Imperial College London
Department of Computing

Response Times in Healthcare Systems

Susanna Wau Men Au-Yeung

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London, September 2007

Abstract

It is a goal universally acknowledged that a healthcare system should treat its patients – and especially those in need of critical care – in a timely manner. However, this is often not achieved in practice, particularly in state-run public healthcare systems that suffer from high patient demand and limited resources. In particular, Accident and Emergency (A&E) departments in England have been placed under increasing pressure, with attendances rising year on year, and a national government target whereby 98% of patients should spend 4 hours or less in an A&E department from arrival to admission, transfer or discharge.

This thesis presents techniques and tools to characterise and forecast patient arrivals, to model patient flow and to assess the response-time impact of different resource allocations, patient treatment schemes and workload scenarios.

Having obtained ethical approval to access five years of pseudonymised patient timing data from a large case study A&E department, we present a number of time series models that characterise and forecast daily A&E patient arrivals. Patient arrivals are classified as one of two arrival streams (walk-in and ambulance) by mode of arrival. Using power spectrum analysis, we find the two arrival streams exhibit different statistical properties and hence require separate time series models. We find that structural time series models best characterise and forecast walk-in arrivals, but that time series analysis may not be appropriate for ambulance arrivals; this prompts us to investigate characterisation by a non-homogeneous Poisson process.

Next we present a hierarchical multiclass queueing network model of patient flow in our case study A&E department. We investigate via a discrete-event simulation the impact of class and time-based priority treatment of patients, and compare the resulting service-time densities and moments with actual data. Then, by performing bottleneck analysis and investigating various workload and resource scenarios, we pinpoint the resources that have the greatest impact on mean service times.

Finally we describe an approximate generating function analysis (AGFA) technique which efficiently approximates the first two moments of customer response time in class-dependent priority queueing networks with blocking. This technique is applied to the model of A&E and the results compared with those from simulation. We find good agreement for mean service times especially when minors patients are given priority.

Acknowledgements

I would like to thank the following people:

- My supervisor, Dr. William Knottenbelt, for his help, guidance and enthusiasm.

- My second supervisor, Prof. Peter Harrison, for his time, energy and expertise.

- The members of the Analysis, Engineering, Simulation and Optimisation of Performance (AESOP) research group. In particular: Tony Field, Jeremy Bradley, Ashok Argent-Katwala, Uli Harder, David Thornley, Nicholas Dingle, Harini Kulatunga, Helen Yu Zhang, Tamas Suto and Abigail Lebrecht.

- Emma McCoy, Azeem Majeed and Nalan Gulpinar whose assistance was much appreciated.

- The members of staff at our case study hospital and associated institutions.

- The Engineering and Physical Sciences Research Council (EPSRC) for providing me with the funding to do my PhD.

- My family and friends for their love, support and encouragement throughout my PhD.

‘But they are useless. They can only give you answers.’

Pablo Picasso, on computers

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In recent years, there has been much public concern regarding patient waiting times in the National Health Service (NHS). For example, in a 2004 King's Fund report, improved waiting times for patients in Accident and Emergency departments and for cancer and cardiac patients were identified as two of the public's top four priorities for public healthcare in the UK [46]. In response, the NHS has seen unprecedented levels of government investment, rising from £65.4bn in 2002 to £87.2bn in 2006. By 2008, predicted spending will total £105.6bn. Despite this increased level of investment, the state of the NHS is still of great public concern, and the question being increasingly asked is whether this money has been well spent [47].

Healthcare systems are complex processing systems with stringent response time-based targets. These targets relate not only to mean response times, but also to variability of response times. In particular Accident and Emergency (A&E) departments in England are subject to a national target, whereby 98% of patients should spend 4 hours or less in the department from arrival to admission, transfer or discharge.

In this context it is important to develop appropriate performance models and other systematic procedures for identifying the most effective use of resources and locating non-obvious bottlenecks. In addition these models can give insights into how different customer processing strategies affect moments and distributions of response time.

Since 2002, A&E attendances have been rising year on year. Together with some departments being closed or downgraded, existing A&E departments have been placed under increasing pressure [36]. Despite this increase in workload, the introduction of the four hour waiting time target means that patients still need to be seen and treated quickly and safely. Although the vast majority of Acute trusts have managed to achieve this target at a 95% threshold (assisted by innovations identified by the Emergency Services Collaborative such as "see and treat" schemes for minor injuries and near-patient testing [31, 32]), in 2004/2005 44% of Acute trusts failed to meet the 98% target [58]. This reflects the difficulty that many departments are experiencing in making further efficiency improvements [57].

Several studies have been made of healthcare systems in general [33, 35, 100] and A&E departments in particular [16, 27, 68, 73, 75, 30, 74, 93]. However, these studies have had limited success and subsequent impact for two main reasons. Firstly, there has been a lack of sophistication in the models used (mostly simple discrete-event simulations and very high-level queueing models), and in the analysis techniques applied (mostly aimed at computing simple resource based measures such as mean utilisations and mean response times). In this thesis, this shortcoming will be addressed through the creation of a detailed patient flow model of a case study A&E department. Using this model, sophisticated performance measures such as the higher moments and densities of patient service time will be derived. Secondly, existing models often remain unvalidated against real waiting time data, or are parameterised using small quantities of manually-collected data. We now have a prime opportunity to take advantage of the detailed patient waiting time data automatically collected by all A&E departments in England to monitor compliance with government targets (describing time of arrival, various treatment times and time of discharge for every patient). We have applied for and obtained research project status and ethical approval to access pseudonymised patient timing data (that is non-patient identifiable data which has been tagged by a unique reference number, in order to track the patient through stages of treatment) at our case study department for the past five years. The availability of this data enables us to both parameterise and validate models of patient flow. This data will also be used to fit models of patient arrivals and then subsequently to ascertain the accuracy of model forecasts.

1.2 Objectives

The aims and objectives of this thesis are:

- To characterise and forecast patient arrivals into our case study A&E department. This will also provide realistic arrival workloads for subsequent models and simulations of A&E.
- To characterise and model patient flow within our case study A&E department, using actual patient timing data to help parameterise our model.
- To use our model of patient flow to investigate the response time impact of:
  - the introduction of the four hour government target,
  - different patient priority treatment schemes, and
  - different workload and resource scenarios.
- To develop efficient analytical techniques that can be used to approximate the mean and variance of response time in the patient flow model.

1.3 Contributions

This thesis presents techniques and tools to characterise and forecast patient arrivals, to model patient flow and to assess the response-time impact of different resource allocations, patient treatment schemes and workload scenarios in hospital departments.

1.3.1 Time Series Models of Patient Arrivals

We show that walk-in and ambulance patient arrivals into an A&E department exhibit differing statistical properties and so require to be modelled separately. We demonstrate that walk-in arrivals exhibit a strong seven day seasonality that is best modelled with a structural time series model. Such a time series model provides one to six day ahead forecasts with good predictive power. However, we experience less success with our ambulance arrivals models. The poor performance of the ambulance arrival time series

model forecasts may be because the ambulance arrivals do not exhibit very strong periodicities or other regularity; nor do they appear to belong to familiar classes of stochastic processes. We also demonstrate that arrivals into an A&E department by hour varies predictably, with weekdays exhibiting similar hourly arrival patterns; as do weekends. Finally, we find that there is no significant relationship between weather-related variables and patient arrivals into our case study department.

1.3.2 Patient Flow Modelling

We create a hierarchical Markovian multiclass queueing network model of patient flow and parameterise it using electronic patient record data. We implement a discrete-event simulation of this model in Java and compare the resulting service time densities with those observed in the actual A&E. Having investigated the effects of different patient priority schemes, we find the impact of the introduction of the 4 hour waiting time target has been similar to a move from a system in which majors (seriously ill/injured) patients are given priority, to a system in which minors (less seriously ill/injured) patients are given priority treatment. We also gain some insights into how the system behaves when the workload levels are varied. We find that for low to medium workloads, mean service times for ambulance arrivals benefit from a system under majors priority, but under high workloads both arrival types perform better under a minors priority scheme. The nature of the model allows us to obtain performance measures at the resource level, allowing us to pinpoint the main bottlenecks in the system and to quantify the impact of various resource allocations.

1.3.3 Efficient Approximate Response Time Analysis

Finally, we present an efficient and novel approximate generating function analysis (AGFA) technique which approximates the first two moments of response time in a multiclass queueing network with non-pre-emptive priority. We compare the results from the AGFA method with corresponding results obtained by simulation. We show that the technique works well for mean response times although discrepancies are noted when the system modelled starts to become saturated under high workloads. The

corresponding standard deviations show generally adequate agreement with simulation results but (not atypically for this kind of technique) are less accurate.

1.4 Thesis Outline

The remainder of this thesis is set out as follows:

Chapter 2 describes the background theory to the work presented in this thesis. An overview of time series analysis is provided. This is followed by an introduction to the theory of Markov processes and queueing networks. The properties of the Laplace transform are discussed before considering a response time analysis method based on numerical Laplace transform inversion. Next we present the prior work in the area of healthcare modelling, focusing on the current work on patient arrivals and A&E modelling. Finally, we provide a description of our case study A&E department.

Chapter 3 presents a number of different time series models used to characterise and forecast daily arrivals into our case study A&E department. We describe a rolling six week average model, an auto-regressive model, and a structural time series model. In each case, forecasts for each of these models are presented and compared with observed arrivals. We also experiment with the use of non-homogeneous Poisson processes to further characterise ambulance arrivals. Next we present the hourly breakdown of patient arrivals into the department. Finally, the impact of weather-related variables on patient arrivals is investigated.

Chapter 4 describes a multiclass Markovian queueing network model of patient flow in our case study A&E department. Using patient timing data to help parameterise the model, we implement a discrete-event simulation, from which we obtain moments and probability density functions of patient response time and associated utilisations. We investigate both class-based and time-based patient handling priority schemes and compare the resulting response time moments and densities with the actually observed quantities. Finally, we investigate the impact on patient response time and resource utilisations of implementing various workload and resource availability scenarios.

Chapter 5 presents an approximate generating function analysis (AGFA) technique, which approximates the Laplace transform of the probability density function of customer response time in networks of queues with class-based priorities. From the approximated Laplace transform, we derive the first two moments of customer response time. This technique is applied to the queueing network model of patient flow in an A&E department, as introduced in Chapter 4, to obtain the mean and standard deviation of total patient service time. These AGFA moments are then compared with the results from the discrete-event simulation.

Chapter 6 concludes the thesis by providing a summary and an evaluation of the work presented. This chapter also discusses possible applications and opportunities for future work.

Appendix A presents the *R* code used to create and fit the time series models detailed in Chapter 3 and the use of these models to derive forecasts.

Appendix B presents plots of walk-in and ambulance arrivals by hour, for each day of the week and for each financial year between 2002 and 2007.

Appendix C presents detailed patient flow diagrams for self-referred patient arrivals (additionally illustrating the patient pathways for minors patients), GP-referred patient arrivals and ambulance patient arrivals (additionally illustrating the patient pathways for majors patients).

Appendix D presents tables of the mean and standard deviation of patient service time under differing workloads and patient class-based priority schemes, as calculated via simulation and by the AGFA technique.

Appendix E presents code for the Mathematica implementation of the approximate generating function analysis (AGFA) technique and its application to the adapted model of patient flow.

1.5 Publications and Statement of Originality

I declare that this thesis was composed by myself, and that the work that it presents is my own, except where otherwise stated.

The publications referred to below arose from the work carried out during the course of this PhD:

- **European and Simulation Modelling Conference (ESM 2006)** [10] presents a hierarchical multiclass Markovian queueing network model of patient flow in the A&E department of a major London hospital. We solve for moments and probability density functions of patient response time using discrete-event simulation under different patient priority schemes and compare the resulting response time moments and densities with real data. The queueing network model of A&E presented in Chapter 4 is based on this paper.

- **International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2007)** [11] presents an approximate generating function analysis (AGFA) technique that provides an efficient analytical way to approximate the mean and variance of response time in networks of multiclass queues with blocking and class-dependent priorities. This technique is later applied to a hierarchical queueing network model of an A&E department as a case study. The AGFA technique presented in Chapter 5 is based on this paper. This is joint work with Peter Harrison.

- **Emergency Medicine Journal** [9] (submitted for publication) describes structural time series models of daily patient arrivals to a case study A&E department. Patient arrivals are aggregated by day and then allocated to one of two arrival streams (walk-in or ambulance) by mode of arrival. Using the first four years of patient arrivals data as a "training" set, a structural time series model is fitted to characterise each arrival stream. These models are used to forecast walk-in and ambulance arrivals for one to seven days ahead and then compared with the observed arrivals given by the remaining one year of "unseen" data. Material from this paper appears in Chapter 3. This is joint work with Uli Harder and Emma McCoy.

Chapter 2

Background and Related Work

2.1 Introduction

This chapter presents the background theory underlying the work in this thesis. We also discuss related work and existing approaches to healthcare modelling. We begin with a general overview of time series analysis. This is followed by a discussion of Markov processes and queueing theory, before we consider a recent response time analysis method based on numerical Laplace transform inversion. Next we present current work in healthcare modelling, especially in the area of patient arrivals and A&E modelling. This chapter concludes with a description of our case study A&E department.

2.2 Time Series Analysis

A *time series* is a sequence $x_1, \ldots, x_T$ of T observations taken sequentially in time. Examples occur in a wide range of fields such as economics, engineering, meteorology, geophysics and business. Data sets which appear as time series include a daily series of share prices, a weekly series of sales figures and a monthly series of rainfall observations. Due to this widespread occurrence, the theory of times series analysis has been extensively developed and is discussed in many books including [19, 21, 25, 54, 56].

There are two main objectives in the study of time series: characterisation and modelling. The aim of characterisation is to obtain an insight into the nature of the time

series and to summarise its main properties. The modelling of a time series is important as it enables forecasts of future values to be made. In a time series model, the changes in x_t are described only by current and past values, and forecasts can be made via extrapolation.

2.2.1 Stochastic Processes

In almost all cases, future values of a time series will be influenced by some unknown factors and will only be partly determined by past observations. For this reason, a time series can be thought of as the realisation of a *stochastic process*.

Definition 2.1 (Stochastic Process) *A stochastic process* $\mathbf{X}$ *is a family of random variables* $\{X_t \in \mathbf{\Omega} \mid t \in T\}$*, where each is defined on some sample space* $\mathbf{\Omega}$ *for a parameter space* T*.*

Generally, T is discrete time and $\mathbf{\Omega}$ is the set of values (also known as the state space) that each X_t may take. The observed value of a time series at time t (x_t) is a single observation of the random variable at time t (X_t). Thus the observed time series is just one example of an infinite set of time series which might have been observed. This infinite set is called the *ensemble*, with every member of the ensemble a possible *realisation* of the stochastic process.

The moments of a stochastic process are a useful way of summarising the process, and are defined with respect to the distribution of the random variables $X_1, \ldots, X_T$. The mean (μ_t) of the process at time t is:

$$\mu_t = E(X_t), \quad t = 1, \ldots, T \tag{2.1}$$

which can be interpreted as the average value of X_t over all possible realisations.
The variance (σ_t^2) at time t is:

$$\sigma_t^2 = E[(X_t - \mu_t)^2], \quad t = 1, \ldots, T \tag{2.2}$$

Finally, the covariance $\gamma(t, t+\tau)$ between X_t and $X_{t+\tau}$ is given by:

$$\gamma(t, t+\tau) = E[(X_t - \mu_t)(X_{t+\tau} - \mu_{t+\tau})], \quad t = 1, \ldots, T-\tau \tag{2.3}$$

2.2.2 Autocorrelation Function

An important guide to the properties of a time series is provided by the sample autocorrelation coefficients. The autocorrelation between observations at distance k apart, known as the autocorrelation coefficient at lag k, and denoted by r_k, is given by:

$$r_k = \frac{\sum_{t=1}^{T-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{T}(x_t - \bar{x})^2} \tag{2.4}$$

where $\bar{x} = \sum_{t=1}^{T} x_t/T$ is the overall mean of the observed time series.

