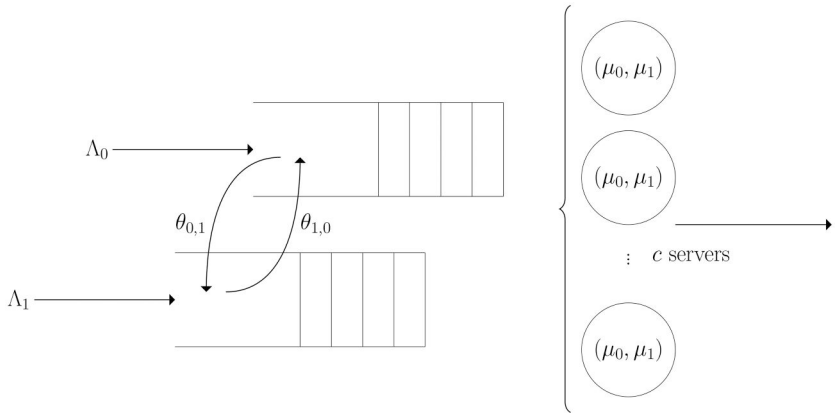


Figure 2. An example of a two-class priority queue.

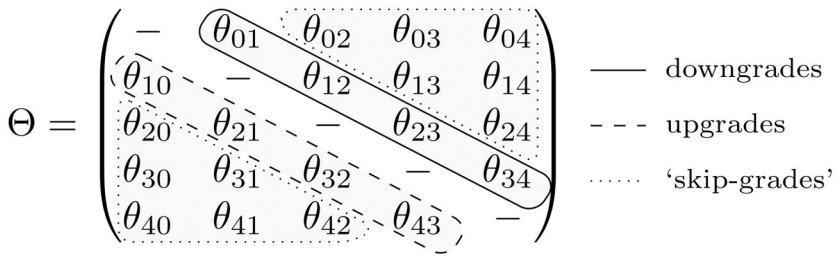


Figure 3. Representations of parts of the matrix Θ. Example when $K = 5$.

class not immediately above or below the current class), not commonly considered in the literature. 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 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 two implementations of this model:

- a discrete-event simulation,
- an exact model using Markov chains.

3. Simulation model logic

Discrete-event simulation is a common way of modelling queueing systems, especially those with non-standard customer behaviours as the development time for capturing new or complex behaviours is a lot quicker than Markov modelling (Standfield et al., 2014). One standard way of implementing discrete-event simulation is through the event scheduling approach (Robinson, 2014), a variant of the three-phase approach. In this work we use the Ciw library (Palmer et al., 2019) to simulate customers changing priority class. Ciw is an open-source Python library for discrete-event simulation of open queueing networks, which itself is built using the event scheduling approach. A key contribution of this work is the adaptation of the library's logic to facilitate the type of stochastic priority switching 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, and works by considering different classes of customer and mapping each class to a priority ranking, and re-sampling these rankings over time. Appendix A gives an overview of the event scheduling approach and details the adaptations required for priority switching logic.

Figure 4 shows example 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. 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 used. For the systems described in this paper, we choose Exponential distributions with rates determined by the class change matrix Θ. Ciw allows for pre-empted customers to resume, re-start, or re-sample their service time.

```
>>> import ciw
>>> N = ciw.create_network(
...     arrival_distributions={
...         'C1': [ciw.dists.Exponential(rate=1)],
...         'C2': [ciw.dists.Exponential(rate=3)]
...     },
...     service_distributions={
...         'C1': [ciw.dists.Exponential(rate=3)],
...         'C2': [ciw.dists.Exponential(rate=2)],
...     },
...     number_of_servers=[1],
...     priority_classes=({'C1': 0, 'C2': 1}, ["resample"]),
...     class_change_time_distributions={
...         'C1': {'C2': ciw.dists.Exponential(rate=1.5)},
...         'C2': {'C1': 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 stochastic priority switching.

3.1. Modelling unknown service disciplines

The stochastic priority switching model presented in Section 2 may be used to model situations where FIFO is not appropriate, but specific service disciplines may not be known. The premise here is that sequences of stochastic upgrades and downgrades of customers within the queue can model the same effect as the unknown service discipline. Recall now the motivating example discussed in Section 1.1, where it was established that FIFO was not appropriate and the service discipline was unknown. Here we attempt to use stochastic priority switching to model the surgical endoscopy waiting list.

According to the observed data referrals roughly follow a Poisson distribution with rate $\lambda = 0.463$, that is an average of 0.463 referrals per day, or one referral every 2.16 days. Service rate data is estimated to be around 0.5 a day, that is around one endoscopy procedure every other day. To determine the appropriateness of the stochastic priority switching 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 upgraded during their waiting time. (Note that an alternative model might be $\lambda_1 = 0$ and $\lambda_2 = 0.463$, that is all new referrals are not the most prioritised customers.)

We simulate this system under different parameters of θ_{12} and θ_{21}. For each parameter set we observe 1 year of referrals, over five trials. One KPI of interest is the amalgamated distribution of rank differences, or overtakes, over the trials; and we compare that with the distribution of the original system with indeterminate service discipline. The Wasserstein distance (Mostafaei & Kordnourie, 2011) between the modelled and actual distributions is calculated to measure the models' accuracies in approximating the indeterminate service discipline. Another KPI is the mean absolute percentage error, MAPE, between the simulated and actual average waiting times for patients that with bumped up or down the waiting list. Figure 5 compares the modelled and actual distributions of net rank difference, along with the Wasserstein metric W and MAPE in waiting times, 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 both in terms of Wasserstein distance and MAPE. This indicates that, with further parameter tuning, these stochastic priority switching models can be used to model unknown service disciplines.

We may then be tempted to use these found values of Θ to parametrise a model, to perform standard exercises such as what if scenarios. However we need more understanding on the dynamics of stochastic priority switching, and in particular the effect of Θ and other parameters on the system. As an example, consider the case above with $\lambda_1 = 0.463$, $\lambda_2 = 0$, $c = 1$, and now with $\theta_{12} = 3$, $\theta_{21} = 1$ as found above. Consider a small what-if scenario where upgraded customers are served quicker than downgraded customers, $\mu_1 = 0.4$ and $\mu_2 = 0.6$. Comparing the base scenario with this new scenario produced vastly different results, in particular, this new scenario results in an infinitely growing queue, as shown in Figure 6.

Simulation alone does might not give us sufficient insight into why this occurs. In the next section we built analytical models of stochastic priority switching, allowing a deeper insight into the system behaviour.

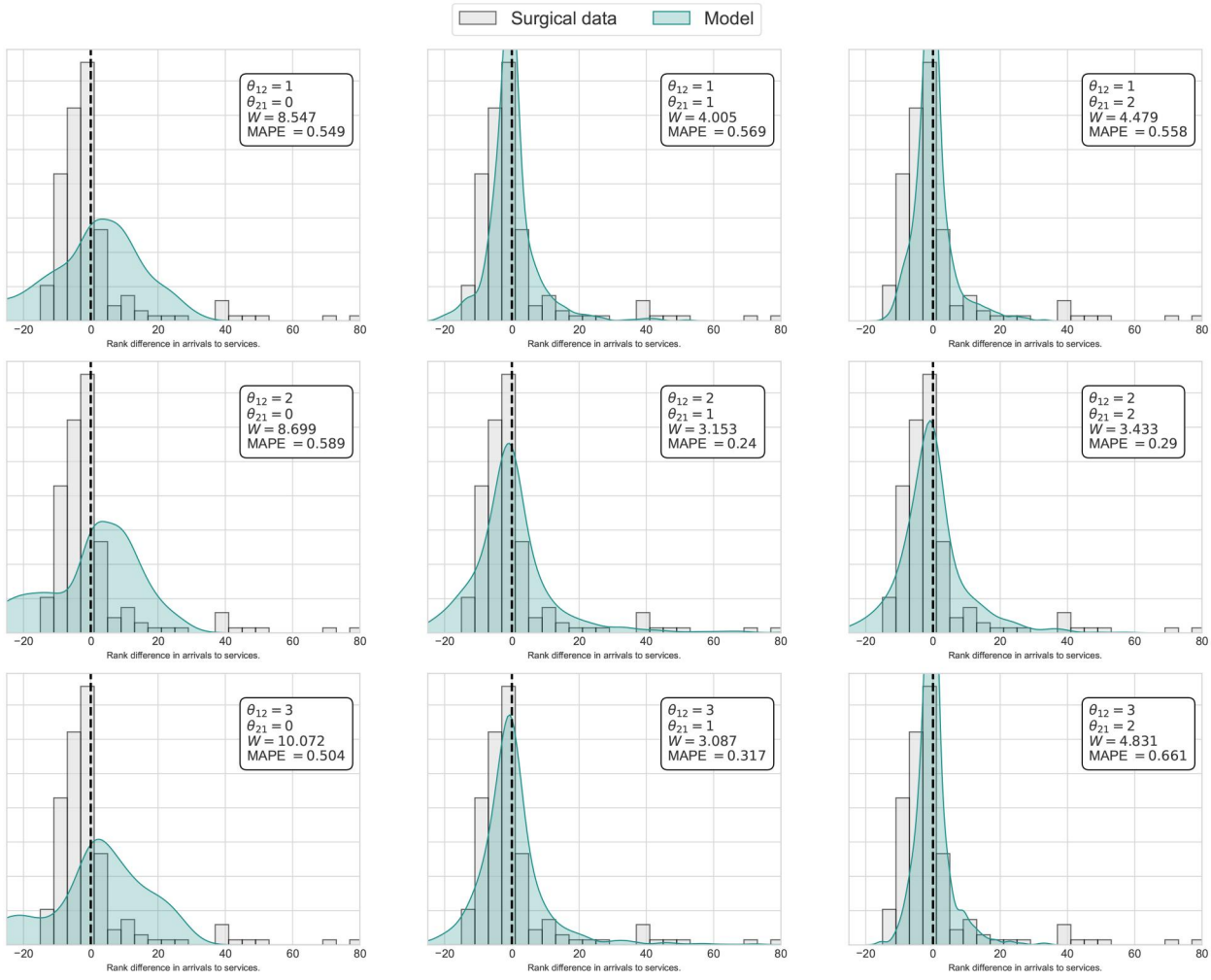


Figure 5. Comparison between simulated and observed overtakes for different values of θ_{12} and θ_{21}.

