Operations Research for Health Care 3 (2014) 73–79

Contents lists available at ScienceDirect

Operations Research for Health Care

journal homepage: www.elsevier.com/locate/orhc

A multi-class multi-server accumulating priority queue with application to health care

Azaz Bin Sharif[a,*], David A. Stanford[a], Peter Taylor[b], Ilze Ziedins[c]

[a] Department of Statistical and Actuarial Sciences, University of Western Ontario, London, Ontario, Canada N6A 5B7

[b] Department of Mathematics and Statistics, University of Melbourne, Vic 3010, Australia

[c] Department of Statistics, University of Auckland, 1010, New Zealand

ARTICLE INFO

Article history:
Received 8 May 2013
Accepted 9 January 2014
Available online 21 January 2014

Keywords:
Multi-server priority
Time-dependent priority
Non-preemptive priority
Health care priority

ABSTRACT

We consider the accumulating priority queue (APQ), a priority queue where customer priorities are a function of their waiting time. This time-dependent priority model was first proposed by Kleinrock (1964), and, more recently, Stanford et al. (2013) derived the waiting time distributions for the various priority classes when the queue has a single server. The present work derives expressions for the waiting time distributions for a multi-server APQ with Poisson arrivals for each class, and a common exponential service time distribution. It also comments on how to choose feasible accumulation rates to satisfy specified performance objectives for each class.

1. Introduction

One way of dealing with waiting line problems in the presence of diverse client needs is a priority queueing mechanism. A practical example from the field of health care would be the acuity rating systems which have been employed in many countries to classify emergency patients according to their level of severity. In the context of emergency medicine, the Canadian Triage and Acuity Scale (CTAS) [1] and the Australian Triage Scale (ATS) [2] (on which CTAS is based) are two examples, where patients are classified into five priority classes (see Table 1). Each class is associated with a specified performance target assessed in terms of a set of Key Performance Indicators (KPIs). Each KPI comprises a threshold time standard, along with the proportion of patients who should not exceed that time standard. These standards are ostensibly based upon clinical need, although the case can be made that for the lower acuity classes, the KPIs reflect performance benchmarks more than clinical need.

A different situation where a prioritized system arises in health care is in hip and knee replacement surgery [3], where distinctions are made among various elective classes (see Table 2).

There is no reason to presume, a priori, that the stipulated KPIs for each customer class will be met under a classical priority service discipline, for any given set of patient presentation rates. It might well be the case in a two-class system, for instance, that high-priority patients may receive better service than their specified target, while the service level of the low-priority patients misses its target. This indicates the need for a priority mechanism that can provide the extra degree of flexibility required to align the observed performance levels with the specified KPIs. The first model to do this was due to Kleinrock [4], who let customers from a given class (say, k; $k \in \{1, 2, \ldots, K\}$, where K denotes the number of classes) accumulate 'priority' at a rate $b_k > 0$, where $b_k > b_j$ for $k < j$. In this way, a customer from a non-urgent class who experiences a very long wait will eventually accumulate sufficient priority to access the server even when some customers from a more urgent class may be present, and at an earlier time point than if a static priority mechanism were in place.

Kleinrock [4]'s analysis gave a set of recursive formulae for the mean waiting time before service for each class. However, as illustrated in the two previous examples, the performance of a queueing system in a health care setting is usually specified by the tail of the waiting time distribution for each class, and not by the average waiting times. With this in mind, Stanford et al. [5] recently reconsidered Kleinrock's model, which they renamed the "accumulating priority queue" (APQ), and obtained the waiting time distribution for each priority class in the single server setting.

It can be argued that a variant of the accumulated priority approach is being used already in certain priority health care settings, on an implicit level at least, whenever the deciding health care professional factors the time spent waiting as well as the patient's acuity level in the decision to select the next patient for

* Corresponding author. Tel.: +1 5197098204.
E-mail addresses: asharif9@uwo.ca (A.B. Sharif), stanford@stats.uwo.ca (D.A. Stanford), p.taylor@ms.unimelb.edu.au (P. Taylor), i.ziedins@auckland.ac.nz (I. Ziedins).

2211-6923/$ – see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.orhc.2014.01.002

Table 1
CTAS key performance indicators.

Category	Classification	Access	Performance level
1	Resuscitation	Immediate	98%
2	Emergency	15 min	95%
3	Urgent	30 min	90%
4	Less urgent	60 min	85%
5	Not urgent	120 min	80%

Table 2
Key performance indicators for hip and knee replacement surgery, Canada.

Category	Wait time target
Emergency	Immediate to 24 h
Urgent, priority 1	Within 30 days
Urgent, priority 2	Within 90 days
Scheduled	Consultation within 3 mon, Treatment in next 6 mon

treatment. In fact, Hay et al. [6] in their simulation model present an approach employing what they call "operational priority", in which each patient is assessed, and assigned an initial priority score which then increases over time.