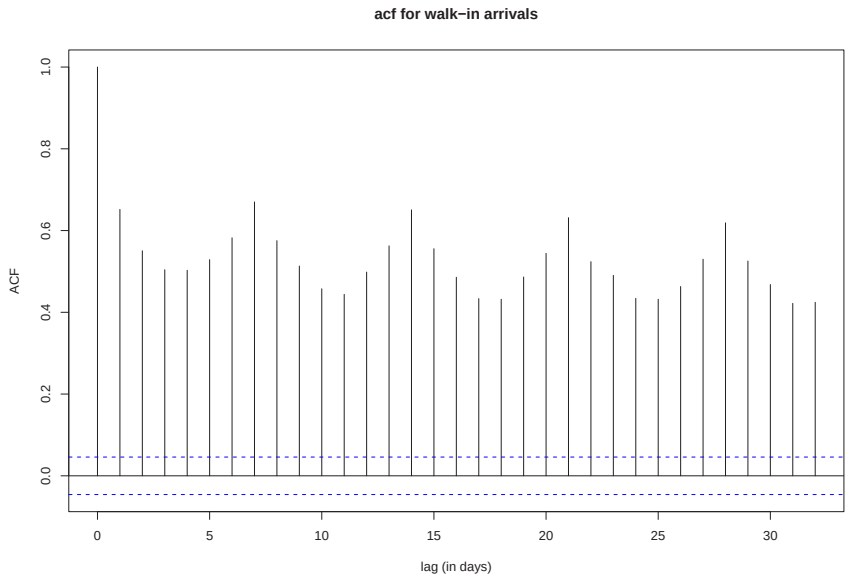


Figure 2.1: Example autocorrelation function.

A plot of r_k against non-negative values of k gives the *autocorrelation function* (also known as the correlogram). Fig. 2.1 shows an example of an autocorrelation function (acf); here this is the acf of the walk-in patient arrivals to our case study department. The acf indicates that the time series has a seven day seasonality.

2.2.3 Stationary Processes

Most of the theory of time series is concerned with the analysis of *stationary* time series. This is the case when fitting some of the most often used time series models including moving average, auto-regressive and auto-regressive integrated moving average (ARIMA) models. Intuitively a time series is stationary if there is no systematic change in mean and variance over time and if strictly periodic variations have been removed.

Definition 2.2 (Stationary Time Series) *A time series $x_1, \ldots, x_T$ is (weakly) stationary if*

- *$E(x_t) = \mu(t)$ is independent of $t = 1, \ldots, T$, and*
- *$Cov(x_t, x_{t+\tau}) = \gamma(t, t+\tau)$ is independent of t for each τ.*

In order to determine whether a time series is stationary we will use the *Kwaitowski, Phillips, Schmidt and Shin (KPSS)* stationarity test [67]. In the KPSS test, a time series x_t with T observations is decomposed into the sum of a deterministic trend, a random walk and a stationary error according to the following regression model:

$$x_t = r_t + \beta t + \epsilon_t$$

r_t is a random walk, i.e.

$$r_t = r_{t-1} + u_t$$

where the u_t are normally distributed random variables with mean zero and variance σ_u^2, βt is the deterministic trend and ϵ_t is a stationary error term.

The null hypothesis is that x_t is stationary, that is $\beta = 0$. To test if this is the case, we calculate the residuals e_t from a regression of x_t on the intercept only, that is, $e_t = x_t - \bar{x}$, where $\bar{x}$ is the overall mean of the series x_t. The partial sums denoted S_t, of the e_t are:

$$S_t = \sum_{j=1}^{t} e_j$$

If σ^2 is the long-run variance of e_t, then σ^2 can be estimated using the Newey-West estimator [85]:

$$\hat{\sigma}^2(p) = \frac{1}{T}\sum_{t=1}^{T} e_t^2 + \frac{2}{T}\sum_{j=1}^{p} w_j(p) \sum_{t=j+1}^{T} e_t e_{t-j}$$

where p is the truncation lag (here we use $p = \lfloor \frac{5\sqrt{T}}{7} \rfloor$), and $w_j(p)$ is a Bartlett window [13] given by $w_j(p) = 1 - \frac{j}{p+1}$.

The KPSS test statistic is then given by:

$$KPSS = T^{-2} \sum_{t=1}^{T} \frac{S_t^2}{\hat{\sigma}^2(p)} \tag{2.5}$$

Critical values for this test statistic are then interpolated from Table 1 of [67].

If a time series is found to be non-stationary, it is often necessary to transform this time series in order to achieve stationarity. The main ways of transforming a time series are:

- **Detrending** - a time series may be made stationary by removing any obvious trend (e.g. a linear trend) from the time series.
- **Logarithms or square roots** - a time series may be made stationary by taking logarithms or the square root of the data.
- **Differencing** - a time series may be made stationary by differencing; that is, given a time series x_t, we create the new series $y_t = x_{t+1} - x_t$. This known as the first-order difference and is usually sufficient; however, this process may be repeated n times to obtain the n^{th}-order difference.

2.2.4 Method of Maximum Likelihood

Once the class of time series model to be utilised has been decided on, we next need to estimate the model parameters that provide the best fit to the data. This is often done via the method of maximum likelihood which also provides an efficient method for quantifying uncertainty through confidence bounds. The idea is to determine the parameters that maximize the probability (likelihood) of the observed (past) data.

Suppose that we have a data set of n observations $\mathbf{x} = (x_1, \ldots, x_n)$, with which we associate an n-dimensional random variable, whose known probability distribution $\mathbb{P}(\mathbf{x} \mid \boldsymbol{\psi})$ depends on some unknown set of parameters $\boldsymbol{\psi}$. In our case, this set of parameters refers to the parameters of our time series model. Before the data set was available $\mathbb{P}(\mathbf{x} \mid \boldsymbol{\psi})$ will associate a density with each possible different outcome of $\mathbf{x}$, for fixed $\boldsymbol{\psi}$. However, after observing the data, we can instead vary the values of $\boldsymbol{\psi}$ in order to obtain the probability of observing (the now fixed) $\mathbf{x}$. This is known as the *likelihood function* $L(\boldsymbol{\psi} \mid \mathbf{x})$. Often it is more convenient to work with the log-likelihood function, $\log L(\boldsymbol{\psi})$ since in this way products can be represented by sums.

The values of the parameters that maximise the likelihood function, or equivalently the log-likelihood function, are called the *maximum likelihood estimates*. These are found by solving the *likelihood equations* that arise from differentiating the log-likelihood function with respect to the unknown parameters, and then setting the first derivative to zero, i.e.

$$\frac{\partial \log L}{\partial \boldsymbol{\psi}} = 0 \tag{2.6}$$

The second derivatives of the log-likelihood function can be used to calculate the approximate standard errors for the parameter estimates and from these confidence intervals. As a rule, the likelihood equations are non-linear and so the maximum likelihood estimates must be found by an iterative procedure such as the *Newton-Raphson* method [89].

2.2.5 Checking the Time Series Model Fit

Once a time series model has been fitted, we need to check how well this model describes the data. This is done by analysing the residuals (the differences between the observed and fitted values) of the model, to determine any systematic trends. This is generally done via graphical methods; as quoted from [19]: "It cannot be too strongly emphasized that *visual inspection of a plot of the residuals themselves* is an indispensable first step in the checking process." The model residuals are plotted against the following:

- **time** to check for outliers and any correlation with time,

- **fitted values** to check that there is no obvious pattern in the spread of the residuals, and
- **the normal distribution** with the same mean and standard deviation as the residuals, to verify that the residuals are approximately normally distributed.

The autocorrelation function (acf) of the residual values is also calculated to determine if there is correlation within the residuals, which would indicate that there is some structure not yet incorporated into the model. We also perform the *Ljung-Box* test for independence [19, 55], which instead of analysing each distinct r_k (the autocorrelation coefficient at lag k), considers the first M values of the acf all at once. The test statistic Q is given by:

$$Q = (T(T+2)) \sum_{k=1}^{M} \frac{r_k^2}{T-k} \tag{2.7}$$

where T is the number of observations in the fitted time series, r_k the autocorrelation coefficient at lag k and M the number of lags being tested. In this thesis, we use $M = 30$ when referring to the Ljung-Box test. The null hypothesis is that the residuals are random and this is rejected if $Q > \chi^2_M$ at $p \leq 0.05$ where χ^2_M is the χ^2 distribution with M degrees of freedom.

2.2.6 Quality of Forecast Measures

To indicate the quality of a set of time series model predictions a number of quality of forecast metrics may be used. These include mean bias, root mean square error (RMSE) and the Pearson product-moment correlation coefficient between observed and predicted values (r). Graphical procedures such as scatterplots are also used to show visually the quality of the forecast.

Mean Bias The mean bias of a set of predictions is used to give an indication of whether the time series model forecasts tend to over or under estimate. The bias at time t is given by $b_t = \hat{x}_t - x_t$ where $\hat{x}_t$ is the predicted value for time t and x_t the observed value at time t. Once the set of biases have been calculated for all t, the mean is taken to obtain the mean bias.

Root Mean Square Error (RMSE) The RMSE is a commonly-used forecast accuracy metric, quantifying the amount by which the forecast differs from the actual values. The root mean square error (RMSE) is calculated as follows:

$$RMSE = \left(\frac{1}{n} \sum_{t=1}^{n} b_t^2 \right)^{1/2} \tag{2.8}$$

where n is the number of predictions and b_t the bias at time t.

Pearson Product-Moment Correlation Coefficient The Pearson product-moment correlation coefficient (r) is a measure of the tendency of two sets of variables X and Y to increase or decrease together (known as correlation). In the context of forecasting, X represents observed values and Y predicted values. It is calculated as follows:

$$r = \frac{\sum Z_x Z_y}{n-1} \quad \text{where} \quad Z_x = \frac{x - \mu_x}{\sigma_x} \tag{2.9}$$

where n is the number of variables (the same in each set) and μ_x is the mean and σ_x is the standard deviation of X.

The coefficient ranges from -1 to 1. A value of 1 shows that a linear equation describes the relationship exactly, with Y increasing with X. A score of -1 shows an exact inverse relationship, with Y decreasing as X increases. A value of 0 shows that there is no linear relationship between the variables.

Scatterplots A scatterplot displays values for two sets of variables as a collection of points, each having one co-ordinate on the horizontal axis and one on the vertical axis. The resulting pattern indicates the type and strength of the relationship between two or more variables.

A scatterplot shows various kinds of relationships, including positive and negative correlation and no relationship. Fig. 2.2 shows an example scatterplot comparing a set of model forecasts (on the y-axis) against actual observations (on the x-axis). We see that there is a positive correlation between the observed and predicted values.

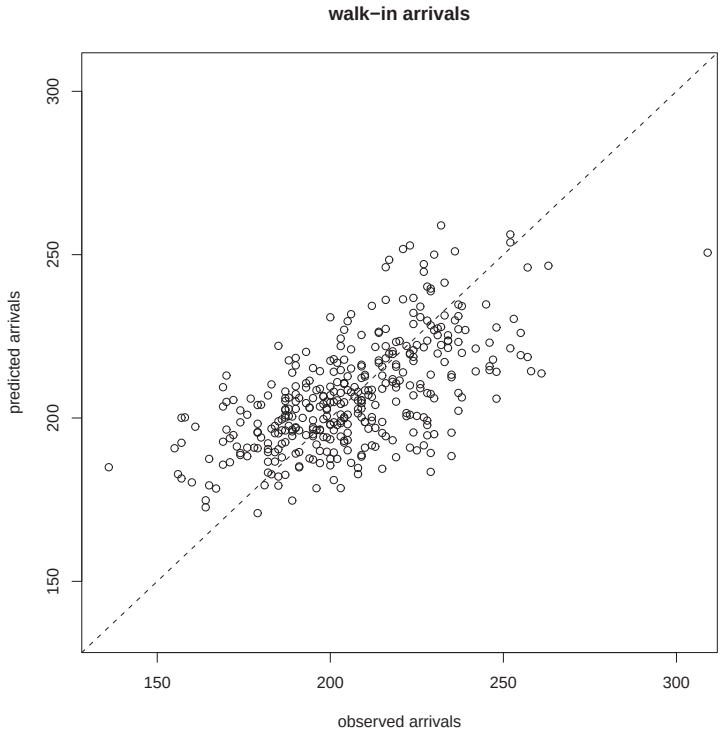


Figure 2.2: Example scatterplot.

2.3 Markov Processes

A Markov process is a class of stochastic process (as defined in Section 2.2.1) which satisfies the *Markov property*. For a more in depth discussion of Markov processes, see for example [53, 14, 82, 17].

Consider a stochastic process X defined on state (sample) space $\mathbf{\Omega}$ and parameter space time T.

Definition 2.3 (Markov Property) *Given the state of the process $X_t = x_t$ at time t, the probability distribution of any future state, $X_{t+s} = x_{t+s}$ at time $t + s$, $s > 0$ is dependent on s only.*

Intuitively this means the future states of the process from time t onwards are independent of the states before t. This means in order to predict (probabilistically) the future behaviour of the process, it is sufficient to know only the current state and not the past states.

We consider a Markov processes with discrete state space, that is where the values in the state space ($\mathbf{\Omega}$) of X_t are finite or countably infinite. If the time parameter is also discrete then the process is known as a *Markov chain* or *Discrete-time Markov Chain*. If the time parameter is continuous, then the process is know as a *Markov Process* or *Continuous-time Markov Chain*.

2.3.1 Discrete-time Markov Chains

Let $X = \{X_n \mid n = 0, 1, \ldots\}$ be an integer valued discrete-time Markov chain (DTMC), $X_i \geq 0, X_i \in \mathbb{Z}, i \geq 0$. The Markov property states that given the state of X at time n, its state at time $n+1$ is independent of the states at times $0, 1, \ldots, n-1$; that is:

$$\mathbb{P}(X_{n+1} = j \mid X_0 = x_0, X_1 = x_1, \ldots, X_n = x_n) = \mathbb{P}(X_{n+1} = j \mid X_n = x_n) \quad (2.10)$$

Hence the evolution of the DTMC is completely described by the *one-step transition probabilities* $p_{ij}(n)$ that the DTMC will move to state j at time $n+1$, given that it is in state i at time n:

$$p_{ij}(n) = \mathbb{P}(X_{n+1} = j \mid X_n = i) \quad \text{for} \quad i, j, n = 0, 1, \ldots \quad (2.11)$$

We assume that the one-step transition probabilities are *time homogeneous*, that is independent of time n:

$$p_{ij}(n) = p_{ij} \quad \text{for} \quad i, j, n = 0, 1, \ldots \quad (2.12)$$

Therefore, a time homogeneous DTMC defines a *transition probability matrix* P, containing all the one-step transition probabilities:

$$P = \begin{pmatrix} p_{00} & p_{01} & p_{02} & \cdots \\ p_{10} & p_{11} & p_{12} & \cdots \\ \vdots & \vdots & \vdots & \\ p_{i0} & p_{i1} & p_{i2} & \cdots \\ \vdots & \vdots & \vdots & \end{pmatrix} \quad (2.13)$$

where the dimension of P is the number of states in $\mathbf{\Omega}$ and, since the DTMC must be in some state at any observation instant, all the rows of P sum to one. Conversely, for any real matrix P such that $p_{ij} \geq 0$ and $\sum_j p_{ij} = 1$ (known as a *stochastic matrix*), one can construct a DTMC which has P as its transition matrix.

We now consider transitions made in two steps where:

$$p_{ij}^{(2)} = \mathbb{P}(X_{n+2} = j \mid X_n = i) \quad \text{for} \quad i, j, n = 0, 1, \ldots \tag{2.14}$$

In order to move from state i to state j the DTMC has to move to some intermediate state k; hence:

$$\begin{aligned} p_{ij}^{(2)} &= \sum_{k \in \mathbf{\Omega}} \mathbb{P}(X_{n+2} = j \mid X_n = i, X_{n+1} = k) \mathbb{P}(X_{n+1} = k \mid X_n = i) \\ &= \sum_{k \in \mathbf{\Omega}} \mathbb{P}(X_{n+2} = j \mid X_{n+1} = k) \mathbb{P}(X_{n+1} = k \mid X_n = i) \\ &= \sum_{k \in \mathbf{\Omega}} p_{ik} p_{kj} \end{aligned} \tag{2.15}$$

where the second equality uses the Markov property and the third equality is a result of X being time homogenous. Note that $p_{ik}p_{kj}$ is the ij^{th} element of P^2. Similarly by induction we obtain the s-step transition probabilities:

$$p_{ij}^{(s)} = \mathbb{P}(X_{n+s} = j \mid X_n = i) \quad (s \geq 1) \tag{2.16}$$

The matrix of these probabilities, is given by:

$$(p_{ij}^{(s)}) = P^{(s)} = P^s \tag{2.17}$$

This result may in principle be used to compute the probabilistic behaviour of a DTMC over any finite period of time. However, this can become computationally intractable for more complex models. Therefore there is a need to determine the long term behaviour of the DTMC.

We define the probability of being in state j at time s after starting in state i at time 0, denoted $\pi_{ij}^{(s)}$ as:

$$\pi_{ij}^{(s)} = \mathbb{P}(X_s = j \mid X_0 = i) \tag{2.18}$$

Under certain conditions, the more steps the DTMC makes, the less it matters what state it was in when it started. When the observation instant is infinitely removed from the starting point, the probability π_j of finding the DTMC in state j, is independent of the initial state:

$$\pi_j = \lim_{n\to\infty} \mathbb{P}(X_n = j \mid X_0 = i) = \lim_{n\to\infty} p_{ij}^{(n)} = \lim_{n\to\infty} \pi_{ij}^{(n)} \tag{2.19}$$

Definition 2.4 (Stationary Probability Distribution) *The stationary probability distribution is defined in terms of P, the one-step transition probability matrix of a DTMC, and the vector $\mathbf{z}$ whose elements z_i denote the probability of being in state i. The vector $\mathbf{z}$ is a probability distribution:*

$$z_i \in \mathbb{R}, \quad 0 \leq z_i \leq 1 \quad \text{and} \quad \sum_i z_i = 1$$

$\mathbf{z}$ is said to be a stationary distribution if and only if $\mathbf{z}P = \mathbf{z}$.