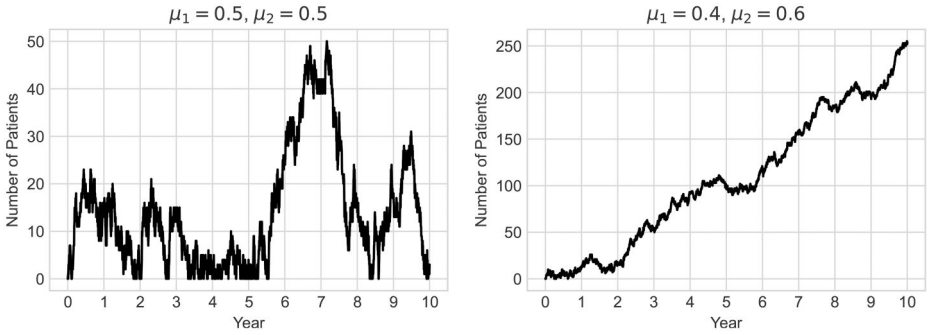


Figure 6. Comparison between two simulated scenarios, with stochastic priority switching, and differing service rates for each priority. In one scenario the queue size reaches a steady state, while in the other the queue size grows infinitely.

4. Markov chain models

The situation described in Section 2 can be modelled using a pair of 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 priority class present. 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. Given that service times are exponentially distributed and hence memoryless, the Markov chains are equivalent regardless of 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 (1a-1d), where $\underline{\boldsymbol{\delta}} = \underline{\mathbf{s}}_{t+1} - \underline{\mathbf{s}}_t$,

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

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. Here case (1a) denotes transitions representing customers arriving to the system, case (1b) denotes transitions representing customers finishing services, and case (1c) denotes transitions representing customers switching priorities.

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

Let $\pi_{\underline{\mathbf{s}}}$ denote the steady state probability of being in state $\underline{\mathbf{s}} \in S$ (omitting the time step index notation t), while L_k represents the expected number of customers of class k, and $\overline{L}$ represents the total expected number of customers present, given by Equations (3) and (4), respectively.

$$L_k = \sum_{\underline{\mathbf{s}}} \pi_{\underline{\mathbf{s}}} s_k \tag{3}$$

$$\overline{L} = \sum_{k=0}^{K-1} \sum_{\underline{\mathbf{s}}} \pi_{\underline{\mathbf{s}}} s_k \tag{4}$$

4.2. Sojourn time Markov chain formulation

To analyse individual customer statistics, the sojourn time Markov chain formulation is employed. Let $\underline{\mathbf{z}}_t = (z_{0,t}, z_{1,t}, ..., z_{n,t}..., z_{K-1,t}, m_t, n_t) \in \mathbb{N}^{K+2} \times (1, ..., K-1)$ 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 represents 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 Equations (5a)-(5j), where $\underline{\boldsymbol{\delta}} = \underline{\mathbf{z}}_{t+1} - \underline{\mathbf{z}}_t$,

$$q_{\underline{\mathbf{z}}_t, \underline{\mathbf{z}}_{t+1}} = \begin{cases} \mu_n & \text{if } z_{t+1} = \star \text{ and } \sum_{k \leq n} z_{k,t} < c\$, & (5a) \\ \lambda_n & \text{if } \delta_K = 1 \text{ and } \delta_i = 0 \ \forall \ i \neq K, & (5b) \\ \lambda_k & \text{if } \delta_k = 1 \text{ and } \delta_i = 0 \ \forall \ i \neq k \text{ and } k \neq n, & (5c) \\ A_{k,n,t}\mu_k & \text{if } \delta_k = -1 \text{ and } \delta_i = 0 \ \forall \ i \neq k \text{ and } k < K, & (5d) \\ \tilde{A}_{n,t}\mu_n & \text{if } \delta_K = -1 \text{ and } \delta_i = 0 \ \forall \ i \neq K, & (5e) \\ (z_{k,t} - A_{k,n,t})\theta_{k,\ell} & \text{if } \delta_k = -1 \text{ and } \delta_\ell = 1 \text{ and } \delta_i = 0 \ \forall \ i \neq k, \ell \text{ and } k < K \text{ and } \ell \neq n, K, K+1, & (5f) \\ (z_{K,t} - \tilde{A}_{n,t})\theta_{n,k} & \text{if } \delta_K = -1 \text{ and } \delta_k = 1 \text{ and } \delta_i = 0 \ \forall \ i \neq k, n \text{ and } k < K, & (5g) \\ (z_{k,t} - A_{k,n,t})\theta_{k,n} & \text{if } \delta_k = -1 \text{ and } \delta_K = 1 \text{ and } \delta_i = 0 \ \forall \ i \neq k, K, & (5h) \\ \theta_{n,k} & \text{if } \delta_n = z_{K,t} \text{ and } \delta_K = -z_{K,t} \text{ and } \delta_{K+1} = n - k \text{ and } \delta_i = 0 \quad \text{otherwise, and } \sum_{k \leq n} z_{k,t} < c, & (5i) \\ 0 & \text{otherwise} & (5j) \end{cases}$$

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 (6) and (7), respectively. Here cases correspond to arrivals, services, and priority switching, all in relation to the customer currently under consideration. This customer's class in n, which is subject to change. Cases (5a), and (5i) correspond to transitions that affect the customer under consideration, finishing service and switching priority, respectively. Cases (5b) and (5c) correspond to arrivals of other customers, of the same and different class to the considered individual, respectively. Cases (5e) and (5d) correspond to services of other customers, of the same and different class to the considered individual, respectively. Cases (5h), (5 g), and (5f) all correspond to customers switching

priorities, to n, from n, and from two classes neither of which are n, respectively.

$$A_{k,n,t} = \begin{cases} \min\left(c, \sum_{i\le k} z_{i,t}\right) - \min\left(c, \sum_{i<k} z_{i,t}\right) \\ \quad \text{if } k \le n \\ \min\left(c, \sum_{i\le k} z_{i,t} + 1 + z_{K,t}\right) \\ \quad -\min\left(c, \sum_{i<k} z_{i,t} + 1 + z_{K,t}\right) \\ \quad \text{if } n < k < K \end{cases} \tag{6}$$

$$\tilde{A}_{n,t} = \min\left(c, \sum_{i\le n} z_{i,t} + 1 + z_{K,t}\right) - \min\left(c, \sum_{i\le n} z_{i,t} + 1\right) \tag{7}$$

4.2.1. Sojourn time CDF

The sojourn time cumulative distribution function (CDF) is a phase-type distribution formed by this absorbing Markov chain, which can be found using standard methods (Stewart, 2009). Let X be the random variable representing the sojourn time of any customer, and let X_k represent the sojourn time of a customer who enters the system as a customer of class k. Then the CDFs we are interested in are $\mathbb{P}(X < x)$, and $\mathbb{P}(X_k < x)$ for each k.