This note is the first to present distributional results of a queueing-theoretic form for an APQ in a multi-server setting. The results that we present have restricted applicability, in that they require us to assume that all treatment times are exponentially distributed with the same mean. As such, they could be applied in settings such as hip and knee surgery, where treatment durations are comparable for all patient groups (except for Emergency cases such as hip fractures, which are handled separately). The present model cannot be applied in an Emergency Department setting, where treatment times are clearly different for patients of the various acuity levels. (This case is the subject of ongoing work, for which substantial further analytical effort is required.)

The purposes of this note are two-fold. In the first instance, we wish to present the exact transform of the waiting time distribution for each class in the case where treatment times are identical. The second purpose is to carry out numerical investigations to assess the performance of the multi-server APQ model. Typically, for a multi-server system with two or three classes and KPIs with a doubling time benchmark (such as was seen for the lower classes under CTAS and ATS), we are interested in addressing questions such as which are the limiting KPIs, what are the optimal accumulation rates to assign, and what is the maximal traffic load that can be accommodated by a given number of servers.

The remainder of the paper is arranged as follows. In the next section we describe the model. Section 3 contains our derivation of the waiting time distribution for each class. A series of numerical investigations are reported in Section 4, where we also present a method for choosing the optimal value of the accumulation rates to satisfy given performance objectives in the two-class case. The final section of the paper gives conclusions and future research directions.

2. Description of the model

The model considered in this note is essentially that in [4,5], but with $c > 1$ servers, and it is restricted to the case of a common exponential service time distribution for all classes. Customers of class-k, $k = 1, 2, \ldots, K$ arrive to the queue according to a Poisson process with rate λ_k. If a server is free when the customer of class-k arrives, then that customer enters service immediately. Otherwise, they wait in the queue for service, accumulating priority at rate b_k where $b_1 > b_2 > \cdots > b_K$, so class-1 here is the highest priority class, and class-K the lowest. Thus a customer of type k arriving at time t will have accumulated priority $b_k(t' - t)$ by time t'. If all servers are busy, then at the time of the next service completion, the customer that enters service will be the one with the highest accumulated priority at that instant. The common exponential service time distribution has mean $1/\mu$ and Laplace Stieltjes Transform (LST) $\widetilde{B}(s) = \mu/(\mu + s)$. All inter-arrival times and service times are independent of one another. As in [5], throughout this note, the LST of a random variable with distribution function F will be denoted by $\widetilde{F}$.

In the interests of tractability, we restrict ourselves to the case where the service times are exponentially-distributed with a common mean. Whereas, in a single server queue, the commencement of service for a waiting customer occurs when the service of the preceding customer is completed, in a multi-server queue, it occurs when one of the servers becomes free. In the single-server case there are no ongoing services to worry about but, in the multi-server case, the future evolution of the queue will depend on the stage of service of those customers whose service is continuing.

Specifically we need to know the minimal residual service time among the continuing customers, in order to specify when the next customer can move into service. For non-exponential distributions, this task is tractable only when the number of servers is small, and the service time distributions are simple extensions of the exponential, such as Erlang distributions of low order, or mixtures of two exponentials. Even in the case where the other service times are exponential, but with class-dependent means, to characterize the minimum residual time we need to know the mix of continuing customers, and the different possibilities for such a mix make the analysis, at least, very complicated. Furthermore, it is at present unclear how the reordering of service times in an APQ setting affects the duration of the busy periods.

For these reasons, we have opted to solve the common exponential case first and, as we have already noted, it can be a good model for situations such as hip and knee surgery. We are pursuing the non-identical service time case in ongoing work, both analytically and, as in [7] via a near-perfect simulation approach which can be applied to this situation.

3. Waiting time distributions

We turn now to finding the distribution of the waiting time before service commences for the various classes. Let $\widetilde{W}^{(k)}(s)$ denote the Laplace transform of the stationary waiting time distribution for customers of class-k; $k = 1, 2, \ldots, K$. We begin by observing that the waiting time prior to service is strictly positive only if an arrival finds all c servers busy, and otherwise it is 0. Any priority mechanism that selects among waiting customers with service time requirements drawn from the same distribution will have no impact upon the chance that an arrival finds all of the servers busy, which can be identified from the corresponding $M/M/c$ queue.

With $C(A, c)$ being the probability that all servers are simultaneously busy in a stationary $M/M/c$ queue with $A = \lambda/\mu$ and $\lambda = \sum_{i=1}^{K} \lambda_i$, it immediately follows that

$$\widetilde{W}^{(k)}(s) = (1 - C(A, c)) + C(A, c)\widetilde{W}_{+}^{(k)}(s); \quad k = 1, 2, \ldots, K \quad (1)$$

where $\widetilde{W}_{+}^{(k)}(s)$ is the LST of the class-k waiting time distribution, conditional on it being positive, that is, conditional on a class-k customer arriving to find all servers busy.

Thus we need to find $\widetilde{W}_{+}^{(k)}(s)$, the LST of the class-$k$ waiting time distribution, conditional on an arrival of class-k finding all servers busy. In the following lemma we will denote this by $\widetilde{W}_{+}^{(k)}(s; \mu, c)$, to explicitly state the dependence of the results on the number of servers c and the common service rate μ for all classes.