When the limiting probabilities π_j exist, and add up to 1, they are referred to as the *steady-state distribution* of the DTMC. The theory of whether a steady-state distribution exists and of determining it when it does requires the following definitions:

Definition 2.5 (Irreducible DTMC) *A DTMC is irreducible if every state is reachable from every other state in one or more transitions. If this is not the case, the DTMC is said to be reducible.*

The states in a DTMC can be distinguished as being either recurrent or transient. If $f_j^{(m)}$ is the probability of leaving state j and then first returning to it in m transitions, it follows that the probability of ever returning to state j is:

$$f_j = \sum_{m=1}^{\infty} f_j^{(m)}$$

If $f_j = 1$ then it is certain that we will return to state j at some point in the future and so j is said to be *recurrent*. Otherwise, states which are not recurrent are called *transient*.

Definition 2.6 (Periodic states) *A state j is said to be periodic, with period m (where $m > 1$), if the consecutive returns to j occur only at multiples of m steps:*

$$\mathbb{P}(X_{n+s} = j \mid X_n = j) = 0 \quad \text{if} \quad s \neq km \quad \text{for some} \quad k \geq 1$$

Note that a periodic state j is also recurrent. If there is no integer $m > 1$ which satisfies the above equation, then j is said to be *aperiodic*. If the DTMC is irreducible, then either all its states are periodic, with the same period, or all of them are aperiodic. The DTMC itself is then said to be periodic or aperiodic respectively. If an irreducible DTMC has at least one state to which it can return in a single step, then it is aperiodic.

From the probability $f_j^{(m)}$ of returning to state j in m steps, we define the *mean recurrence time* of state j, that is the average number of steps needed to return to state j for the first time after leaving it, denoted by M_j, as:

$$M_j = \sum_{m=1}^{\infty} m f_j^{(m)} \tag{2.20}$$

Further a state j is said to be *recurrent null* if $M_j = \infty$, whereas it is *recurrent non-null* if $M_j < \infty$. An irreducible DTMC can only have recurrent null states if the number of states are infinite.

We can now state the following result:

Theorem 2.1 *If $X = \{X_n \mid n = 0, 1, \ldots\}$ is an irreducible, aperiodic and recurrent DTMC, then the steady-state (limiting) probabilities π_j, exist and are given by :*

$$\pi_j = \frac{1}{M_j} \quad \text{for} \quad j = 0, 1, \ldots \tag{2.21}$$

Where M_j is defined in Equation 2.20.

If all states of X are are recurrent null then $\pi_j = 0$ for all j. If all states are recurrent non-null then $\pi_j > 0$ for all j. If X is finite, then it is recurrent non-null.

In order to calculate the quantities M_j and hence the probabilities π_j we have the following result, known as the *steady-state theorem*:

Theorem 2.2 *For a finite, irreducible and aperiodic DTMC, with N states, the values of the steady-state probabilities π_j are uniquely determined by the equations:*

$$\pi_j = \sum_i \pi_i p_{ij} \tag{2.22}$$

subject to

$$\sum_i \pi_i = 1 \tag{2.23}$$

Equations 2.22 are referred to as the *balance equations* of X, while Equation 2.23 is the *normalising equation*. Introducing the row vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$, Equations 2.22 can be written in matrix notation as:

$$\boldsymbol{\pi} = \boldsymbol{\pi} P \tag{2.24}$$

Thus the steady-state analysis of a system modelled by an irreducible and aperiodic DTMC largely consists of solving the corresponding balance and normalising equations. The complexity of this task depends on the size and structure of the one-step transition probability matrix P. Note that the fact that a DTMC has a stationary probability distribution does not imply that it has a steady state distribution.

2.3.2 Continuous-time Markov Chains

The continuous-time analogue of a DTMC, is which transitions can occur at arbitrary points in time, is known as a continuous-time Markov chain (CTMC). Let $X = \{X(t) \mid t \geq 0\}$, with $X(t) \in \boldsymbol{\Omega}$, where $\boldsymbol{\Omega}$ is a countable set, be a CTMC. By the Markov property, the path followed by X after given a given moment t, depends only on the state at that moment $X(t)$, and not on the past behaviour:

$$\mathbb{P}(X(t) = x \mid X(t_0) = x_0, \ldots, X(t_n) = x_n) = \mathbb{P}(X(t) = x \mid X(t_n) = x_n) \tag{2.25}$$

for any sequence $t_0, t_1, \ldots, t_n$ such that $t_0 < t_1 < \ldots < t_n$ and $t > t_n$.

A CTMC is said to be *time homogeneous* if the right hand side of Equation 2.25 does not depend on the moment of observation.

The *transition probability functions* $q_{ij}(s)$ of a time homogenous CTMC are analogous to the s-step transition probabilities $p_{ij}^{(s)}$ of a DTMC as defined in Equation 2.16:

$$q_{ij}(s) = \mathbb{P}(X(t+s) = j \mid X(t) = i) = \mathbb{P}(X(s) = j \mid X(0) = i) \tag{2.26}$$

for $i, j = 0, 1, \ldots$ and $s \geq 0$. The evolution of a time homogenous CTMC is described by a matrix Q (known as the generator matrix) of the (q_{ij}), where q_{ij} is the infinitesimal rate of moving from state i to state j, $i \neq j$, and $q_{ii} = -\sum_{i \neq j} q_{ij}$.

The Markov property implies that, if at time t the process is in state j, the time remaining in state j is independent of the time already spent in state j. This is known as the *memoryless property*. This means that, if S is the time spent in any state (known as the *sojourn time*), then:

$$\mathbb{P}(S \leq t + s \mid S > t) = \mathbb{P}(S \leq s) \tag{2.27}$$

A consequence of Equation 2.27 is that all sojourn times in a CTMC must be exponentially distributed since this is the only continuous distribution function which satisfies this condition (see for example [14] for a proof). The rate out of state i, and therefore the parameter of the sojourn time distribution, is μ_i and is equal to the sum of all rates out of state i, that is $\mu_i = -q_{ii}$. This means that the density function of the sojourn time in state i is $f_i(t) = \mu_i e^{-\mu_i t}$ and the average sojourn time in state i is μ_i^{-1}.

An important special case of a CTMC is the Poisson process. The Poisson process is a renewal process with exponentially distributed renewal time. The parameter of the exponential distribution λ, is known as the *rate* of the Poisson process. The Poisson process is often used as an approximation of the arrivals into a number of systems such as tasks arriving at a processor input buffer or the number of people joining a post office queue.

Definition 2.7 (The Poisson Process) *The Poisson process is a renewal process with interarrival time having probability distribution function F and density function (pdf) f, given by:*

$$F(x) = P(X \leq x) = \begin{cases} 1 - e^{\lambda x} & for \quad x \leq 0 \\ 0 & otherwise \end{cases}$$

$$f(x) = F'(x) = \begin{cases} \lambda e^{-\lambda x} & for \quad x \leq 0 \\ 0 & otherwise \end{cases}$$

where λ is the rate of the Poisson process.

As the Poisson process has exponentially distributed inter-arrival times, the Poisson process has the memoryless property as defined above.

The definitions in Section 2.3.1 for recurrence and irreducibility in DTMCs also hold for CTMCs. The steady-state distribution for a CTMC is defined similarly as for a DTMC. Once again, we denote the set of steady-state probabilities as π_j.

Definition 2.8 *In a CTMC which has all states recurrent non-null and which is irreducible and time homogenous, the limiting or steady-state distribution π_j is given by:*

$$\pi_j = \lim_{t \to \infty} \mathbb{P}(X(t) = j \mid X(0) = i) \tag{2.28}$$

This leads us to the steady-state theorem for CTMCs:

Theorem 2.3 *For a finite, irreducible and time homogenous CTMC, the steady-state probabilities π_j always exist and are independent of the initial state distribution. They are uniquely given by the solution of the equations:*

$$-q_{jj}\pi_j + \sum_{k \neq j} \pi_k q_{kj} = 0 \tag{2.29}$$

subject to

$$\sum_i \pi_i = 1 \tag{2.30}$$

Writing as a vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$, the above equations can be expressed in matrix form as:

$$\boldsymbol{\pi} Q = 0 \tag{2.31}$$

where Q is the generator matrix of the CTMC.

If the times at which a CTMC X moves from one state to another are ignored, and we number the state transitions, then the resulting sequence of states $\{X_n \mid n = 0, 1, \ldots\}$, is a DTMC. This DTMC is known as the embedded Markov chain (EMC) and describes the behaviour of the CTMC at state-transition instants. The EMC of a CTMC has a one-step transition matrix P where $p_{ij} = \frac{q_{ij}}{-q_{ii}}$ for $i \neq j$ and $p_{ij} = 0$ for $i = j$.

2.4 Queueing Theory

Queueing network modelling is a particular approach to system modelling in which a system is represented as a network of queues. Queueing network models have become important tools in the design and analysis of computer and communications systems and a vast body of related theory, known as *queueing theory* has been developed, see for instance [53, 14, 82, 17, 62].

2.4.1 Queueing Networks

Queueing networks model distributed systems, which consist of entities requiring service from some resource or set of resources. These entities may have to wait to receive service in some sort of order.

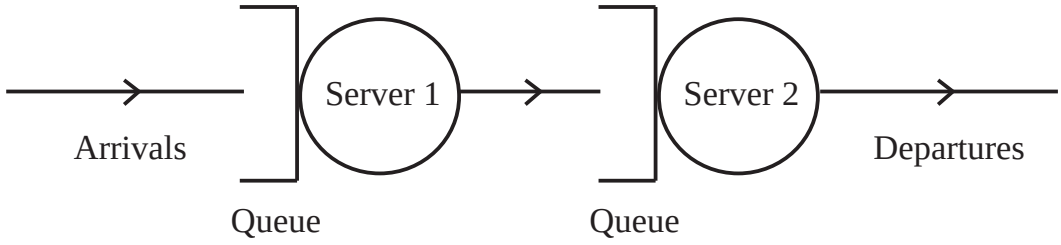


Figure 2.3: Example queueing network.

A queueing network (an example of which is illustrated in Fig. 2.3) is made up of four basic components:

- **Customers** are entities requiring service. They move between queues and receive service at the servers. Customers may be divided into classes, which can affect how they are routed between queues and how they are served (for instance, classes maybe assigned different priority-levels and the arrival of a high-priority customer at a queue may pre-empt the service of a lower priority customer).

- **Queues** store customers while they wait for service at one or more servers. Queues may have a fixed capacity or have the ability to store an infinite number of customers.

- **Servers** are the resources that provide service to customers, who are served according to some *scheduling strategy*. The time taken for a server to serve a customer is a random variable, which may be drawn from a number of distributions. A server may have one or more queues connected to it.

- **Arcs** interconnect servers and queues, indicating the paths that may be taken by customers. An arc without origin leading into a queue indicates that customers arrive from outside the network; conversely for departures.

Additionally, when customers depart from one server and can arrive at one of several destination queues it is necessary to specify routing probabilities. These are the probabilities that a customer leaving that server will be routed to each of the possible destinations. Different classes of customers can have different routing probabilities. Queueing networks can be classed as either *open* or *closed*. When a network has no external arrivals or external departures, the network is closed. Otherwise, it is an open network.

Queues in a queueing network are often described using the Kendall notation:

$$A/B/m/K/Z/Sched$$

which is a shorthand way of describing the *arrival process/service distribution/number of servers/queue capacity/customer population* (in a closed network)/*scheduling strategy* of a queue. If not specified, the scheduling strategy is assumed to be First In First Out (FIFO), also called First Come First Served (FCFS). When describing the arrival

process or service distributions, the following conventions are used: G for general distribution, M for memoryless distribution and D for deterministic distribution. The queue of most interest to us in this thesis is the $M/M/m$ queue in which there is a memoryless (i.e. Poisson) arrivals process, memoryless (i.e. exponentially distributed) service times and m identical parallel servers.

For closed queueing networks with exponential arrival and service distributions, it is possible to generate a CTMC for the network, where a state in the CTMC is described by the number of customers at each queue. This permits the analysis of models for quantitative performance measures through the analysis of the CTMC, for steady-state, transient and passage time quantities.

We now present some queueing theory results which will be used in this thesis.

2.4.2 Little's Law

A widely applied queueing theory result is Little's law [71], which relates the mean queue length with the mean time spent by a customer in a system for an arbitrary queueing system at equilibrium. Under steady-state behaviour, the arrival rate λ does not change with time and is equal to the departure rate. The average number of customers in the system N does not change, and the average time in the system T does not change. Under these circumstances the following theorem holds:

Theorem 2.4 (Little's Law) *The average number L of customers in a queueing system with arbitrary service and arrival distribution, average arrival rate λ and average time in system (including queueing and service time) W, is given by:*

$$L = \lambda W \tag{2.32}$$

This result has been extended to deal with the calculation of higher moments [53]. For a queueing system with Poisson arrivals in a steady-state:

$$L_k^f = \lambda^k W_k \tag{2.33}$$

where W_k is the k^{th}-moment of a task's waiting time and $L_k^f = E[L(L-1)\dots(L-k+1)]$ is the k^{th} factorial moment of the number of customers in the system; both moments are assumed to be finite.

2.4.3 Steady-state Probability Distribution

In this section we consider closed queueing networks where there are no external arrivals or departures. These models are often used to model networks where the resources have limited capacity.

Consider a single class, closed queueing network with K customers and N FIFO servers with exponential service times. After completing service at node i, where $(i = 1, 2, \dots, N)$, a customer moves to node j with probability q_{ij} for $(i, j = 1, 2, \dots, N)$. Since no job ever leaves the network, these routing probabilities satisfy:

$$\sum_{j=1}^{N} q_{ij} = 1 \quad \text{for} \quad i = 1, 2, \dots, N$$

The state of the network at any time is described by the vector $\mathbf{n} = (n_1, n_2, \dots, n_N)$, where n_i is the number of customers at node i. Since the only possible states are such that:

$$\sum_{i=1}^{N} n_i = K$$

the state space is finite, with size given by the binomial coefficient:

$$\begin{pmatrix} K+N-1 \\ N-1 \end{pmatrix}$$

We denote the average number of customers arriving into node i where $i = (1, 2, \dots, N)$ per unit time by λ_i. These customers can only come from other nodes in the network; thus we can write a set of traffic equations that the arrival rates must satisfy:

$$\lambda_i = \sum_{j=1}^{N} \lambda_j q_{ji} \quad \text{for} \quad i = 1, 2, \dots, N \tag{2.34}$$

These equations do not have a unique solution; however, if one of the λ_i is fixed arbi-

trarily the resulting quantities are proportional to the true arrival rates.

The steady-state distribution of the network can be expressed as the product of factors describing the state of each node known as a *product-form solution* [48].

Theorem 2.5 (Gordon-Newell Theorem) *Let $\lambda_1, \lambda_2, \ldots, \lambda_N$ be any solution of Equations 2.34, b_i the average service time at node i and $\rho_i = \lambda_i b_i$. Then the steady-state probability distribution of the network state, denoted $\pi(n_1, \ldots, n_N)$, is given by:*

$$\pi(n_1, \ldots, n_N) = \frac{1}{G} \prod_{i=1}^{N} \beta(n_i) \rho_i^{n_i}$$

where

$$\beta_i(n_i) = \begin{cases} 1 & \text{if node } i \text{ has a single server} \\ 1/n_i! & \text{if } i \text{ is an infinite server node} \end{cases}$$

and G is the normalising constant determined from the condition that the sum of all probabilities is 1:

$$G = \sum_{\mathbf{n} \in \mathbf{\Omega}} \prod_{i=1}^{N} \beta_i(n_i) \rho_i^{n_i}$$

2.4.4 Arrival Theorem

The arrival theorem states that for a closed queueing network in a steady-state, an arrival entering a queue observes the steady-state distribution for the network with one customer removed. This tells us that the arriving customer behaves as a random observer in a network with population reduced by one. This is intuitively appealing since we can think of the removed customer as the arriving customer itself.