Let T be the transition rate matrix constructed from the rates in Section 4.2 above without the rows and columns associated with the absorbing state $\star$. Customers arrive in all states where $z_K = 0$, and their class can be determined by n. Therefore 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 arrive, and define $\tilde{Z}_k = \{\underline{\mathbf{z}} \in \tilde{Z} \mid z_{K-1} = n = k\} \subset Z$ as the states where newly arriving customers of class k arrive.

Let $f : \tilde{Z} \to S$ be a map between states in $\tilde{Z}$ and S, given in Equation (8).

$$f(\underline{\mathbf{z}} = (z_0, z_1, ..., z_{K-1}, m, n)) = (z_0, z_1, ..., z_{K-1}) \tag{8}$$

Note that f 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 $\frac{\lambda_k}{\sum_{i=0}^{K-1}\lambda_i}$.

Now the overall sojourn time CDF is given by Equation (9), and the sojourn time CDF for customers of class k is given by Equation (10); where e^M represents the matrix exponential of a matrix M, and $[M]_{zs}$ represents an entry of the matrix M with row corresponding to state z and column corresponding to state s.

$$\mathbb{P}(X \le x) = 1 - \sum_{\underline{\mathbf{z}}\in\tilde{Z}} \sum_{s\in Z\setminus\{\star\}} \sum_{k=0}^{K-1} \frac{\lambda_k}{\sum_{i=0}^{K-1}\lambda_i} \pi_{f(\underline{\mathbf{z}})} [e^{Tx}]_{zs} \tag{9}$$

$$\mathbb{P}(X_k \le x) = 1 - \sum_{\underline{\mathbf{z}}\in\tilde{Z}_k} \sum_{s\in Z\setminus\{\star\}} \pi_{f(\underline{\mathbf{z}})} [e^{Tx}]_{zs} \tag{10}$$

4.2.2. Mean sojourn time calculation

Matrix exponentials can be computationally expensive to evaluate, and for a complete CDF Equation (9) would need to be evaluated many times for each value of x. Additionally, these CDF equations have no known easily computable inverse. However, matrix methods can give us summary statistics (Stewart, 2009).

Let $a_{\underline{\mathbf{z}}}$ denote the expected time to absorption from state $\underline{\mathbf{z}} \in Z$, and $Y = -T^{-1}$, then $a_{\underline{\mathbf{z}}}$ is given by elements of the vector Ye where e is a vector or ones. Similar to the CDFs then, we can get the overall mean sojourn time by Equation (11) and the mean sojourn time for customers arriving as class k by Equation (12).

$$\overline{\Psi} = \sum_{\underline{\mathbf{z}}\in\tilde{Z}} \sum_{k=0}^{K-1} \frac{\lambda_k}{\sum_{i=0}^{K-1}\lambda_i} \pi_{f(\underline{\mathbf{z}})} a_{\underline{\mathbf{z}}} \tag{11}$$

$$\Psi_k = \sum_{\underline{\mathbf{z}}\in\tilde{Z}_k} \pi_{f(\underline{\mathbf{z}})} a_{\underline{\mathbf{z}}} \tag{12}$$

4.2.3. Sojourn time variance calculation

Similarly to the mean sojourn time, the variance in sojourn times can be calculated by first calculating the variance in the times to absorption from each state. Let $\phi_{\underline{\mathbf{z}}}$ represent the variance in the times to absorption from state $\underline{\mathbf{z}} \in Z \setminus \{\star\}$, given by elements of the vector ϕ given in Equation (13) (Stewart, 2009).

$$\phi = 2Ye - \mathrm{sq}\{Y\}e \tag{13}$$

where $\mathrm{sq}\{Y\}$ is the matrix Y with each element squared.

In a similar calculation to Equation (11), the appropriate aggregation of states to give an overall variance in sojourn times, Φ, is given in Equation (14). This is derived by considering the overall time to absorption as a mixture distribution of times to absorption of each arriving state $\underline{\mathbf{z}} \in \tilde{Z}$, with weights corresponding to the probabilities of encountering those states upon arrival.

$$\Phi = \left(\sum_{\underline{\mathbf{z}}\in\tilde{Z}} \sum_{k=0}^{K-1} \frac{\lambda_k}{\sum_{i=0}^{K-1}\lambda_i} \pi_{f(\underline{\mathbf{z}})} \left(\phi_{\underline{\mathbf{z}}} + a_{\underline{\mathbf{z}}}{}^2 \right) \right) - \overline{\Psi}^2 \tag{14}$$

4.3. Bounded approximation

In order to analyse the above Markov chain models numerically, finite approximations are necessary. Let $b \in \mathbb{N}$ 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 occur when that number is exceeded. The equivalent b-bounded Markov chains associated with this system are

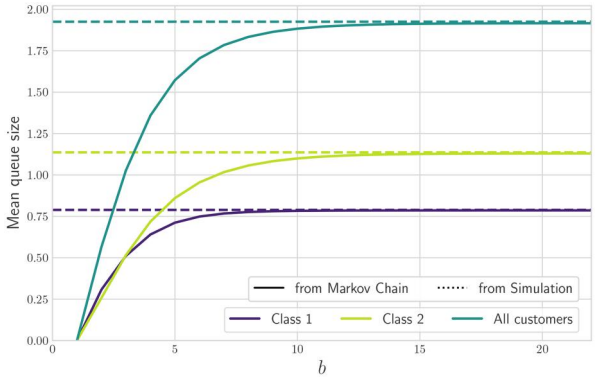


Figure 7. Demonstrating that as b increases, the expected number of customers of each class approaches that found using a long run simulation. The number of class 1 and class 2 customers are found using Equation (3), and the overall number of customers found using Equation (4).

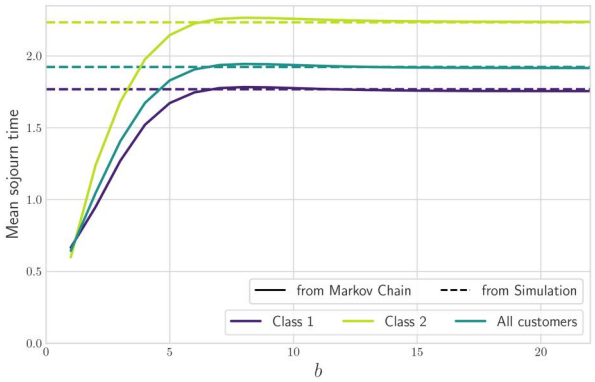


Figure 8. Demonstrating that as b increases, the expected sojourn time of customers of each class approaches that found using a long run simulation. The sojourn time for class 1 and class 2 customers are found using Equation (12), and sojourn time for the overall number of customers is found using Equation (11).

identical to those described in Sections 4.1 and 4.2 except with bounded state spaces $\underline{\mathbf{s}}_t =\in (0,1,...,b)^K$ and $\underline{\mathbf{z}}_t =\in (0,1,...,b)^{K+2} \times (1,...,K-1)$, respectively. These Markov chains are finite and are therefore 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, $\underline{\tilde{\pi}}$ 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 7 and 8, 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 $S_b = \left\{\underline{\mathbf{s}} \in S \mid b \in \underline{\mathbf{s}}\right\}$, 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, given in Equation (15), 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 normalisation by the equally likely state probabilities is necessary because the larger b is, the larger the state space is, meaning 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.

$$\mathcal{Q}(b) = \frac{|S|}{|S_b|} \sum_{s \in S_b} \tilde{\pi}_s \tag{15}$$

To demonstrate the effect of b on $\mathcal{Q}(b)$ under different systems, consider the stochastic priority switching 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 9 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.

The above measure cannot be used 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 (16) (Privault, 2013), where $q_{i,k}$ is the transition rate from state i to state j

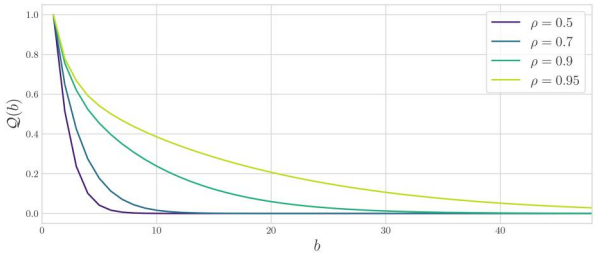


Figure 9. Demonstration of the effect of b on $\mathcal{Q}(b)$.

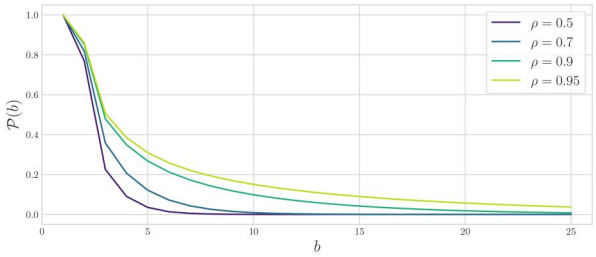


Figure 10. Demonstration of the effect of b on $\mathcal{P}(b)$.

