CA4011 (2016-2017)
Continuous Assignment Specification

Part A (Simulation)

CONTENTS

1 Overview of continuous assessment

1.1 Marks breakdown

Part A of CA4011 continuous assessment involves simulation. Part B (to be specified later in a separate document) will be on Mathematical Programming. The continuous assessment part of this module is worth 30% of the overall module marks and these will be shared equally between Parts A and B. Thus,

> Part A is worth 50% of the CA mark or 15% of the overall module mark.
>
> Part B is worth 50% of the CA mark or 15% of the overall module mark.

1.2 Deadlines

The deadlines for submission of the work (through LOOP) are as follows:

> Part A: 23.00 on Monday March 20th
>
> Part B: 23.00 on Friday April 21st

1.3 Performance of Part A

This part of continuous assessment may be done individually or in pairs. Please inform the lecturer by email by Friday February 24th (2017) whether you are doing the assessment as an individual or with a partner. If the latter, specify your partner's name and student ID.

The main deliverable (in pdf format) is to be a report containing a concise description of the design or structure of your software simulation model, preferably with some diagrams. This should be followed by the results you obtained for each of the sub-tasks (see Section 2, below). The report should finish with a short conclusion section. The source code (the choice of language is up to you) should be included as an appendix.

Time-permitting, you may be asked to present a short informal demo of your system.

2 Specification of Part A

2.0 Introduction

The objective is to build a software simulation model of a system to which customers arrive to receive a service. There may be one or more servers to provide the service. The purpose of the model is to evaluate the performance of the system as the pattern of arrivals and services are varied.

In the following, the particular system described is a general medical practice but the model should be applicable to other analogous systems. For a medical practice, the customers are patients (or clients) and the servers are doctors. The overall goal is to model the system for a typical day in which the practice is open for patient arrivals between 09.00 and 17.30.

The task is broken into a number of sub-tasks of increasing complexity. However, the software should be structured as a unified system where users can specify input parameter values to define the different scenarios to be modelled.

In queueing systems, there are various measures of performance and these are essential results from your software model. Key measures of performance include

Average time a customer spends in the system (queueing & being served)

Average time a customer spends waiting for service (that is, in the queue)

Maximum time a customer spends in the system (queueing & being served)

Maximum time a customer spends waiting for service (that is, in the queue)

Proportion of time each server is idle

Average number of customers in the system (queueing & being served)

Average number of customers in the queue

In stochastic simulations (i.e. those involving some random elements) it is essential to provide for replications of experiments. This is necessary in order to provide a measure of the confidence that may be placed on the results obtained. Typically, there might be 100 or more replications and from these averages and standard deviations can be calculated as explained in lecture notes (see section 1.2 of the notes on simulation, for example). The number of replications to be applied should be an input parameter for the simulation system.

In all the following tasks, the distribution of service times should be assumed to be exponential. However, the distribution of arrivals varies according to the particular task.

2.1 Task 1: Compare random and scheduled patterns of customer arrival

This task builds on Section 1.2 of the lecture notes on simulation so familiarising with that section is a good point to start from.

As in that section assume there are two servers (doctors) who are identical in terms of how quickly they treat patients. For example, you might assume to begin with that each doctor can see 4 patients in one hour so that the service rate $\mu = 4$.

In section 1.2 of the notes, a **scheduled pattern of arrivals** was considered:

Arrivals at times 1, 2, … but with some uncertainty around the arrival time

To begin with[1], for the medical practice, you might assume arrivals scheduled at 09.05, 09.15, 09.25, … ,17.25 with the actual arrival points independently distributed around the scheduled arrival times points as mean and with standard distribution 5 minutes. In this case, the average arrival rate $\lambda = 6$ per hour.

Note: Following section 1.2 of the notes, this gives $\lambda/\mu = 6/4 = 3/2$ so that traffic intensity $\rho = \lambda/(2\mu) = ¾$.

(a) The first part of this task is to carry out the simulation, essentially following the approach of Section 1.2 of the notes. The output from the simulation should be estimates of the various performance parameters listed in section 2.0 (above). In addition, it would be of interest to estimate when the last patient leaves.

(b) The second part of the task is to replace the pattern of arrivals by a random pattern where the arrivals are Poisson distributed (equivalent to the inter-arrival times being exponentially distributed) with parameter λ. The service times are unchanged. Then the simulation should be performed again leading to estimates of the various performance parameters.

You should then compare the results of (a) and (b) and draw conclusions as to which system works better. It would be good to vary the values of λ and μ to check whether the conclusions hold good.

2.2 Task 2: Is one experienced server better than two novices?

This task builds on the work of Task 1. It is required to examine whether an experienced doctor (who can see for example 8 patients per hour) is more efficient than two slower doctors (who can each see, for example, 4 patients per hour).

Thus, the task to simulate a one server system and compare its performance with the previous two-server system. As for task 1 it would be best to vary the values of λ and μ to check whether any conclusions hold good.

For this task, you should at least apply the scheduled arrival pattern specified in Task 1 (though it would be interesting to see the results for the random pattern also).

[1] However, your software should provide for these to be input parameters to allow for other values to be used.

2.3 Task 3: More realistic pattern of service (with scheduled arrivals)

For this task it is sufficient to use the scheduled arrival pattern specified for Task 2. As in Task 1, you may assume two identical servers (though it would be very nice to have the capability of varying the number of servers).

Here, it is proposed to vary the server availability to a more realistic pattern where each server (doctor) has a break both morning and afternoon. For example, to begin with, you might specify:

> Doctor 1 takes a break for 30 minutes at 10.30 and at 14.30, or whenever she/he finishes with his current patient.
>
> Doctor 2 takes a break for 30 minutes at 11.30 and at 15.30, or whenever she/he finishes with his current patient.

Clearly the sample values given here should be replaced by input parameters in the software simulation.

The task then is to estimate the performance parameters and compare the performance of this somewhat more realistic system with that of (a) of Task 1.

2.4 Task 4: Customers not all the same – how best to manage?

In this task we assume that there are two different kinds of patient, regular ones whose service times are as indicated in the foregoing tasks, and new patients whose service times are much longer (though still exponentially distributed). For example, we might have that the average consultation time for a regular patient is 15 minutes but that for a new patient is 30 minutes. (These are just indicative values that should be parameterised in the software). It is also necessary to specify the proportion of regular and new patients – for example, one might specify that 5/6 are regular and 1/6 are new.

(a) The first part of this task is repeat the simulation of Task 3 with the regular and new patients intermixed. For example, on average every 6th patient might be new. Then compare the results with those for Task 3. As usual, it is important to vary the parameter values.

(b) The second part of the task is to perform a simulation in which the regular patients are all scheduled before the new ones. Then, compare the performance with the inter-mixed simulation (a).