Theorem 2.6 (Arrival Theorem) *For a closed queueing network, suppose $\pi(k, \mathbf{n})$ is the steady-state probability that the network is in state $\mathbf{n}$ when the population is $k = \sum_i n_i$. Then, when the network population is K, the probability that an arrival at node i sees the network in state $\mathbf{n}$ denoted by $A_i(\mathbf{n})$, is:*

$$A_i(\mathbf{n}) = \pi(K - 1, \mathbf{n})$$

2.4.5 Mean Value Analysis

The mean value analysis (MVA) algorithm provides a method for determining mean values in a closed queueing network, based on simple applications of Little's law and the arrival theorem. The quantities of interest are the mean queue length at node i where $i = 1, 2, \ldots, N$ denoted by L_i and the average time spent at a node i per visit denoted by W_i. In addition we are also interested in the average number of visits v_i that a customer makes to node i and the throughput T which is the average number of customers departing the network per unit time. In order to calculate this latter quantity in the context of a closed network, we convert the closed network into an equivalent open network that has all the same characteristics of the closed network [53, 82].

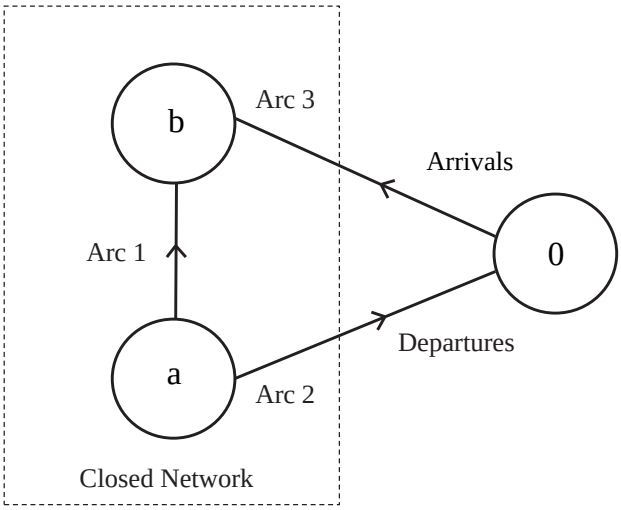


Figure 2.4: Equivalent open network.

In a real system where the number of customers are kept constant, the membership of the set of customers usually changes, with a customer leaving the network once service has been completed immediately replaced by a new customer. We will replicate this behaviour in our model. Assume that we have within a closed network two connected nodes a and b, connected by arc 1. We introduce a new node labelled 0 outside of the closed network and replace arc 1 with arc 2, node 0 and arc 3 such that the source of arc 1 is the source of arc 2 and the destination of arc 1 is the destination of arc 3 as illustrated in Fig. 2.4. Whenever a customer moves along arc 2 and passes through node 0, it immediately departs the network and is replaced by a stochastically identical new customer, which is immediately moved to node b via arc 3. This modification of

the model has no effect other than to transform the customers from permanent entities into temporary ones. The time a customer spends in the network is now the time between two consecutive passes of the same customer through node 0. We can now define the throughput T, as the average rate at which customers pass through node 0 in the steady state, therefore T now corresponds to both the external departure rate and the external arrival rate.

If the average arrival rate to node i is λ_i then, if each customer makes on average v_i visits to node i, we have $\lambda_i = Tv_i$, which implies that the v_i satisfy the traffic equations (Equation 2.34) giving:

$$v_i = \sum_{j=1}^{N} v_j q_{ji} \quad \text{for} \quad i = 1, 2, \ldots, N \tag{2.35}$$

These equations do not have a unique solution; however, if one of the v_i is known, then all others can be obtained. Since all traffic going from node a to node b passes through node 0, we have $v_0 = v_a q_{ab}$. By definition every customer passes through node 0 exactly once (since after passing through node 0, the customer is immediately replaced by a stochastically identical new customer), so using this we can find v_a:

$$v_a = \frac{1}{q_{ab}}$$

Applying Little's law to node i yields:

$$L_i = \lambda_i W_i = T v_i W_i \quad \text{for} \quad i = 1, 2, \ldots, N$$

The sum of the individual queue lengths is exactly K; thus we obtain another relationship between the unknown performance measures T and W_i:

$$\sum_{i=1}^{N} L_i = K = T \sum_{i=1}^{N} v_i W_i \quad \text{or} \quad T = \frac{K}{\sum_{i=1}^{N} v_i W_i}$$

Suppose we let Y_i be the mean number of customers seen by an arrival to node i. The mean waiting time of this customer is the sum of the service times of those customers

and its own service time:

$$W_i = \frac{1}{\mu_i}(Y_i + 1)$$

The arrival theorem (given in Section 2.4.4) tells us that the arriving customer observes a network with population reduced by one. Writing Y_i and L_i as functions of K, this property gives $Y_i(K) = L_i(K-1)$ for $K > 0$.

We can now develop a solution based on recurrence relations for $i = 1, 2, \ldots, N$ and $K > 0$

$$W_i(K) = \frac{1}{\mu_i}[L_i(K-1) + 1] \tag{2.36}$$

$$T(K) = \frac{K}{\sum_{i=1}^{N} v_i W_i(K)} \tag{2.37}$$

$$L_i(K) = T(K) v_i W_i(K) \tag{2.38}$$

with initial conditions $L_i(0) = 0$. We can compute the values of $W_i(K), T(K)$ and $L_i(K)$ by a simple iteration as follows. Starting with the base case $L_i(0) = 0$ we obtain values of $W_i(1), T(1)$ and $L_i(1)$ for $1 \leq i \leq N$. From this we derive the next population level and so on until the iteration reaches the desired population level. On each iteration we compute $2N + 1$ quantities; thus for K iterations we need $O(NK)$ operations to compute the performance measures.

2.4.6 Cobham's Formula

Often in real-life queueing systems, we require priority service to be given to certain customers. Typically, this priority system not only reduces waiting times for the high priority users, but can also improve overall resource utilisations. We consider a non-pre-emptive queueing discipline where a high priority customer must wait for service if there is a customer already in service, even if the arriving customer has a higher priority than the customer in service. We now present *Cobham's formula* [28] for the mean queueing time for each priority class of customer for non-pre-emptive priority queues.

Assume there are R customer classes, numbered $1, 2, \ldots, R$ where each class is assigned a priority level such that class r customers have priority over class s customers if $r < s$. Thus the priority levels are such that class 1 customers have highest priority and class R customers have lowest priority. We consider a single server queueing system with R FIFO queues (one for each customer class). Customers of class r arrive according to a Poisson process with rate λ_r, with some general service distribution with mean rate μ_r for class r customers. After service completion, the next customer chosen for service is the one with the highest priority. Then Cobham's formula states that:

$$\begin{aligned} W_1' &= \frac{W_0}{1-\rho_1} \quad \text{for highest priority customers } (r=1) \quad \text{and,} \\ W_r' &= \frac{W_0}{(1-\sigma_{r-1})(1-\sigma_r)} \quad \text{where} \quad \sigma_r = \sum_{j=1}^{r} \rho_j \quad \text{for} \quad r = 2, \ldots, R \end{aligned}$$

where W_r' is the mean time spent by a class r customer waiting to start service, W_0 the mean time spent waiting for the departure of the customer in service at the time of arrival, $\rho_r = \lambda_r/\mu_r$ is the *load* for class r and σ_r is the total load of priority higher than or equal to class r.

2.5 Laplace Transforms

The Laplace transform is an integral transform that arises in many areas of science and engineering. It is often applied to change a hard-to-solve problem in the real-valued time t-domain into an easier problem in the complex-valued s-domain. For example, they are used to transform differential equations into a simple algebra problem where a solution can be easily obtained, then transformed back to retrieve the solution of the original problem. This is the approach taken for the passage time analysis of Markov chains in the next section.

Definition 2.9 (Laplace Transform) *When it exists, the Laplace transform (denoted by either $L\{f(t)\}$, $L(s)$ or $f^*(s)$) of a real-valued function $f(t)$ is given by:*

$$L(s) = f^*(s) = \int_0^\infty e^{-st} f(t) \mathrm{d}t$$

where s is a complex number.

A sufficient condition for the existence of the Laplace transform of a function $f(t)$, is if $f(t)$ is of exponential order [38]. This requires $|f(t)|$ not to grow too rapidly as t tends to infinity. Specifically we say that the function $f(t)$ is of exponential order if there exists real constants $M > 0$ and K such that:

$$|f(t)| \leq Me^{Kt}$$

holds for all $t \geq 0$. All functions in this thesis are assumed to be of exponential order.

2.5.1 Laplace Transform Properties

The reason that Laplace transforms are widely utilised is because they have a number of useful properties [38]. These include:

Uniqueness If $f(t)$ and $g(t)$ are functions of t and $f^*(s)$ and $g^*(s)$ are their corresponding Laplace transforms then $f^*(s) = g^*(s) \Leftrightarrow f(t) = g(t)$.

Linearity If a and b are constants and $f(t)$ and $g(t)$ are functions of t, then:

$$L\{af(t) + bg(t)\} = aL\{f(t)\} + b\{g(t)\}$$

Convolution This Laplace transform property is particularly useful in passage time analysis. The calculation of the probability density function of a passage time between two states is achieved by convolving the probability density functions of the sojourn times of the states along all the paths between the source and target states. The convolution of two functions $f(t)$ and $g(t)$ denoted $f(t) * g(t)$ is given by:

$$f(t) * g(t) = \int_0^t f(\alpha)g(t-\alpha)\mathrm{d}\alpha$$

The convolution of n functions requires the evaluation of an $(n-1)$ dimensional integral. To perform such a calculation for large values of n (perhaps in the millions) would be

impractical. Instead, we exploit the convolution property of Laplace transforms, which states that the Laplace transform of the convolution of two functions is the product of the functions' individual Laplace transforms.

Theorem 2.7 (Convolution Theorem) *The Laplace transform of the convolution of two functions $f(t)$ and $g(t)$, denoted by $L\{f(t) * g(t)\}$, is the product of the Laplace transforms of the two functions, that is:*

$$L\{f(t) * g(t)\} = f^*(s)g^*(s)$$

Thus the convolution of the functions $f(t)$ and $g(t)$ can be obtained by inverting the Laplace transform of the convolution. The next section outlines the inversion of Laplace transforms using numerical methods.

Integration The final property of Laplace transforms which is particularly useful in the context of passage time analysis is that dividing the Laplace transform of a function by s corresponds to the integration of $f(t)$ in the t-domain. Thus if $f(t)$ is a probability density function and $F(t)$ is the corresponding cumulative distribution $\int_0^\infty f(t)dt = F(t)$, then the Laplace transform of $F(t)$ can be calculated from the Laplace transform of $f(t)$ by dividing $L\{f(t)\}$ by s:

$$L\{F(t)\} = \frac{L\{f(t)\}}{s}$$

2.5.2 Laplace Transform Inversion

The inverse of the Laplace transform $f^*(s)$ of a function $f(t)$ denoted by $L^{-1}\{f^*(s)\}$ is the function $f(t)$ itself. This is also known as the Bromwich integral, and is a complex integral given by:

$$L^{-1}\{f^*s\} = f(t) = \frac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty} e^{st}f^*(s)ds$$

where α is a real number lying to the right of the real part of all the singularities of $f^*(s)$.

The methods presented in this thesis utilise the *numerical* inversion of Laplace transforms, for which there are numerous algorithms available; however, in this thesis, the methods presented use the Laguerre method [1, 52] (also referred to as Weeks' method) as the default method.

The Laguerre Method

The Laguerre method represents a function $f(t)$ in terms of its Laguerre series representation [38]:

$$f(t) = \sum_{n=0}^{\infty} q_n l_n(t) \quad t \geq 0$$

where the l_n are the Laguerre polynomials given by:

$$l_n(t) = \left(\frac{2n-1-t}{n}\right) l_{n-1}(t) - \left(\frac{n-1}{n}\right) l_{n-2}(t)$$

starting with $l_0(t) = e^{-t/2}$ and $l_1(t) = (1-t)l_0(t)$. The q_n are the Laguerre coefficients, given by:

$$q_n = \frac{1}{2\pi r^n} \int_0^{2\pi} Q(re^{iu}) e^{-inu} du$$

where $r = (0.1)^{4/n}$ and $Q(z) = (1-z)^{-1} f^*((1-z)/2(1-z))$.

This integral can be approximated using the trapezoidal rule with p trapezoids so that

$$q_n \approx \frac{1}{2nr^n} \left(Q(r) + (-1)^n Q(-r) + 2 \sum_{j=1}^{n-1} (-1)^j Re(Q(re^{\pi ji/n})) \right) \tag{2.39}$$

As described in [52], the Laguerre method can be modified by noting that the Laguerre coefficients q_n are independent of t. The $|l_n(t)| \leq 1$ for all n and t, and $l_n(t)$ approaches 0 as $n \to \infty$. However, the latter rate of convergence is very slow, so the convergence of the Laguerre series effectively depends on the decay rate of q_n as $n \to \infty$. If f is continuous and has continuous derivatives, then the convergence of the Laguerre coefficients is rapid. Slow convergence of the q_n coefficients can often be addressed by

exponential damping and scaling using two real parameters σ and b [99]. The idea is to apply the Laguerre inversion algorithm to the function:

$$f_{\sigma,b}(t) = e^{-\sigma t} f(t/b)$$

Then $f(t)$ can be recovered as:

$$f(t) = e^{\sigma b t} f_{\sigma,b}(bt)$$

The corresponding Laguerre generating function for $f_{\sigma,b}$ is:

$$Q_{\sigma,b}(z) = \frac{b}{1-z} f^* \left(\frac{b(1+z)}{2(1-z)} + b\sigma \right)$$

Each q_n coefficient is computed as in Equation 2.39, using the trapezoidal rule with $2n$ trapezoids. However, if we apply scaling to ensure that q_n has decayed to (almost) zero by term p_0 (say $p_0 = 200$), we can instead make use of a constant number of $2p_0$ trapezoids when calculating each q_n [52]. This allows us to calculate each q_n with the same or higher accuracy as in [1] while simultaneously providing the opportunity to cache and re-use values of $Q(z)$. Since q_n does not depend on t, and each evaluation of $Q(z)$ involves a single evaluation of $f^*(s)$, we can therefore obtain the $f(t)$ at an arbitrary number of t-values at the fixed cost of evaluating $Q(z)$ (and hence $f^*(s)$) just $2p_0$ times.

2.6 Response Time Analysis

This section describes methods by which response time densities may be extracted from Markov processes, by analytically inverting the Laplace transform $L(s)$ of the required response time density $f(t)$. From the Laplace transform we can recover the value of $f(t)$ at any t by using one of several algorithms for numerical transform inversion. Examples of well-known numerical inversion algorithms include the Euler, Post-Widder, Gaver and Laguerre methods [3, 4, 1, 2]. These algorithms compute $f(t)$ at a given t by evaluating $L(s)$ at several values of s. We have employed the Laguerre method [52]

which was discussed in detail in the previous section. For an in-depth discussion on obtaining response time densities from Markov and semi-Markov processes see [38, 52, 20, 39].

2.6.1 Passage Time Distributions in Markov Chains

This subsection describes a technique to numerically evaluate passage time distributions in continuous-time Markov chains [52].

First passage time equations

Consider a finite irreducible, continuous-time Markov Chain with n states $\{1, 2, \ldots, n\}$ and generator matrix Q. If $X(t)$ denotes the state of the CTMC at time t $(t \geq 0)$, then the first passage time from a source state i into a non-empty set of target states $\vec{j}$ is:

$$T_{i\vec{j}}(t) \quad = \quad \inf\left\{u > 0 : X(t+u) \in \vec{j} \mid X(t) = i\right\} \quad (\forall t \geq 0)$$

For a stationary time-homogeneous CTMC, $T_{i\vec{j}}(t)$ is independent of t, so:

$$T_{i\vec{j}} \quad = \quad \inf\left\{u > 0 : X(u) \in \vec{j} \mid X(0) = i\right\}$$

$T_{i\vec{j}}$ is a random variable with an associated probability density function $f_{i\vec{j}}(t)$ such that

$$\mathbb{P}(a < T_{i\vec{j}} < b) \quad = \quad \int_a^b f_{i\vec{j}}(t)dt \quad (0 \leq a < b)$$

Our aim is to determine $f_{i\vec{j}}(t)$. In effect, this involves convolving state holding times over all possible paths (including cycles) from state i into any of the states in the set $\vec{j}$. By shifting the problem into the Laplace domain we can exploit the basic transform property that the transform of a convolution of two functions is the product of the transforms of those functions [2]. Another important advantage of working with Laplace transforms is that we can derive arbitrary moments of $f_{i\vec{j}}(t)$ by evaluating derivatives of its Laplace transform $L_{i\vec{j}}(s)$ at $s = 0$. In general, the value of $L_{i\vec{j}}(s)$ can

be computed by solving a set of n linear equations that are derived using a first step analysis:

$$\begin{aligned} L_{i\vec{j}}(s) &= \int_0^\infty e^{-st} f_{i\vec{j}}(t)dt \\ &= E\left[e^{-sT_{i\vec{j}}}\right] \\ &= \sum_{k\notin\vec{j}} -\frac{q_{ik}}{q_{ii}} E\left[e^{-s(S_i+T_{k\vec{j}})}\right] + \sum_{k\in\vec{j}} -\frac{q_{ik}}{q_{ii}} E\left[e^{-s(S_i)}\right] \\ &= \sum_{k\notin\vec{j}} \frac{q_{ik}}{(s-q_{ii})} L_{k\vec{j}}(s) \qquad + \sum_{k\in\vec{j}} \frac{q_{ik}}{(s-q_{ii})} \end{aligned}$$

i.e

$$(s-q_{ii})L_{i\vec{j}}(s) = \sum_{k\notin\vec{j}} q_{ik}L_{k\vec{j}}(s) + \sum_{k\in\vec{j}} q_{ik} \tag{2.40}$$

where $S_i \sim Exp(-q_{ii})$ is the sojourn time in state i $(1 \leq i \leq n)$. Expressing this system of n linear equations in standard matrix-vector form $(Ax = b)$ yields:

$$\begin{pmatrix} s-q_{11} & -q_{12} & \cdots & -q_{1n} \\ 0 & s-q_{22} & \cdots & -q_{2n} \\ 0 & -q_{32} & \cdots & -q_{3n} \\ 0 & \vdots & \ddots & \vdots \\ 0 & -q_{n2} & \cdots & s-q_{nn} \end{pmatrix} \begin{pmatrix} L_{1\vec{j}}(s) \\ L_{2j}(s) \\ L_{3\vec{j}}(s) \\ \vdots \\ L_{n\vec{j}}(s) \end{pmatrix} = \begin{pmatrix} 0 \\ q_{21} \\ q_{31} \\ \vdots \\ q_{n1} \end{pmatrix} \tag{2.41}$$

where $\vec{j} = \{1\}$ in this case.