defined in Section 4.2.

$$h_{i,J} = \begin{cases} \sum_k q_{i,k} h_{k,J} & \text{if } i \notin J \\ 1 & \text{if } i \in J \end{cases} \tag{16}$$

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 4.2.2, and given in Equation (17).

$$\mathcal{P}(b) = \sum_{\underline{\mathbf{z}} \in \tilde{Z}} \sum_{k=0}^{K-1} \frac{\lambda_k}{\sum_{i=0}^{K-1} \lambda_i} \pi_{c(\underline{\mathbf{z}})} h_{\underline{\mathbf{z}}, Z_b} \tag{17}$$

To demonstrate the effect of b on $\mathcal{P}(b)$ under different system, consider the same stochastic priority switching system with two customer classes used in the previous demonstration. Figure 10 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 accuracy of the bounded system; and similarly as ρ increases the boundary b needs to be larger before it can be considered irrelevant.

Appendix B shows examples where the simulation and Markov chains given the state probabilities, for $K = 2$, $K = 3$, and $K = 4$ priority classes.

4.4. Existence of stationary distributions

In all cases, the b-bounded system will only be an approximation of the infinite system if that infinite system is stationary, that is it reaches a steady-state and the queue does not grow indefinitely. Proposition 1 gives a naive check for the existence or non-existence of steady states for work conserving queues, but does not cover all possibilities. A work conserving queue is one where the total work that needs to be done by the servers is not lowered or increased by the priority or service discipline (Wolff, 1970).

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 stochastically changing 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. □

Proposition 1 applies to the stochastic priority switching system of this paper. If pre-empted customers resume their service upon re-entering service, then the system is work conserving. Otherwise, if the pre-empted customers restart or resample their service, despite not technically being work conserving any more, the systems are equivalent under Exponential service times, and so still applies here.

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 stochastic priority switching, 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 (Dickey & Fuller, 1979) 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 (The Traces library developers, 2023) is used to take regularly spaced moving averages before the hypothesis test is undertaken.

Table 1. Parameters used in demonstrations of the ADF test.

	λ_1	λ_2	c	μ_1	μ_2	θ_{12}	θ_{21}
Example 1	2	1	1	4	4	1	1
Example 2	2	1	1	1	1	1	1

To demonstrate this, consider two examples, with parameters defined in Table 1. Example 1 is guaranteed to reach steady state by Proposition 1, while Example 2 is guaranteed not to reach steady state.

Figures 11a and 11b 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.

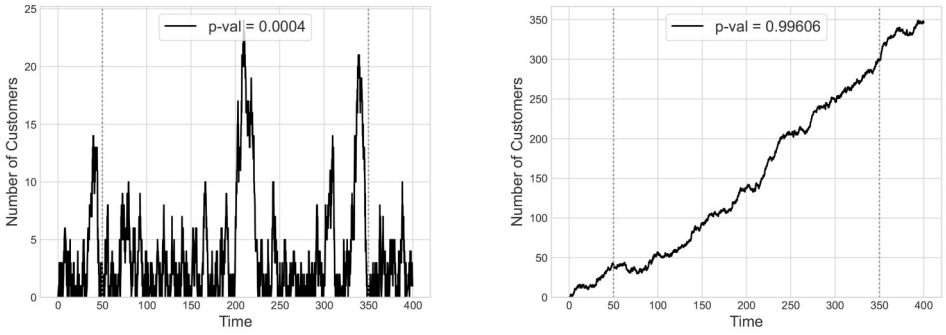
There is a gap in Proposition 1 for systems where $c\min_i \mu_i \leq \sum_i \lambda_i < c\max_i \mu_i$. Indeed it is in this gap that our previous scenario in Figure 6 falls, and it is here where stochastic priority switching can influence the stationarity of the system. Consider a two class system 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 $\rho_{\min} < 1 < \rho_{\max}$, and the prioritised class receive a slower service rate;
- $\mu_1 = 5$ and $\mu_2 = 3$: here $\rho_{\min} < 1 < \rho_{\max}$, 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 $\theta_{21} = 1$: both 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 12 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



(a) State time series for Example 1. (b) State time series for Example 2.

Figure 11. Demonstration of the ADF test on states that do and do not reach steady state according to Proposition 1.

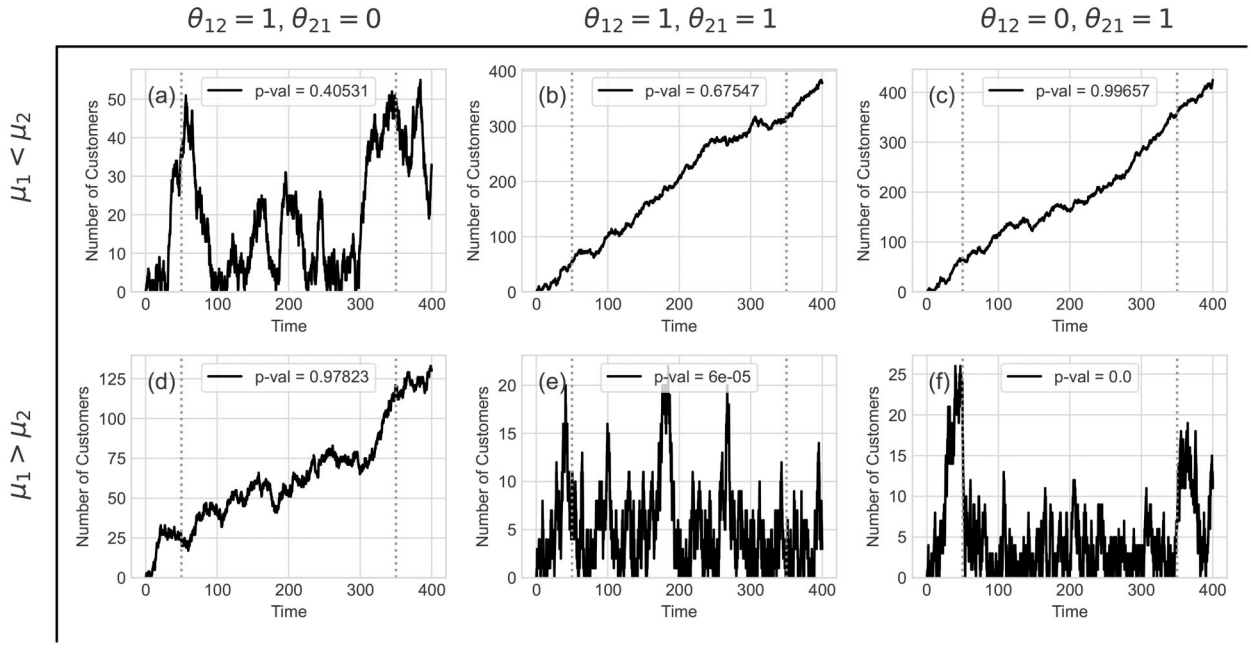


Figure 12. Investigating the stationarity under six cases not covered by Proposition 1.

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.

4.5. Effect of Θ on customer experience

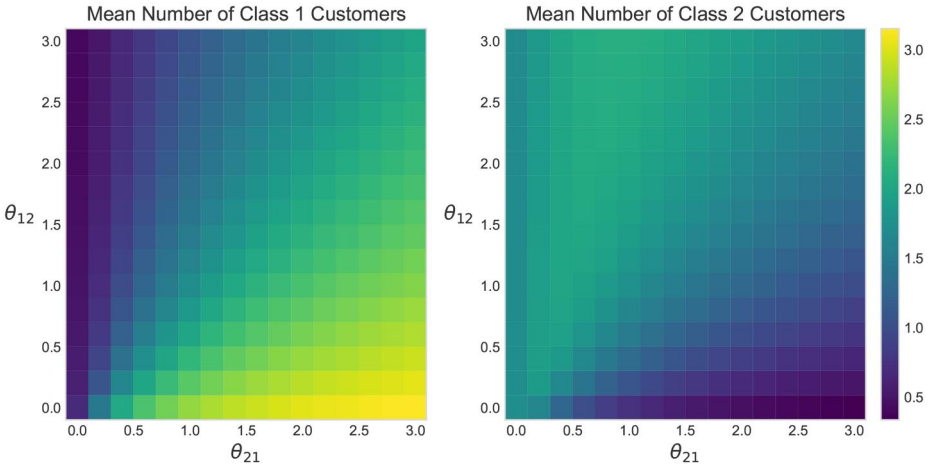
In Section 3.1 that it is established that some service disciplines can be modelled as stochastically changing priorities with a class change matrix $\boldsymbol{\Theta}$, and we now have Markov chain models that can be used to consider the effect of $\boldsymbol{\Theta}$ on the system behaviour. We now investigate the effect of this matrix on customer experience, as this 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 three scenarios that we will use to investigate the effect of $\boldsymbol{\Theta}$, defined in Table 2. Each scenario involves two classes of customer: in Scenario A prioritised customers have a slower service rate than unprioritised customer, in Scenario B both customer classes have the same service rate, while in Scenario C prioritised customers have a faster service rate than prioritised customers. The Markov chain models of Section 4 are built with these parameters, and all values of θ_{12} and θ_{21} ranging 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 statistic are found and compared.