The problem can also be readily extended to multiple initial states. In particular, if the probability distribution of the initial states is known – typically the steady-state distribution – the problem reduces to that of weighting the first passage time densities for each initial state.

Moments

The n^{th} moment of the first passage time between a given source state i and set of target states $\vec{j}$ is:

$$M_{i\vec{j}}(n) = (-1)^n \left.\frac{d^n L_{i\vec{j}}(s)}{ds^n}\right|_{s=0}$$

This can be found by differentiating Equation 2.40 n times at $s = 0$ and solving a similar set of equations, for $n \geq 0$:

$$-q_{ii}M_{i\vec{j}}(n) = \sum_{k \notin \vec{j}} q_{ik}M_{k\vec{j}}(n) + nM_{i\vec{j}}(n-1) \tag{2.42}$$

for $i \notin \vec{j}$ and $M_{i\vec{j}}(n) = 0$ for $i \in \vec{j}$. For $n = 0$, we have $M_{i\vec{j}}(0) = 1$ and so each set of moments can be computed iteratively.

2.6.2 Passage Time Analysis Pipeline

A complete passage time analysis pipeline, implementing the theory discussed has been implemented as shown in Fig 2.5 [38]. Models are specified in an enhanced form of the DNAmaca Markov chain analyser interface language [63, 64], which supports the specification of queueing networks, stochastic Petri nets, stochastic process algebras and other high-level formalisms that can be mapped onto Markov and semi-Markov chains.

From the input model, DNAmaca's state generator produces the generator matrix Q of the model's underlying Markov chain, as well as a list of the initial states (with their corresponding weighting) and the target states. The matrix Q is then put through a *hypergraph partitioner* [96] to form partitioned matrix files that incur low communication overhead when performing parallel sparse matrix vector multiplications. Control now passes to the distributed Laplace transform inverter which implements the master-slave structure shown. Both the Laguerre and Euler inversion algorithms are supported. Initially the master runs through the Laplace transform inversion algorithm (the Laguerre method is used as the default) and notes the distinct values of s at which $L(s)$ needs to be evaluated. Those values of s for which there is no corresponding $L(s)$ value stored in a disk cache are added to the global work queue.

At start up, the slave processor groups read in the partitioned matrix files, with each slave within a group reading in a different file. Each slave group then applies for an s-value from the global queue. These groups then return computed values of $L(s)$ to the master, which stores the value in memory and disk caches, before issuing more

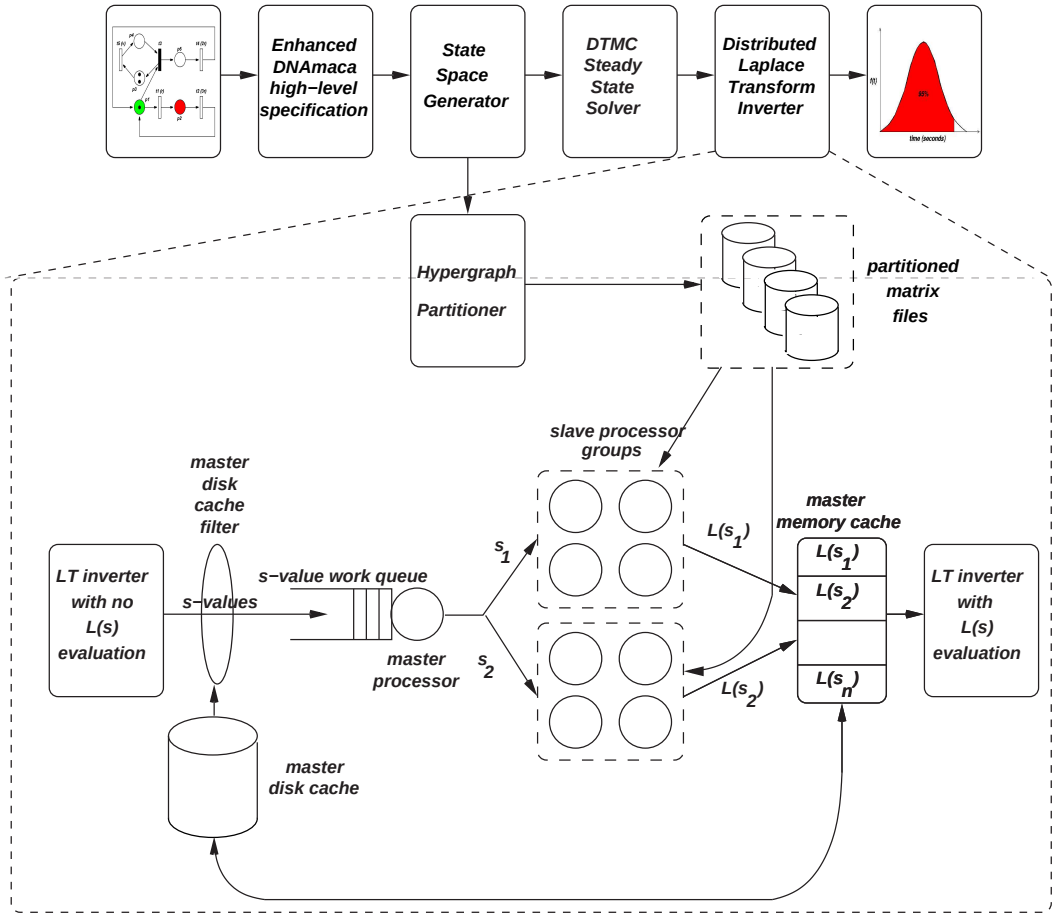


Figure 2.5: Passage time analysis pipeline.

work (if any). When all values of $L(s)$ have been computed, the master runs through the Laplace inversion algorithm once more, this time performing all calculations and obtaining the required $L(s)$ values from the memory cache. Resulting points on the response time density are written to a disk file and displayed using GNUplot.

2.7 Modelling in Healthcare

In this section we discuss existing work in the area of modelling and simulation within healthcare. In particular we concentrate on patient arrivals modelling and the modelling and simulation of patient flow in healthcare systems.

2.7.1 Patient Arrivals Modelling

In the literature, the modelling of patient arrivals to emergency services is undertaken to achieve two objectives: firstly, to characterise the nature of arrivals and to forecast future numbers. Secondly, to parameterise patient flow models and simulations of healthcare systems.

Generally, when arrivals modelling is used to facilitate the objective of parameterising a patient flow model or simulation, a straightforward approach is taken, whereby either a Poisson arrivals process is assumed or historical attendance is replicated [30, 22, 27, 75, 93, 80, 41, 29].

When patient arrivals modelling is used to forecast future arrivals in order to plan ahead for staffing and resource needs two approaches have been taken. Firstly there are papers which look to link patient arrivals to external factors such as the weather and calendar events such as the day of the week and bank holidays. Other studies have utilised various time series models to characterise and forecast future emergency services arrivals.

The relationship between calendar and weather data and the arrival of patients in an emergency department is explored in [37, 26, 60]. The studies [37, 60] investigate patterns in the arrival of patients to walk-in clinics in San Antonio, Texas and Lexington, Kentucky respectively. The main influences on the daily arrival rate of patients are calendar variables, like the year, month, day of the week, bank holidays and in [60], the days when pension cheques get delivered. The main weather-related contribution was the maximum daily temperature; however, [37] concluded that overall weather related components added little to the accuracy of the predictions of the models.

A paediatric emergency department in Chicago, Illinois is modelled in [26]. This study is based on three months of data in 1975-6. The distinction is made between different diagnoses and this study finds that for instance flu and minor injuries are seasonal. Extremely cold weather conditions are shown to reduce the number of visits. Extreme heat on the other hand shows no significant change in the number of visits. The paper concludes that the influence of the weather is small and the perceived large fluctuations in patient influx during extreme weather is related to the arrivals being more bursty.

It is known that for some illnesses, in particular chronic obstructive pulmonary disease (COPD), the severity of symptoms can vary throughout the year because of changes in the weather or the presence of infections. The Met Office has studied the relationship between the weather and certain illnesses including COPD and uses its findings – with other factors such as virus levels and air pollution – to produce health forecasts for healthcare providers, enabling them to deliver anticipatory care and reduce the number of hospital admissions [76, 77, 78, 79].

A number of studies [23, 59, 95] have utilised time series models to forecast acute arrivals to hospitals. In [95] two years of data (from 1989 to 1990) were used to fit and validate hourly arrivals at an American Emergency Department, using a number of different time series models including ARIMA and moving average models. They found that simple (moving average based) models performed better than more complex (ARIMA) models.

An analysis of emergency admissions and bed occupancy in an UK hospital is presented in [59]. Data spanning from April 1993 to March 1999 was used to fit SARIMA and GARCH models. This paper found a linear relationship between the seasonally adjusted number of occupied beds and the mean day time temperature, but long term seasonality proved more important than short-term weather effects in general.

Recently in [23] both daily and hourly ambulance arrivals at a Canadian Emergency Department were modelled using a number of auto-regressive and ARIMA models. Using four years of data (from 2000 to 2004), the models were fitted to the first three years of data and validated against the remaining data. This study found that these models were useful for short term forecasts and that there were significant links with public holidays and demand.

2.7.2 Healthcare Systems Modelling

The idea of modelling health service departments is, of course, by no means new. Several studies have been made of patient flow in hospitals in general [33, 35, 100] and Accident and Emergency unit – also known as Emergency Departments (EDs) – in particular [16, 27, 68, 73, 75, 30, 74, 93].

The studies of A&E can be categorized into either analytical or simulation models. Basic queueing theory was applied in [73, 74], where A&E departments were modelled as simple queueing networks consisting of at most 2 to 3 stages. In the case of [74] these models gave good fits to observed patient service times in May to July 2002. Although such analysis is useful in providing performance measures of the system as a whole, details and insights at the resource level cannot be obtained.

Discrete-event simulation software packages have been used to create simplified models of patient flow in Emergency Departments in order to assess the impact of different staff schedule changes [27, 93], patient treatment pathways [30, 75] and the number of beds available [68]. All of these studies suffered from a lack of data, with both [27, 68] only using estimates to parameterise their models and [30, 93, 75] only having access to five days, a week and three months of observed data on which to base their models respectively. Consequently, [68] was unvalidated against real data, while [27, 30] did not result in good agreement with actual patient response times. Similarly, [16] utilised discrete-event simulation (written in SIMIAN and FORTRAN) to model a paediatric Emergency Room, based on three days of observed data. This model was then used to assess the impact of various patient treatment schemes. This study found reasonable fit to various patient waiting time quantities, but was less accurate with the length-of-stay predictions.

Although discrete-event simulation software packages give users an easy to use platform with which to build simulations, the drawback of these packages is the lack of flexibility when tailoring the simulation to more complex scenarios e.g. when assessing the impact of patient priority schemes. However, although many of the studies that use such packages provide poor correlation with the actual results, the main objective of many of these studies is to obtain insight into the impact of various resource and procedural scenarios, without having to disrupt the actual running of an A&E department.

As we have seen, a drawback to many existing studies is that they are frequently parameterised using very little data and either remain unvalidated or are validated against small quantities of real waiting time data, since collecting this data was until recently a time-consuming, expensive, manual operation.

2.8 Case Study Department

Our case study department is a large Accident and Emergency department of a London hospital which serves a large area of North London.

In recent years, the department has seen an increasing number of attendances. Table 2.1 shows the number of attendances to the department during each hospital financial year (1 April to 31 March) for the years 2002 to 2007.

year	no. of attendances
2002/2003	79 029
2003/2004	89 130
2004/2005	97 759
2005/2006	102 191
2006/2007	102 418

Table 2.1: Number of total attendances into our case study department by year.

Table 2.1 shows that there has been a steep increase in attendance levels with a 29.6% increase since 2002/2003, with the latest attendance figures equating to an average of 281 patients coming into the department every day.

We have applied for and obtained research project status and ethical approval to access pseudonymised patient timing data (that is non-patient-identifiable-data which has been tagged by a unique reference number, in order to track the patient through stages of treatment) at our case study department for the past five years. This involved a lengthy approval process which included visits to our case study department, writing a project proposal, submitting the ethical approval forms and presenting our project proposal at meetings with the ethical approval committee. Ethical approval for access to pseudonymised patient records was granted by the Harrow Local Research Ethics Committee (Ref. 04/Q0405/72).

Having obtained ethical approval to access this data, we created and placed this data on our own password protected, restricted-access database. This data was then cleaned (i.e. we removed patient records with obvious data entry errors or inconsistencies such as arrival dates in the future or discharge times preceding arrival) and reformatted for our purposes.

Chapter 3

Patient Arrivals Modelling

3.1 Introduction

Accident and Emergency (A&E) departments are for many patients the first point of contact with an NHS hospital and as a result they tend to have a much larger patient throughput than other hospital departments. With attendances increasing year on year and a national government target whereby 98% of patients must spend 4 hours or less from arrival to admission, transfer or discharge, A&E departments are being placed under increasing pressure to process a large number of patients safely and quickly. It is therefore important to understand and characterise the nature of patient arrivals to plan ahead for staffing and resource needs.

For current and future A&E simulations and models to be effective in providing insights into departmental improvements, they need to be parameterised with a realistic workload. There are many publications describing simulations of A&E departments – also known as Emergency Rooms (ERs) or Emergency Departments (EDs) in other countries – in which either a Poisson arrivals process is assumed or historical attendance is replicated [30, 22, 27, 75, 93, 80, 41, 29]. However, these give a much simplified view of A&E arrivals since in the former case it is known that demand for emergency care follows seasonal patterns at many time scales with attendance varying by month of the year, day of the week and even hour of the day; in the latter case, long term trends are not accounted for.

Previous work in this area (as described in Section 2.7.1), has attempted to characterise and forecast acute arrivals to hospitals using a number of different time series models including auto-regressive, moving average and ARIMA models. We now present what is, to the best of our knowledge, the first research to utilise power spectral density analysis and structural time series models in the context of A&E arrivals modelling. This is also the first research to use separate time series models to characterise and forecast walk-in and ambulance arrivals as opposed to either modelling total arrivals [95] or only ambulance arrivals [23].

The models and forecasts of daily A&E arrivals presented in this chapter are based on five years of pseudonymised patient arrivals data supplied by our case study A&E department. Arrivals to the department are aggregated by day and then allocated to one of two arrival streams (walk-in or ambulance) by mode of arrival. Using the first four years of patient arrivals data as a "training" set we analyse the corresponding power spectrum and fit a number of different time series models to each arrival stream. We then test the predictive ability of these models against the remaining one year of "unseen" data.

Next, we present the pattern of patient arrivals by hour throughout the day for each arrival stream. This indicates the busiest times during the day for each arrival type and will be of use to hospital managers when determining optimal staff shift patterns and staff and resources levels. The impact of weather factors such as temperature and rainfall on patient arrival numbers is also briefly explored and discussed.

The remainder of this chapter is arranged as follows. Section 3.2 describes the preliminary data analysis we use to determine the characteristics of each arrival stream and hence the appropriate models with which to fit to the data. The subsequent sections present the different time series models we use to characterise and forecast daily arrivals: Section 3.3 describes a rolling six week average model, Section 3.4 presents an auto-regressive model, and Section 3.5 describes a structural time series model. In each case, forecasts for each of these time series models are presented and compared with observed arrivals. Section 3.6 describes the use of non-homogeneous Poisson processes to further characterise ambulance arrivals. In Section 3.7 we present the hourly breakdown of each arrival stream into our case study department. In Section 3.8 we de-

scribe our investigations into the impact of weather-related factors on patient arrivals. Section 3.9 concludes.

3.2 Preliminary Data Analysis

We study all patient arrivals to our A&E department from 1 April 2002 to 31 March 2007. First we aggregate this data set to obtain a time series of patients arriving per day. We then classify patients as either ambulance arrivals (where electronic patient records indicate that the patient arrived via an ambulance) or walk-in arrivals (all other modes of patient arrival). The two time series of ambulance and walk-in arrivals are then further split into "training" data consisting of the first four years (1456 days) of arrivals which is used to fit our time series models, and "unseen" data consisting of the remaining 370 days of arrivals which is used to determine the accuracy of our model forecasts. In the period 1 April 2002 to 31 March 2007 there were 471 931 total patient arrivals to our case study A&E department. Of these arrivals 129 241 (27.4%) are classified as ambulance arrivals and 342 690 (72.6%) as walk-in arrivals. The "unseen" data – with which we will compare the model forecasts – consists of the 27 430 ambulance and 75 315 walk-in arrivals observed during the period 1 April 2006 to 31 March 2007. All the time series models in this thesis were created and fitted using the *R* statistical software package [90, 84]; the corresponding *R* code can be found in Appendix A.

Plots of the "training" data of daily walk-in and ambulance arrivals are shown in Fig. 3.1. We apply a power spectral density analysis – which describes how the power (strength) of a time series is distributed by frequency – to these time series to determine the strength of any periodicities present. As shown in Fig. 3.2, walk-in arrivals show distinct peaks in the power spectrum corresponding to weekly (seven day) periodic behaviour in the data. The initial large peak at low frequency indicates an annual periodicity and the lower peaks (at frequencies 2.0 and 3.0 weeks^{-1}) correspond to the harmonics of the main seven day frequency. Ambulance arrivals exhibit a distinct but much weaker seven day periodicity and also an annual periodicity. To further understand the nature of the weekly seasonality shown in both arrival streams, Fig. 3.3

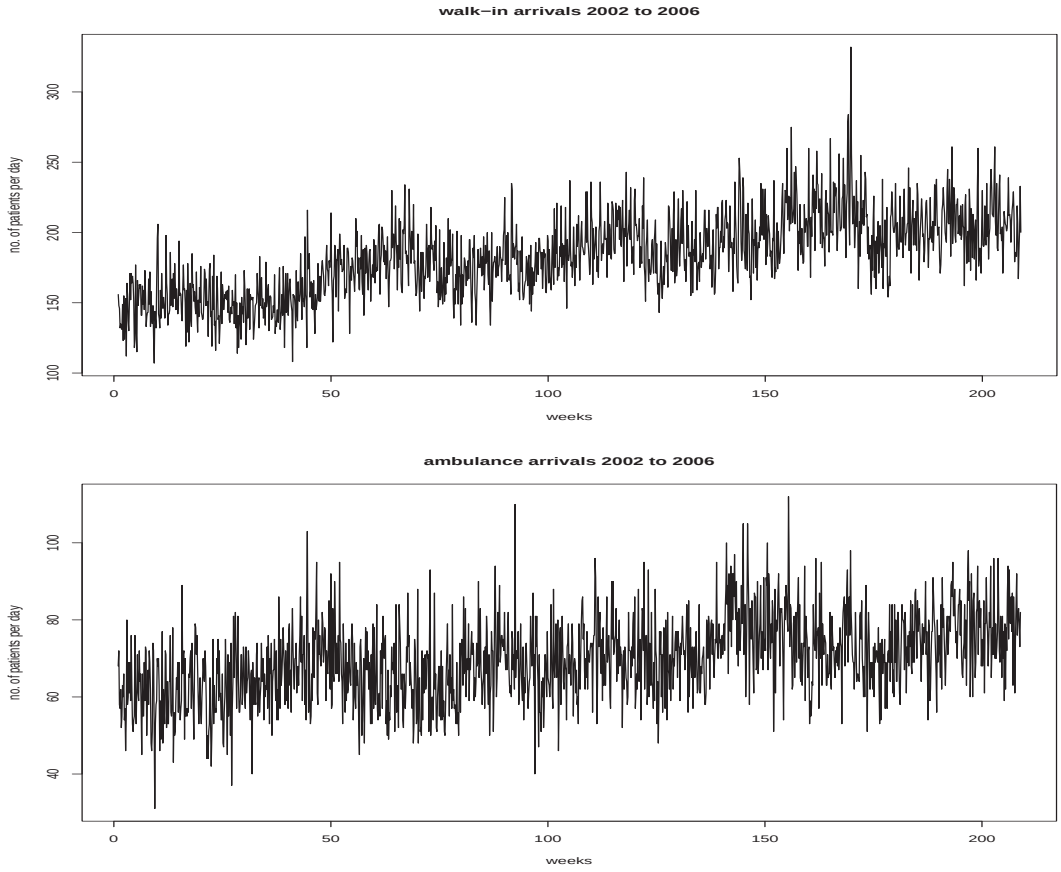


Figure 3.1: Daily walk-in (top) and ambulance (bottom) arrivals for 2002-2006.

shows histograms of the percentage of walk-in and ambulance arrivals by day of week for the "training" data. From these histograms we can see that there are more arrivals on a Monday than for any other day of the week which accounts for most of the weekly seasonality seen in the power spectra. This "Monday effect" is more pronounced for the walk-in arrivals than the ambulance arrivals; hence the much higher weekly peak observed in the power spectrum of walk-in arrivals. The different characteristics exhibited by the power spectra suggest that we need to use separate time series models for each arrival type.

We fitted three types of time series models: rolling average (RA) models, auto-regressive (AR) models and structural time series (ST) models. A six week rolling average model is what is currently used by our case study department to predict the total number of arrivals and so will make a good starting point for comparison with the other time series models. We fit AR models since these are traditionally used for modelling time series

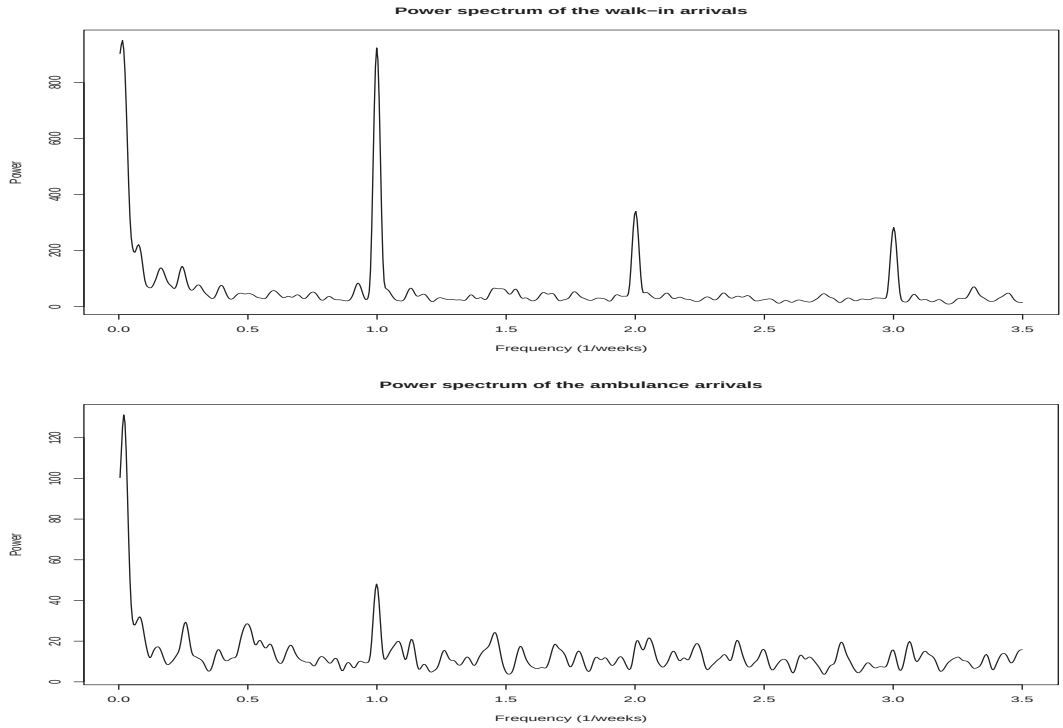


Figure 3.2: Power spectra of the "training" data of walk-in (top) and ambulance (bottom) arrivals.

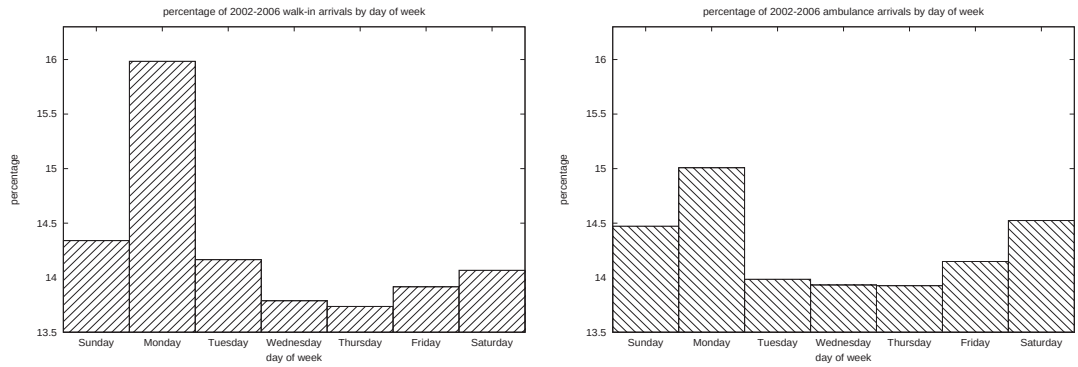


Figure 3.3: The percentage of walk-in (left) and ambulance (right) arrivals by day of week for 2002-2006.

which exhibit regularity. Finally, we use ST models as they – unlike AR models – do not require pre-processing of the data to satisfy stationarity assumptions (cf. Section 2.2.3) and allow us to explicitly incorporate seasonal factors and local linear trends.

To the assess the initial fit of the models, we calculate the Pearson product-moment correlation coefficient (r) of the initial model fit to the "training" data. We also conduct an in-depth analysis of the resulting residuals (cf. Section 2.2.5). First we investigate

the distribution of the residuals; ideally they should be normally distributed with mean zero. The residuals are also checked for independence using the Ljung-Box test, and by examination of their autocorrelation function (acf). Where appropriate we also assess this independence visually using a scatterplot of the residuals against the fitted values.

Using each model fit we then forecast arrivals for the $l = 1^{\text{st}}, 2^{\text{nd}}, \ldots, 7^{\text{th}}$ day ahead. In order to compare the predictions arising from the different types of models, we calculate a number of quality of forecast metrics (cf. Section 2.2.6) including the mean bias of the set of predictions, the root mean square error (RMSE), and the Pearson product-moment correlation coefficient (r) between each set of the l day(s) ahead forecasts and the corresponding observed "unseen" arrivals. When forecasting ahead, we calculate the 95% confidence interval for each prediction and at each forecast horizon we compute the mean width of the 95% confidence intervals for the corresponding set of predictions. We also calculate the fraction p of observed arrivals which lie outside the 95% confidence intervals of the model predictions. Finally, we construct scatterplots to show visually the quality of forecasts from our models.

3.3 Rolling Average Models

3.3.1 Specification

Currently our case study hospital utilises a six week rolling average model to predict the total number of arrivals to the department for the week ahead. In this type of model the predicted value for a given day is the average of the number of arrivals on the corresponding day of the week from the previous n weeks. For instance the predicted number of arrivals into the department on a Monday will be the average of the number of arrivals from the previous n Mondays. For a time series of previous patient arrivals x_t of length T, this can be formalised as follows:

$$\hat{x}_t = \frac{x_{t-7} + x_{t-14} + \ldots + x_{t-7n}}{n} \quad t = 7n+1, \ldots, T \tag{3.1}$$

where $\hat{x}_t$ is the predicted number of arrivals at time t and n is the number of weeks

used to obtain an average ($n = 6$ in our model).

Notice that predictions made by this model do not depend on the values for the previous six days. This means that the same six weeks of data will be used to predict one day ahead as for seven days ahead. For this reason we predict one entire week ahead with the RA models as opposed to performing separate lth day ahead forecasts.

To make predictions with these models, we first fit the rolling average models to the "training" data (the first 1456 days in the time series). From this model fit we calculate the residuals, and assuming that the residuals are normally distributed (which we will verify later), we take 1.96 times the standard deviation of these residuals to be the 95% confidence interval width for our predictions. Failing the normality test, confidence intervals may still be calculated using the residuals via Chebyshev's inequality which states that in any data sample drawn from any probability distribution with finite variance, no more than $1/k^2$ of the values are more than k standard deviations away from the mean (so $k = 4.47$ for a 95% confidence interval). We then forecast arrivals one week ahead at a time. Using the last six weeks (42 data points) of our "training" data, we predict the number of arrivals on each successive day for next seven days ahead; shifting ahead into the "unseen" data by seven days, we use the previous 42 data points to predict the arrivals for the next seven days ahead. This is repeated until we have shifted through the remaining 364 days of data to get 371 predictions in total. This set of one week ahead predictions is truncated to get 370 predictions, which are then compared to the corresponding actual number of walk-in and ambulance arrivals in the "unseen" data. The R code used to create and fit these RA models and to subsequently perform forecasts is shown in Appendix A.1.

3.3.2 Rolling Average Model Fit

To assess the initial fit of the RA models we calculate the Pearson product-moment correlation coefficient (r) of the "training" data with the corresponding model fit. For the walk-in RA model we get $r = 0.7551$, which indicates a good fit. For the ambulance RA model we have $r = 0.4124$, which indicates a relatively poor initial fit. Fig. 3.4 shows plots of the RA model fit to the "training" data for both the walk-in and ambulance

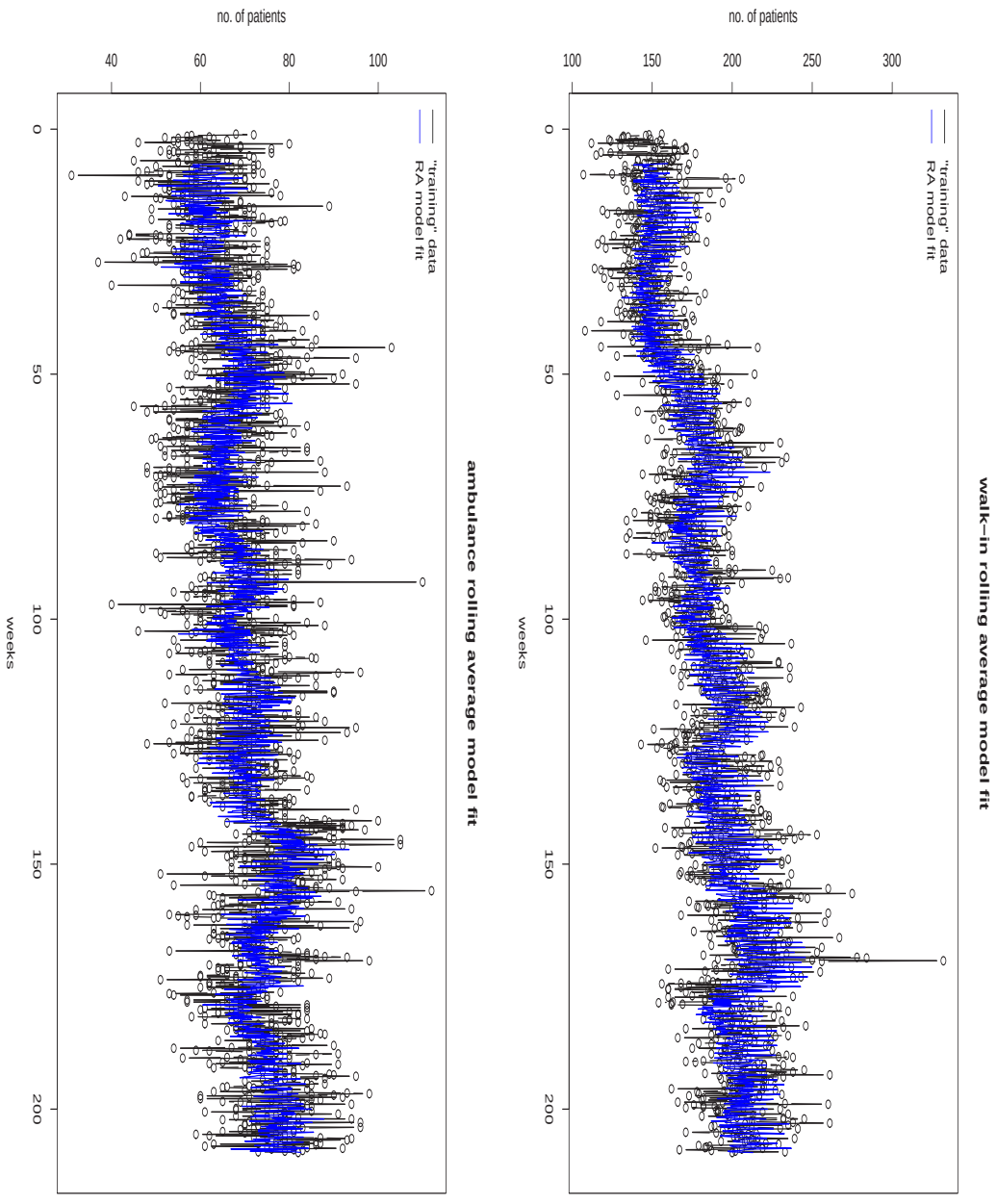


Figure 3.4: The walk-in (top) and ambulance (bottom) arrival RA model fits (in blue) to the "training" data (in black).

arrivals. We also check the residuals ($r_t = \hat{x}_t - x_t$) of the initial model fit. The autocorrelation functions (acfs) of both model residuals are shown in Fig. 3.5; from this we can see that there are many significant peaks in the acfs of both models, but especially for the walk-in model residuals (shown on the left). These correlations within the residuals indicates that there exists remaining structure which has not been incorporated into the RA models. For the walk-in and ambulance model residuals, the Ljung-Box test returned p values $\ll 0.0001$ for both sets of residuals, meaning that the residuals from both model fits are not independent – as was already indicated by the respective acfs. Fig. 3.6 shows the corresponding histogram of the residuals and also a superimposed normal density with the same mean and standard deviation as the residuals; the close correspondence indicates that it is reasonable to assume that the residuals are approximately normally distributed with zero mean and that it is appropriate to take 1.96 times the standard deviation of these residuals to be the 95% confidence interval width for our predictions [24].