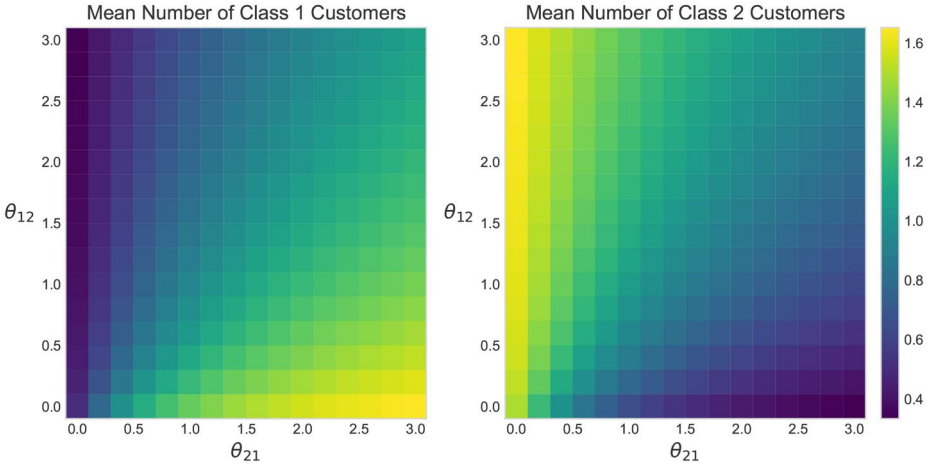
Table 2. Parameters used in experiments that investigate the effect of Θ.

Scenario	λ_1	λ_2	c	μ_1	μ_2
A	1	1	1	5/2	7/2
B	1	1	1	3	3
C	1	1	1	7/2	5/2

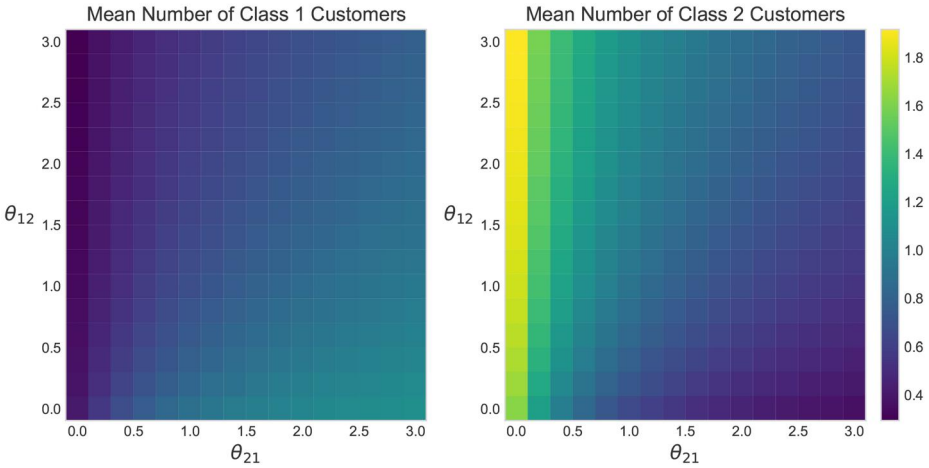
Figures 13a, 13b, and 13c give L_1 and L_2, the steady-state average number of customers of each class, for all pairs of θ_{12} and θ_{21}, under Scenarios A, B, and C, respectively. In general it can be seen that as θ_{12} increases in comparison to θ_{21} then we expect less customers of class 1 and more of class 2, and



(a) Scenario A ($\mu_1 < \mu_2$)



(b) Scenario B ($\mu_1 = \mu_2$)



(c) Scenario C ($\mu_1 > \mu_2$)

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

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, B, and C, as the magnitude $\tilde{\theta}$ changes. In Scenario A, where prioritised customers have a slower service rate, increasing the priority change

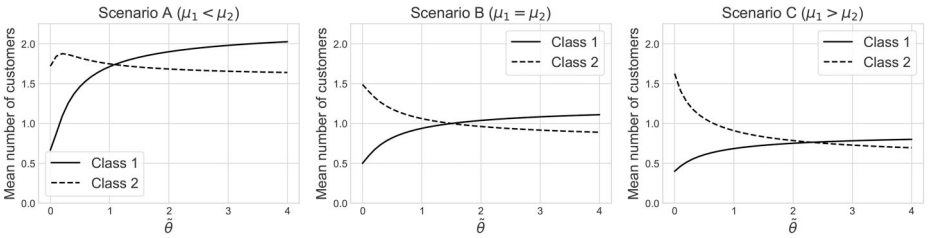


Figure 14. Average number of customers of each class, as $\tilde{\theta}$ changes.

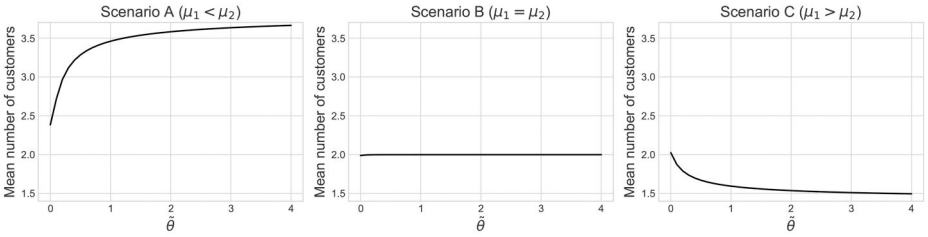


Figure 15. Average number of customers overall as $\tilde{\theta}$ changes.

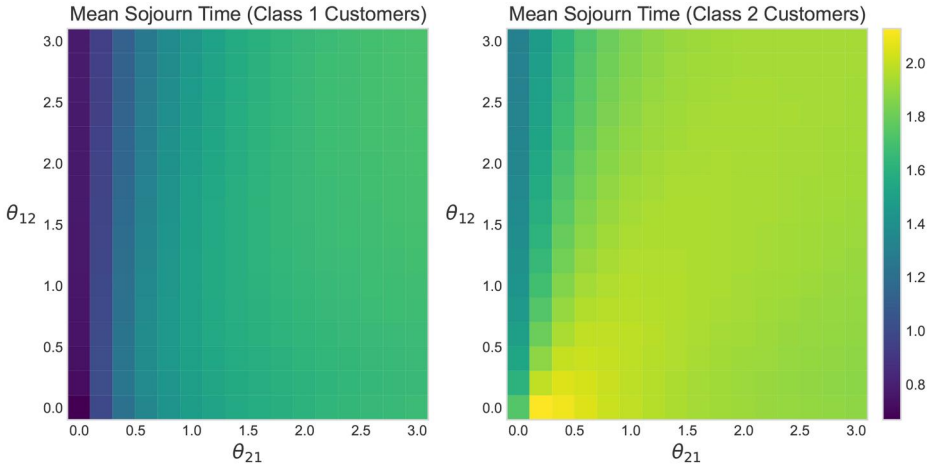
rate increases the prioritised customers present, and decreases the number of unprioritised customers. In Scenario B the same effect can be seen, but the size of this effect is smaller. In Scenario C the opposite trend is true. Looking at the effect of $\tilde{\theta}$ on the overall number of customers 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 C, while in Scenario B where all customers have equal service rates, increasing the rate of priority change does not have an effect of the overall number of customers present. 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 C prioritised customers are processed quicker.

Figures 16a, 16b, and 16c give Ψ_1 and Ψ_2, the steady-state average sojourn time of customers beginning in each class, for all pairs of θ_{12} and θ_{21}, under Scenarios A, B, and C, 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 C they are more likely to be served as a class 1 customer with higher service rate, and so shorter sojourn time. In Scenario B it seems that θ_{21} does not affect the sojourn time of class 2 customers.