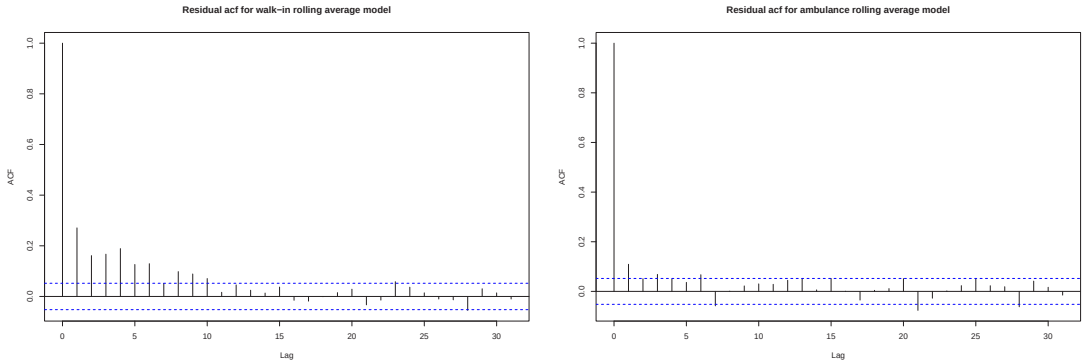


Figure 3.5: The acf of the residuals of the RA models of walk-in (left) and ambulance arrivals (right).

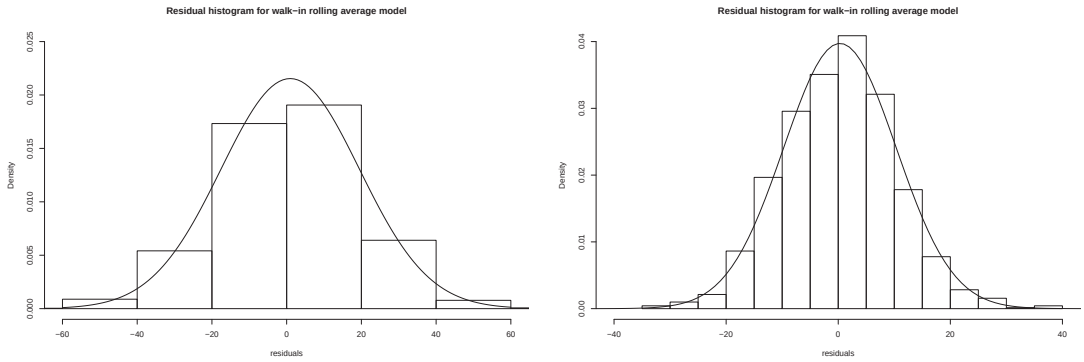


Figure 3.6: The distribution of the residuals of the RA models of walk-in (left) and ambulance (right) arrivals and the corresponding normal distribution.

3.3.3 Rolling Average Model Predictions

Table 3.1 presents the quality of forecast measures for predicting one week ahead at a time using the RA models of walk-in and ambulance arrivals. Plots of the RA model predictions (in blue) for both walk-in and ambulance arrivals with the "unseen" observed patient arrivals (in black) and the corresponding 95% confidence intervals (in red) are shown in Fig. 3.7. The week ahead prediction scatterplots are shown in Fig. 3.8.

RA walk-in model				
Mean Bias	RMSE	r	95% CI width	p
-0.3851	20.0468	0.4965	$\pm$ 36.3052	0.0703
RA ambulance model				
0.3342	9.7517	0.1608	$\pm$ 19.6811	0.0270

Table 3.1: Quality of forecast measures of the one week ahead RA model predictions for walk-in and ambulance arrivals.

Table 3.1 shows that the RA model week ahead predictions show a small negative bias for walk-in arrivals and a small positive bias for the ambulance arrivals. The p values for both the walk-in ($p = 0.0703$) and ambulance ($p = 0.0270$) arrival predictions indicate that 7% of the "unseen" observed patient arrivals lie outside the 95% confidence intervals of our walk-in arrival predictions and 2.7% lie outside for the ambulance arrival predictions. This indicates that the walk-in prediction confidence interval may be slightly too narrow, with the ambulance predictions confidence intervals slightly too wide. Table 3.1 and the scatterplot on the left in Fig. 3.8 show that the RA model of walk-in arrivals performs reasonably well, with our week ahead predictions showing reasonable correlation with the observed "unseen" data ($r = 0.4965$). From Table 3.1 we can see that the quality of the RA ambulance model predictions are worse, with our week ahead predictions showing poor correlation ($r = 0.1608$) with the "unseen" data. This is reinforced by the one week ahead scatterplot shown on the right in Fig. 3.8.

The low correlation between the RA models and the observed data, together with the autocorrelation in the residuals of the model fit and the Ljung-Box test result, suggests that the RA model predictions currently used in our case study department can be improved upon.

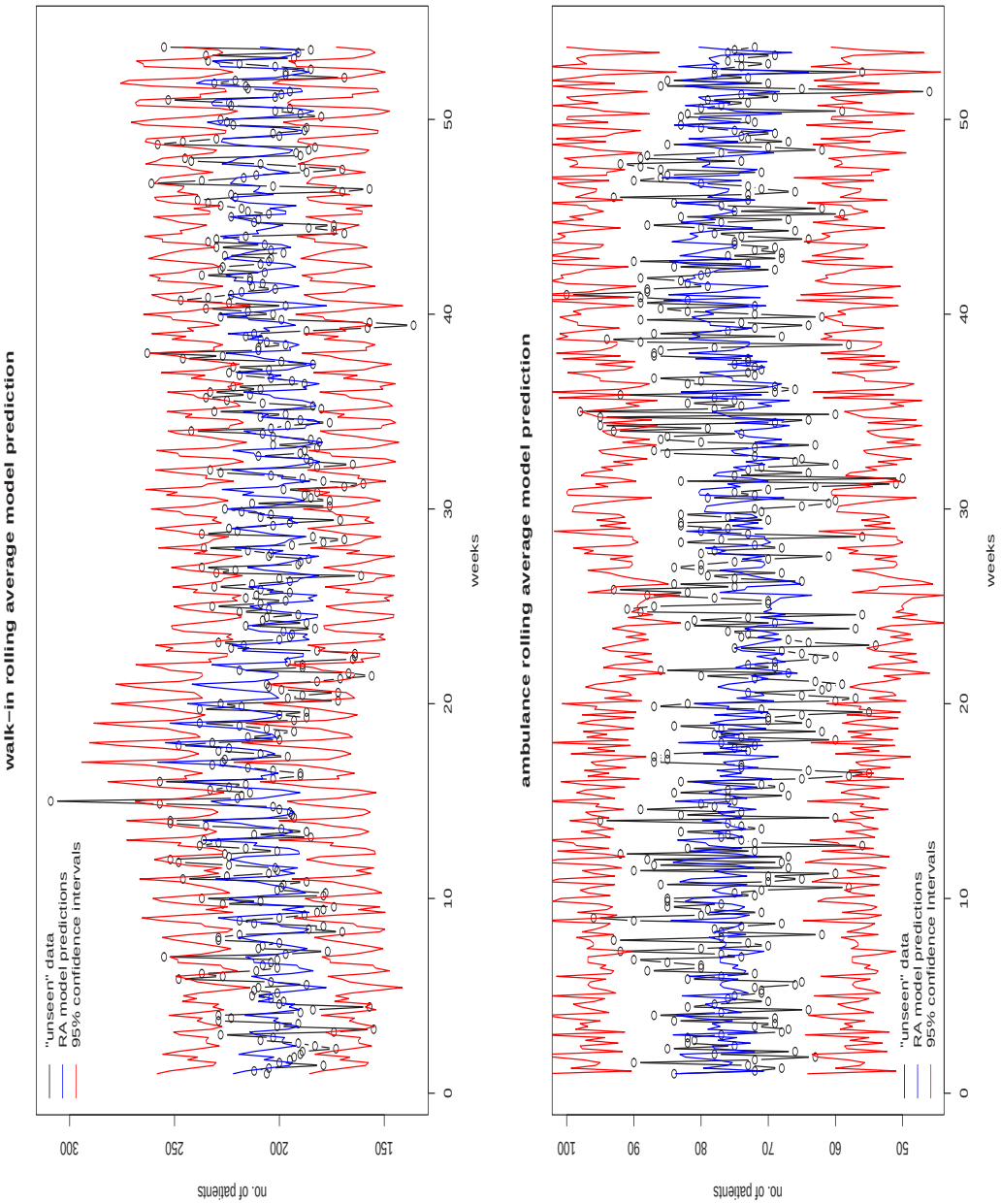


Figure 3.7: The one week ahead walk-in (top) and ambulance (bottom) arrival RA model predictions (in blue) and corresponding 95% confidence intervals (in red), compared with the "unseen" observed patient arrivals (in black).

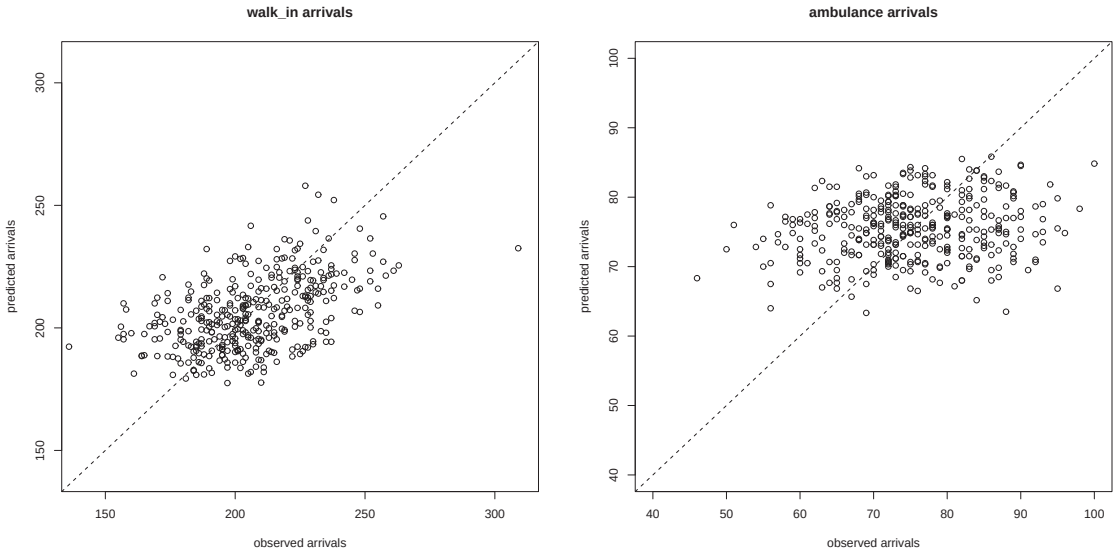


Figure 3.8: Scatterplots comparing the one week ahead RA model predictions for the walk-in (left) and ambulance (right) arrivals with the "unseen" observed patient arrivals.

3.4 Auto-Regressive Models

3.4.1 Specification

In an order q auto-regressive process [54, 19, 25], the current value of the stochastic process x_t of length T is expressed as a finite, linear aggregate of the q previous values of the process and a disturbance term ϵ_t.

Formally an auto-regressive process of order q is written as

$$x_t = \phi_1(x_{t-1} - \mu) + ... + \phi_q(x_{t-q} - \mu) + \epsilon_t, \quad t = 1, ..., T \tag{3.2}$$

where $\psi = \{\phi_1, ..., \phi_q\}$ is the set of adjustable coefficients, μ is the mean of x_t and ϵ_t is a sequence of uncorrelated random variables with mean zero and constant variance.

When fitting auto-regressive models, the time series being modelled is assumed to be stationary (cf. Section 2.2.3). We apply the KPSS stationarity test to both the walk-in and ambulance "training" data, which returns a p value < 0.01 for both. Hence we can reject the null hypothesis that the walk-in and ambulance "training data" are stationary.

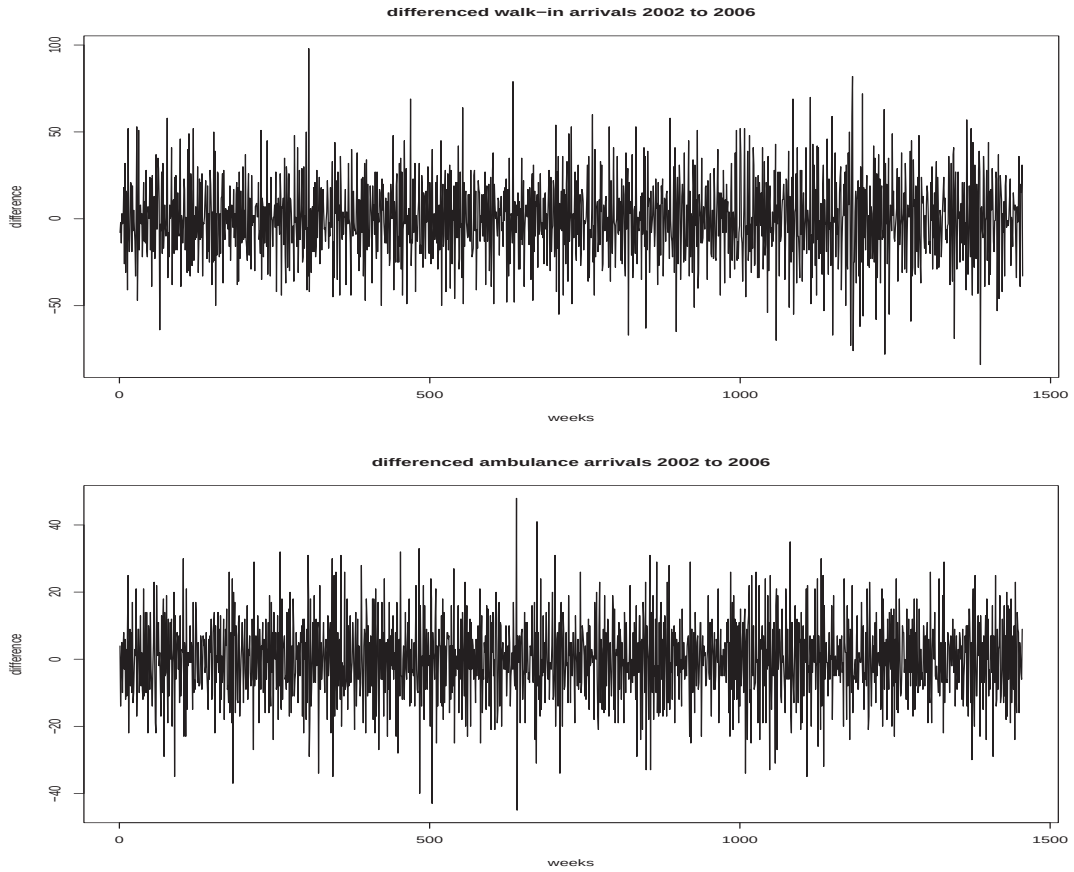


Figure 3.9: The differenced walk-in (top) and ambulance (bottom) arrival "training" data.