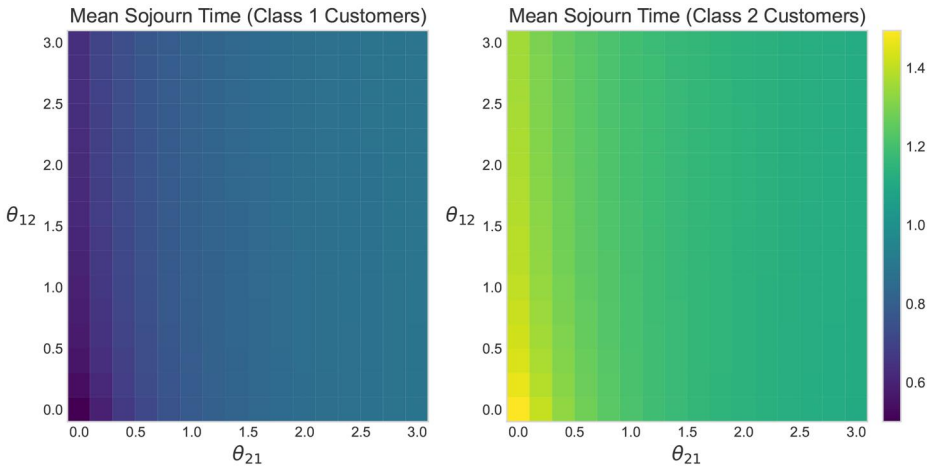
The variance in the sojourn times can also be found. Figure 17 shows the affect of the priority switching rates on the overall variance of the sojourn time Φ. In all scenarios the sojourn time variance decreases as θ_{21} increases. This is expected, as a very high value of θ_{21} would correspond to all customers being of high priority, or equivalently there being no priority, and introducing priority into a system increases its variance. However, in Scenarios A and B, for very small values of θ_{21} the opposite effect is seen. Increasing θ_{12} generally increases the sojourn time variance. This general increase in variance is due to moving people out of the prioritised class increases their variance, as the unprioritised class's sojourn times rely on the behaviour of different customer classes, whereas the prioritised customers' sojourn times only rely on the prioritised queue. This effect is not as pronounced in Scenario C, where the prioritised customers have a higher service rate, as the prioritised queue is reducing faster, and so has less effect on the sojourn time of the unprioritised customers.

4.6. Case study insights

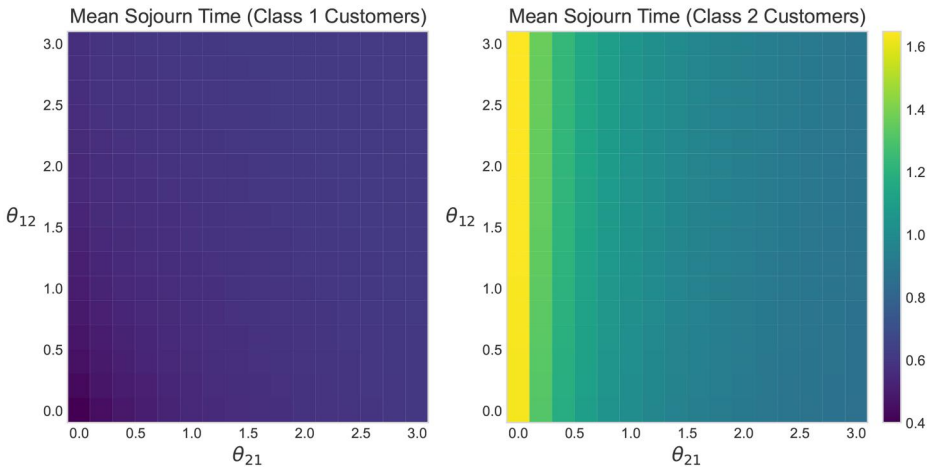
The above Markov chain models can now be used to investigate the effect of the stochastic priority switching on our surgical endoscopy case study. In particular, we will consider the variance in the sojourn time of the customers, as reducing variability amongst patients is one of the core principles of prudent healthcare (Bevan Commission, 2015). First we will compare the sojourn time variance Φ, using Equation (14), between the scenario that does and doesn't use stochastic priority switching, which are given in Table 3. The variability in the customers' sojourn times is much greater when modelling using stochastic priority switching, over 16 times the variance without priority switching, demonstrating that some important performance measures are not captured by FIFO alone.



(a) Scenario A ($\mu_1 < \mu_2$)



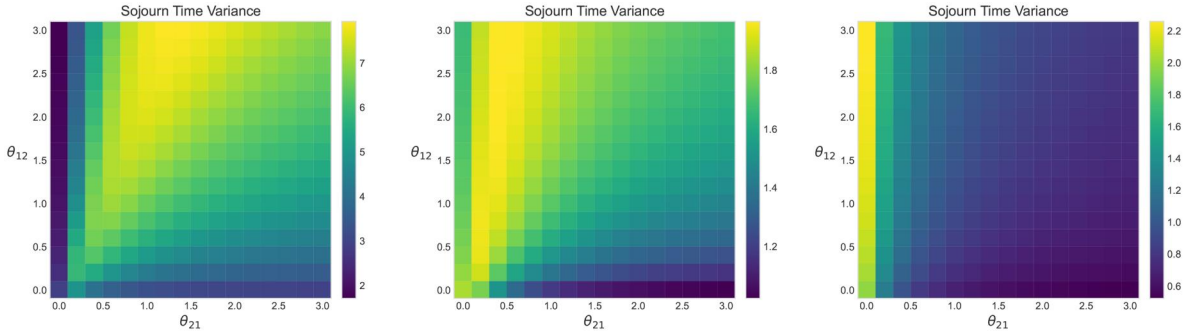
(b) Scenario B ($\mu_1 = \mu_2$)



(c) Scenario C ($\mu_1 > \mu_2$)

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

We previously saw in Figure 6 that small changes to the service rates can have large effects on the system. Figure 18a shows the effect of increasing in the service rate μ on the customers' sojourn time variance Φ, and we see that even just a very slight increase to the service rate could drastically reduce the variation in the customers' waiting times, this may be due to the initially very high traffic intensity ($\rho = \lambda/\mu = 0.463/0.5 = 0.926$) causing long waiting times and therefore lots of customers switching

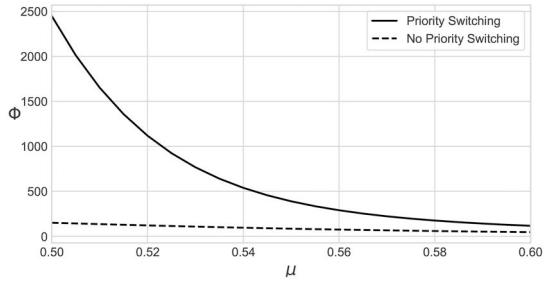


(a) Scenario A ($\mu_1 < \mu_2$) (b) Scenario B ($\mu_1 = \mu_2$) (c) Scenario C ($\mu_1 > \mu_2$)

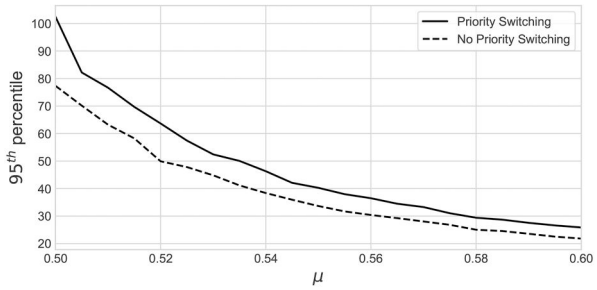
Figure 17. Variance in sojourn time for customers beginning in each class, as Θ changes.

Table 3. Comparison in the variance in customer sojourn time for models with and without stochastic priority switching, for the surgical endoscopy case study of Section 1.1.

Scenario	θ_{12}	θ_{21}	Φ
No priority switching	0	0	150.8
Priority switching	2	1	2449.4



(a) Effect of Φ.



(b) Effect of the 95th percentile of the sojourn time.

Figure 18. The effect of small increases in the service rate, with and without priority switching.

priorities. We can also consider percentiles of the sojourn time, which may be a more interpretable measure of variability than Φ. As the sojourn time distribution of Equations (9) and (10) do not have an easily found inverse, this is difficult to analyse analytically, but can be considered using the simulation described in Section 3. Figure 18b gives the 95th percentile of the sojourn time, with and without priority switching, as the service rate increases. We see that under priority switching the sojourn times always have a longer tail, but the gap shortens as the traffic intensity decreases.

5. Conclusions

In this paper a generalised model of stochastic priority switching within an $M/M/c$ queue is presented. The general formulation was given in Section 2, followed by two methodologies for modelling it, 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 introduce 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 was inappropriate. We show that modelling as stochastically changing priorities can approximate the queue behaviour, 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 to queue controllers, such as waiting list managers, who can influence or tweak the class change matrix, for example if prioritised customers have a faster service rate, then the queue is managed better when the priority change rate is higher, though at the expense of the prioritised customers themselves. In the surgical endoscopy case study, we show that the variance in the customers' sojourn times are highly sensitive to the service rate.

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 Palmer et al. (2024) and all development of the code took place at https://github.com/geraintpalmer/DynamicClasses.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

No funding was received.

References

Adiri, I. (1971). A dynamic time-sharing priority queue. *Journal of the ACM*, *18*(4), 603–610. https://doi.org/10.1145/321662.321675

Bagchi, U., & Sullivan, R. S. (1985). Dynamic, non-preemptive priority queues with general, linearly increasing priority function. *Operations Research*, *33*(6), 1278–1298. https://doi.org/10.1287/opre.33.6.1278

Bevan Commission. (2015). A prudent approach to health: Prudent health principles. https://bevancommission.org/wp-content/uploads/2023/09/A-Prudent-Approach-to-Health-Prudent-Principles.pdf

Bilodeau, B., & Stanford, D. A. (2022). High-priority expected waiting times in the delayed accumulating priority queue with applications to health care kpis. *INFOR: Information Systems and Operational Research*, *60*(3), 285–314. https://doi.org/10.1080/03155986.2022.2038962

Bradford Delong, W., Polissar, N., & Neradilek, B. (2008). Timing of surgery in cauda equina syndrome with urinary retention: Meta-analysis of observational studies. *Journal of Neurosurgery: Spine*, *8*(4), 305–320.

D'Alessandro, C., Golmard, J. L., Lebreton, G., Laali, M., Varnous, S., Farahmand, P., Vidal, C., & Leprince, P. (2017). High-urgency waiting list for cardiac recipients in france: Single-centre 8-year experience. *European Journal of Cardio-Thoracic Surgery*, *51*(2), 271–278.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, *74*(366), 427–431. https://doi.org/10.2307/2286348

Ding, Y., Park, E., Nagarajan, M., & Grafstein, E. (2019). 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. https://doi.org/10.1287/msom.2018.0719

Down, D. G., & Lewis, M. E. (2010). The n-network model with upgrades. *Probability in the Engineering and Informational Sciences*, *24*(2), 171–200. https://doi.org/10.1017/S0269964809990222

Dudin, A., Dudina, O., Dudin, S., & Samouylov, K. (2021). Analysis of single-server multi-class queue with unreliable service, batch correlated arrivals, customers impatience, and dynamical change of priorities. *Mathematics*, *9*(11), 1257. https://doi.org/10.3390/math9111257

Ferrand, Y. B., Magazine, M. J., Rao, U. S., & Glass, T. F. (2018). Managing responsiveness in the emergency department: Comparing dynamic priority queue with fast track. *Journal of Operations Management*, *58*, 15–26.

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

Garbuz, D. S., Xu, M., Duncan, C. P., Masri, B. A., & Sobolev, B. (2006). Delays worsen quality of life outcome of primary total hip arthroplasty. *Clinical Orthopaedics and Related Research*, *447*, 79–84. https://doi.org/10.1097/01.blo.0000203477.19421.ed

Grindlay, A. A. (1965). Tandem queues with dynamic priorities. *Journal of the Operational Research Society*, *16*(4), 439–451. https://doi.org/10.2307/3006711

He, Q.-M., Xie, J., & Zhao, X. (2012). Priority queue with customer upgrades. *Naval Research Logistics (NRL)*, *59*(5), 362–375. https://doi.org/10.1002/nav.21494

Holtzman, J. M. (1971). Bounds for a dynamic-priority queue. *Operations Research*, *19*(2), 461–468. https://doi.org/10.1287/opre.19.2.461

Jackson, J. R. (1960). Some problems in queueing with dynamic priorities. *Naval Research Logistics Quarterly*, *7*(3), 235–249. https://doi.org/10.1002/nav.3800070304

Kella, O., & Ravner, L. (2017). Lowest priority waiting time distribution in an accumulating priority lévy queue. *Operations Research Letters*, *45*(1), 40–45. https://doi.org/10.1016/j.orl.2016.11.007

Kleinrock, L. (1964). A delay dependent queue discipline. *Naval Research Logistics Quarterly*, *11*(3-4), 329–341. https://doi.org/10.1002/nav.3800110306

Klimenok, V., Dudin, A., Dudina, O., & Kochetkova, I. (2020). Queuing system with two types of customers and dynamic change of a priority. *Mathematics*, *8*(5), 824. https://doi.org/10.3390/math8050824

Knessl, C., Tier, C., & Il Choi, D. (2003). A dynamic priority queue model for simultaneous service of two traffic types. *SIAM Journal on Applied Mathematics*, *63*(2), 398–422. https://doi.org/10.1137/S0036139901390842

Lee, S., Dudin, S., Dudina, O., Kim, C., & Klimenok, V. (2020). A priority queue with many customer types, correlated arrivals and changing priorities. *Mathematics*, *8*(8), 1292. https://doi.org/10.3390/math8081292

Li, N., & Stanford, D. A. (2016). Multi-server accumulating priority queues with heterogeneous servers. *European Journal of Operational Research*, *252*(3), 866–878. https://doi.org/10.1016/j.ejor.2016.02.010

Li, N., Stanford, D. A., Taylor, P., & Ziedins, I. (2017). Nonlinear accumulating priority queues with equivalent linear proxies. *Operations Research*, *65*(6), 1712–1721. https://doi.org/10.1287/opre.2017.1613

Mostafaei, H., & Kordnourie, S. (2011). Probability metrics and their applications. *Applied Mathematical Sciences*, *5*(4), 181–192.

Netterman, A., & Adiri, I. (1979). A dynamic priority queue with general concave priority functions. *Operations Research*, *27*(6), 1088–1100. https://doi.org/10.1287/opre.27.6.1088

Palmer, G. I. (2018). *Modelling deadlock in queueing systems* [PhD thesis]. Cardiff University.

Palmer, G. I., Knight, V., Panayidis, M., & Williams, E. (2024). Source code for "queues under stochastic priority switching". https://doi.org/10.5281/zenodo.13710925
Palmer, G. I., Vincent, A., Knight, Paul, R., Harper, Asyl., & L., Hawa. (2019). Ciw: An open-source discrete event simulation library. *Journal of Simulation*, *13*(1), 68–82. https://doi.org/10.1080/17477778.2018.1473909
Panayiotopoulos, J.-C. (1980). Solving queueing systems with increasing priority numbers. *Journal of the Operational Research Society*, *31*(7), 637–646. https://doi.org/10.1057/jors.1980.121
Powers, J., McGree, J. M., Grieve, D., Aseervatham, R., Ryan, S., & Corry, P. (2023). Managing surgical waiting lists through dynamic priority scoring. *Health Care Management Science*, *26*(3), 533–557. https://doi.org/10.1007/s10729-023-09648-1
Privault, N. (2013). Understanding Markov chains. In Examples a*nd applications* (vol. 357, p. 358). Springer-Verlag Singapore.
Robinson, S. (2014). *Simulation: The practice of model development and use*. Palgrave Macmillan.
Sharif, A. B., Stanford, D. A., Taylor, P., & Ziedins, I. (2014). A multi-class multi-server accumulating priority queue with application to health care. *Operations Research for Health Care*, *3*(2), 73–79. https://doi.org/10.1016/j.orhc.2014.01.002
Sharma, K. C., & Sharma, G. C. (1994). A delay dependent queue without pre-emption with general linearly increasing priority function. *Journal of the Operational Research Society*, *45*(8), 948–953. https://doi.org/10.1057/jors.1994.147
Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, *3*(1-2), 59–66. https://doi.org/10.1002/nav.3800030106
Standfield, L., Comans, T., & Scuffham, P. (2014). Markov modeling and discrete event simulation in health care: A systematic comparison. *International Journal of Technology Assessment in Health Care*, *30*(2), 165–172. https://doi.org/10.1017/S0266462314000117
Stanford, D. A., Taylor, P., & Ziedins, I. (2014). Waiting time distributions in the accumulating priority queue. *Queueing Systems*, *77*(3), 297–330. https://doi.org/10.1007/s11134-013-9382-6
Stewart, W. J. (2009). *Probability, Markov chains, queues, and simulation: The mathematical basis of performance modeling*. Princeton University Press.
The Traces Library Developers. (2023). Traces. https://traces.readthedocs.io/en/master/api_reference.html,
Van Mieghem, J. A. (1995). Dynamic scheduling with convex delay costs: The generalized $c\mu$ rule. *The Annals of Applied Probability*, *5*, 809–833.
Williams, P., Beard, D. J., Verghese, N. (2020). Yet another iceberg? the hidden potential harm of elective orthopaedic waiting lists. Retrieved August 15, 2024, from https://www.boa.ac.uk/resource/yet-another-iceberg-the-hidden-potential-harm-of-elective-orthopaedic-waiting-lists.html
Wolff, R. W. (1970). Work-conserving priorities. *Journal of Applied Probability*, *7*(2), 327–337. https://doi.org/10.2307/3211968
Xhafa, A., Tonguz, O. K. (2001). Dynamic priority queueing of handoff requests in PCS. In *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No. 01CH37240)* (vol. 2, pp. 341–345). IEEE.
Xie, J., He, Q.-M., & Zhao, X. (2008). Stability of a priority queueing system with customer transfers. *Operations Research Letters*, *36*(6), 705–709. https://doi.org/10.1016/j.orl.2008.06.007

Appendix A. Simulation details

Here we detail the logic of the simulation model discussed in Section 3, implemented in the Ciw library. It utilised an event scheduling approach, with three phases: an A-phase which advances the clock to the next scheduled event, a B-phase where scheduled events are carried out, and a C-phase where conditional events are carried out. Appendix Figure 1 shows a flow diagram of the logic of the event scheduling approach.

The primary scheduled, or B-events that occur in queueing simulations 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.

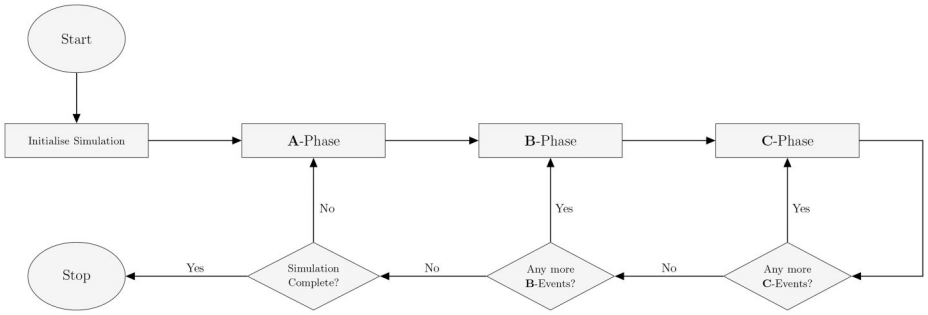
Any other customer, server, or system behaviour to be captured by the simulation corresponds to increasing the range of B- and C-events that can happen during the simulation run. For example if servers are subject to a work schedule, then extra B-events include a server going off and on duty, and extra C-events would include a customer beginning service after another customer has left the server.

In the case of customers randomly changing priority classes while waiting, one additional B-event and two additional C-event need to be included:

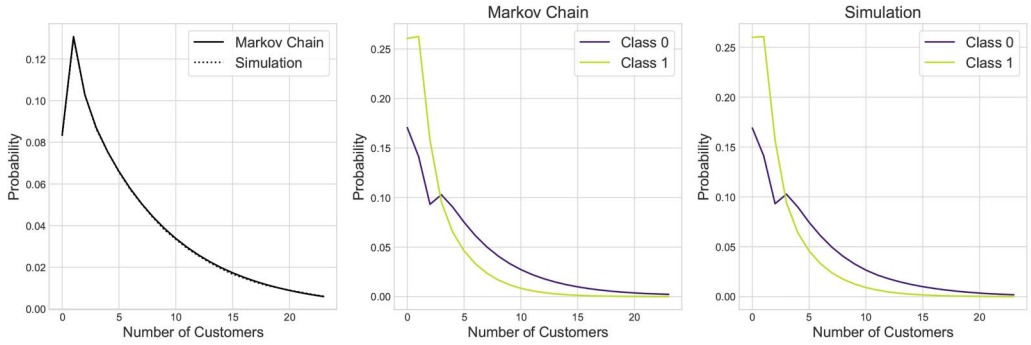
- Upon arrival to the queue customers are assigned a date in which they will change customer class, determined by randomly sampling from a distribution. As such each customer's event of changing customer class is scheduled for the future, and are therefore **B**-events. If those customers begin service (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.
- If a newly arriving customer is of a higher priority than a customer in service, or if a lower priority customer is upgraded to a higher priority a customer in service, then the lower priority customer in service is pre-empted. They stop service and are placed at the head of the queue. This happens immediately after an arrival or after an upgrade, and so is a **C**-event.

For the Ciw code shown in Figure 4, 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 use any probability distribution, and is not restricted to Exponential distributions.

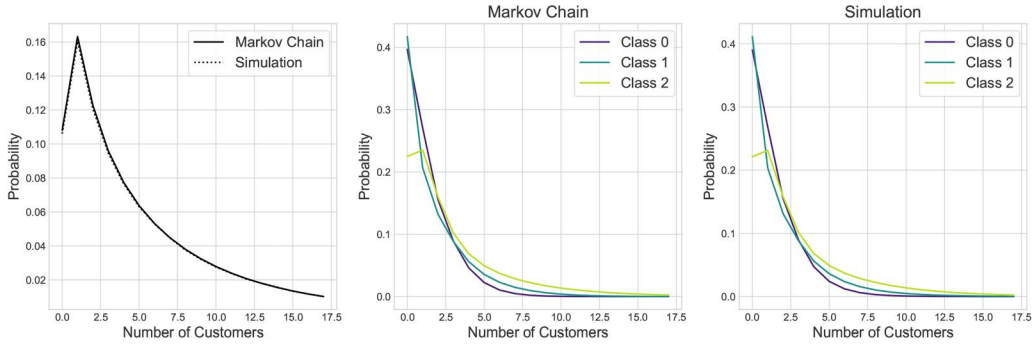
Appendix B. Further experiments



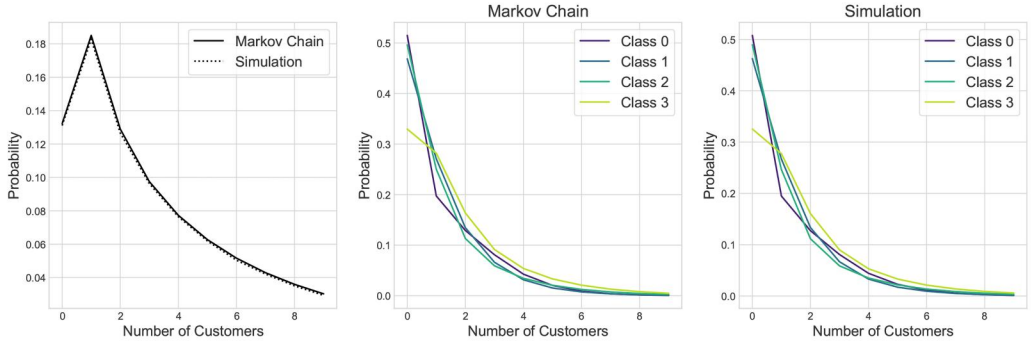
Appendix Figure 1. Flow diagram of the event scheduling approach, adapted from Palmer (2018).



(a) $K = 2$ $\lambda_1 = 3$ $\lambda_2 = 4$ $\mu_1 = 4$ $\mu_2 = 5$ $c = 2$ $b = 24$ $\Theta = \begin{pmatrix} 0 & 7 \\ 12 & 0 \end{pmatrix}$ Simulation time $= 200000$, warm-up $= 5000$



(b) $K = 3$ $\lambda_1 = 2$ $\lambda_2 = 1$ $\lambda_3 = 4$ $\mu_1 = 4$ $\mu_2 = 4.5$ $\mu_3 = 5$ $c = 2$ $b = 18$ $\Theta = \begin{pmatrix} 0 & 2 & 4 \\ 1 & 0 & 5 \\ 1 & 3 & 0 \end{pmatrix}$ Simulation time $= 200000$, warm-up $= 5000$



(c) $K = 4$ $\lambda_1 = 0.5$ $\lambda_2 = 1.5$ $\lambda_3 = 1$ $\lambda_4 = 3$ $\mu_1 = 3.5$ $\mu_2 = 4$ $\mu_3 = 3.5$ $\mu_4 = 5$ $c = 2$ $b = 10$ $\Theta = \begin{pmatrix} 0 & 2 & 4 & 1 \\ 1 & 0 & 5 & 1 \\ 1 & 3 & 0 & 3 \\ 2 & 1 & 1 & 0 \end{pmatrix}$ Simulation time $= 200000$, warm-up $= 5000$

Appendix Figure 2. Simulation and Markov chain methods for finding state probabilities for 2, 3, and 4 priority classes.