To make the walk-in and ambulance arrival "training" data stationary, we difference the data once. That is, given the series x_t, we create the new series $y_t = x_{t+1} - x_t$. The differenced data will contain one less point than the original data, leaving us with 1455 differenced data points of "training" data with which to fit our models. The differenced data is presented in Fig. 3.9. When we now apply the KPSS stationary test to the both the walk-in and ambulance differenced training data, it returns a p value > 0.1 for both. Hence we have no evidence to suggest the differenced training data is not stationary, and we can now fit AR models to this data.

In order to determine the most suitable ψ, we use the method of maximum likelihood (cf. Section 2.2.4). To determine q, the order of the AR model, we must balance the goodness of fit with the complexity of the model. This is formalised in the *Akaike Information Criterion* (AIC).

The *Akaike Information Criterion* (AIC) [5], is the decision rule to select the model for which:

$$AIC = -2logL(\tilde{\psi}) + 2q \tag{3.3}$$

is a minimum; here q is the number of parameters and $L(\tilde{\psi})$ is the maximised value of the likelihood function, where $\tilde{\psi}$ are the maximum likelihood parameter estimates. The first term corresponds to minus 2 times the natural logarithm of the maximised likelihood, while the second term is a "penalty factor" for the inclusion of additional parameters in the model.

The auto-regressive (AR) models are first fitted to the differenced "training" data (the first 1455 differences), using the AIC to determine the order of the models. We use this initial model fit to predict the lth day ahead difference; then we shift ahead into the "unseen" data by l data points and use the previous 1455 differenced data points to fit a new AR model of the same order and again calculate the lth day ahead difference predictions. This is repeated until we have shifted through the remaining 370 days of differenced "unseen" data. The R code used to create and fit these AR models and subsequently perform forecasts is shown in Appendix A.2.

3.4.2 Auto-Regressive Model Fit

Using the AIC (Equation 3.3) we find that the differenced walk-in arrivals are best fitted with a model of order 27 and differenced ambulance arrivals with a model of order 29. To assess the fit of the AR models, we calculate the correlation coefficient (r) of the model fit with the differenced "training" data. We then analyse the residuals (ϵ_t in Equation 3.2) by verifying that they are approximately normally distributed with mean zero, by performing the Ljung-Box test for independence, and by plotting the residuals both against time t and the fitted values to check for any obvious patterns.

The correlation for both the differenced walk-in and ambulance initial fit is reasonable with $r = 0.6984$ for the differenced walk-in arrival model and $r = 0.6693$ for the differenced ambulance arrival model. To determine any remaining structure in the residuals we examine the autocorrelation function (acf) of both sets of residuals, shown

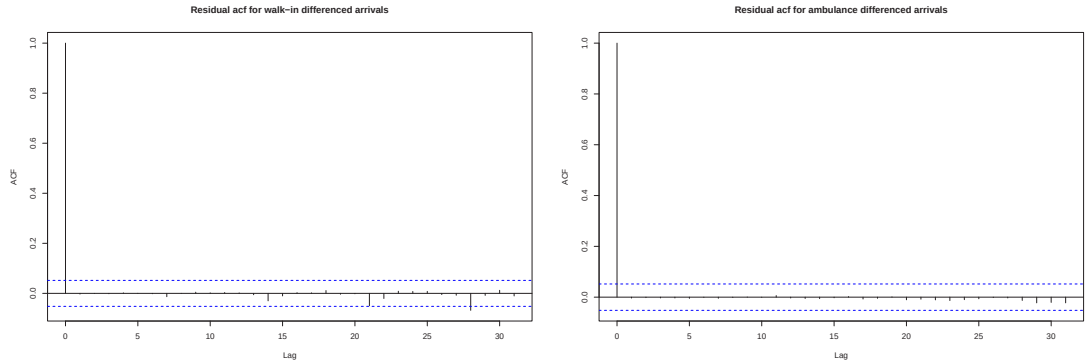


Figure 3.10: The acf of the residuals of the AR models of differenced walk-in (left) and ambulance (right) arrivals.

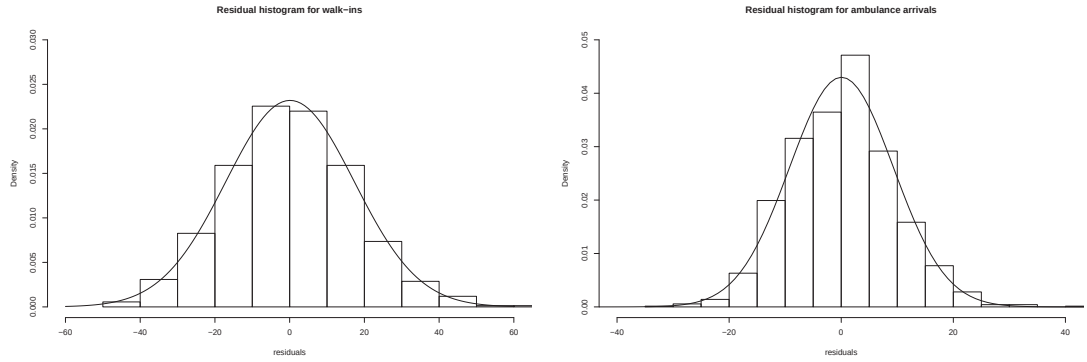


Figure 3.11: The distribution of the residuals of the AR models of differenced walk-in (left) and ambulance (right) arrivals and the corresponding normal distributions.

in Fig. 3.10. We see that there are no significant peaks for up to lag 30 for the differenced ambulance model and only one significant peak for the differenced walk-in model. Fig. 3.11 shows the corresponding histogram of the residuals and also a superimposed normal density with the same mean and standard deviation as the residuals; the close correspondence indicates that it is reasonable to assume the residuals are normally distributed random variables with zero mean. Figs. 3.12 and 3.13 show the residual plots against time and fitted values respectively. These do not show any clear pattern in mean or variance; however, the walk-in model residual plots indicate a number of outliers. The Ljung-Box test returned p value = 0.9954 for the walk-in model residuals and p value = 1 for the ambulance model residuals, indicating that there is no evidence whatsoever to reject the hypothesis that the residuals from both model fits are independently distributed. Plots of the AR difference model fits are shown in Fig. 3.14.

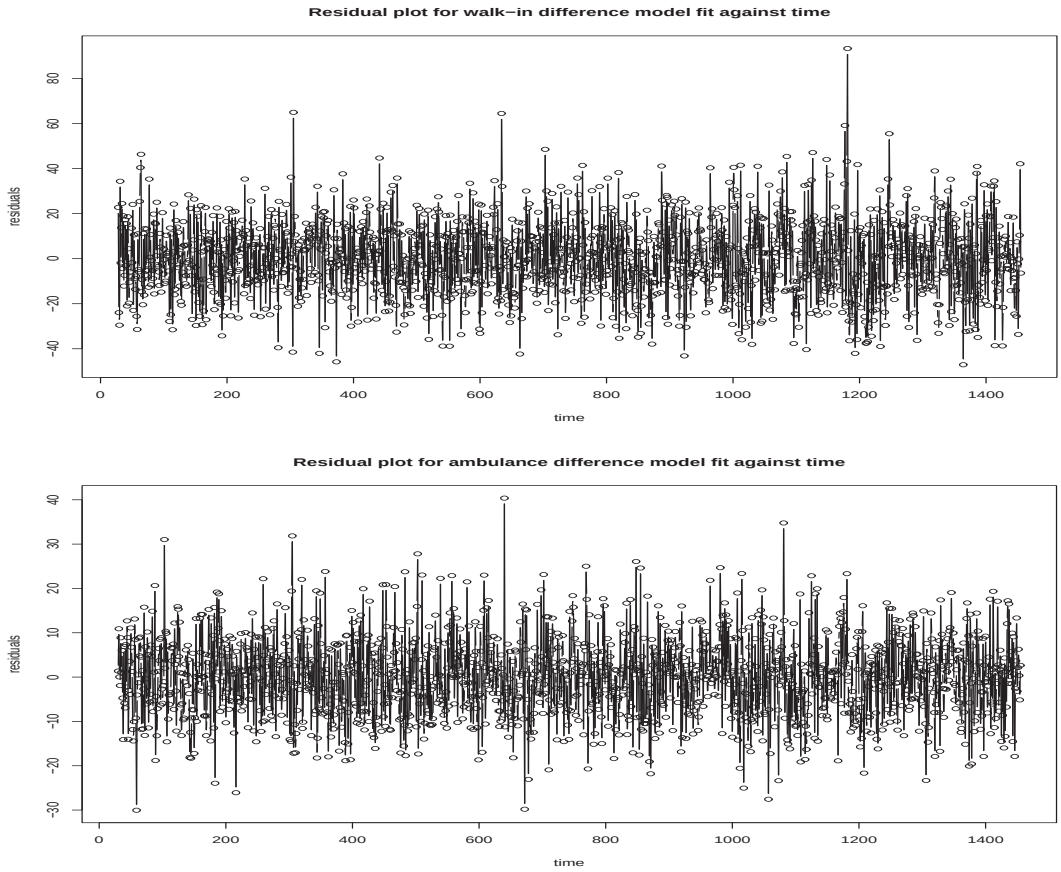


Figure 3.12: The plot of the residuals of the AR models of differenced walk-in (top) and ambulance (bottom) arrivals against time.

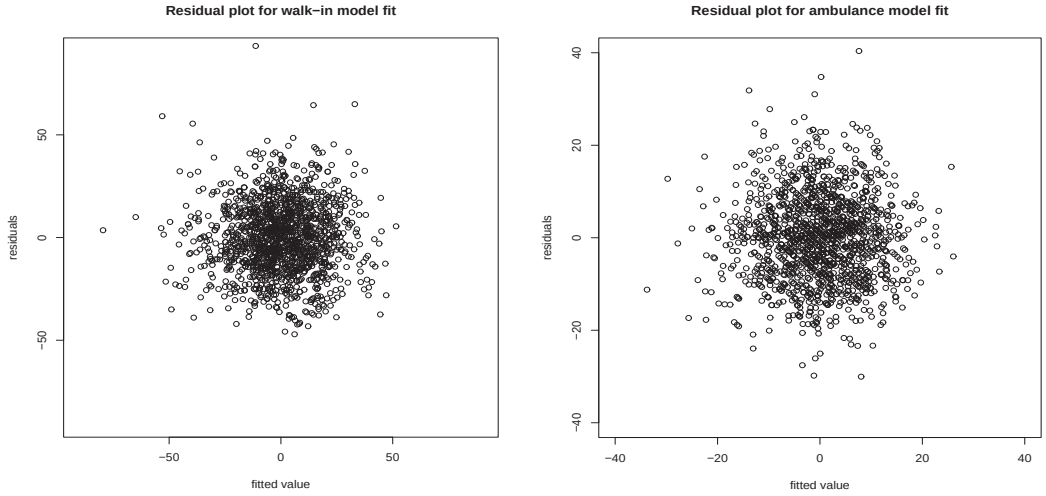


Figure 3.13: Scatterplots of the residuals of the AR models of differenced walk-in (left) and ambulance (right) arrivals against the fitted values.

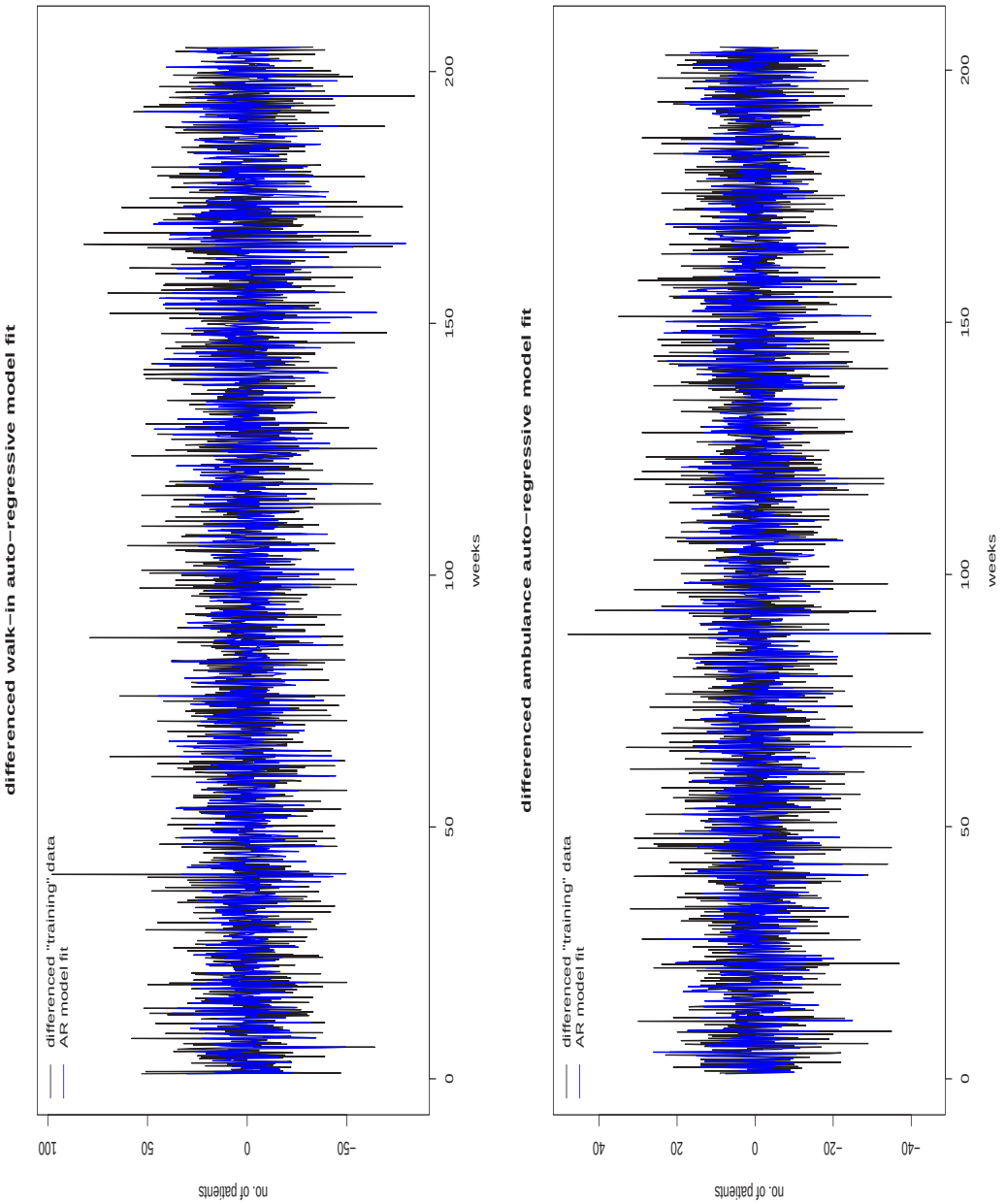


Figure 3.14: The differenced walk-in (top) and ambulance (bottom) AR model fits (in blue) to the "training" data (in black).

3.4.3 Auto-Regressive Model Predictions

In this section we first compare the forecasts made by AR models of differenced arrivals with the "unseen" observed differences. These predicted differences are then summed – if predicting more than one day ahead – and added to the actual number of arrivals on the last known day (i.e. the last data point used to fit the AR model) to get an "undifferenced" prediction for number of arrivals for the lth day ahead. This set of lth day ahead predictions are compared to the observed arrivals for the corresponding days in the "unseen" data. We also compute and compare the entire one week ahead predictions (where all seven predictions when $l = 7$ are taken into account and not just the single lth day ahead prediction) with the "unseen" observed arrivals in order to make a direct quality of forecast comparison with the rolling average model predictions.

Difference Predictions

Table 3.2 presents the quality of forecast measures for the one day ahead AR model difference prediction. We only present the one day ahead difference as difference values are only useful if the number of arrivals for the previous day is known. The one day ahead difference prediction plots and scatterplots are shown in Figs. 3.15 and 3.16 respectively.

AR differenced walk-in model					
l	Mean Bias	RMSE	r	95% CI width	p
1	0.1243	18.3640	0.6691	± 34.7199	0.0649
AR differenced ambulance model					
1	0.2752	9.3754	0.6632	± 18.4156	0.0486

Table 3.2: Quality of forecast measures of the one day ahead AR model difference predictions for differenced walk-in and ambulance arrivals.

Table 3.2 shows that the one day ahead difference predictions by the AR models for both the differenced walk-in and ambulance arrivals show a very slight positive bias. From both Table 3.2 and Fig. 3.16 we can see that both the AR model difference forecasts are good with the correlation with the observed differences being $r = 0.6691$ and $r = 0.6632$ for predicted walk-in and ambulance differences respectively.

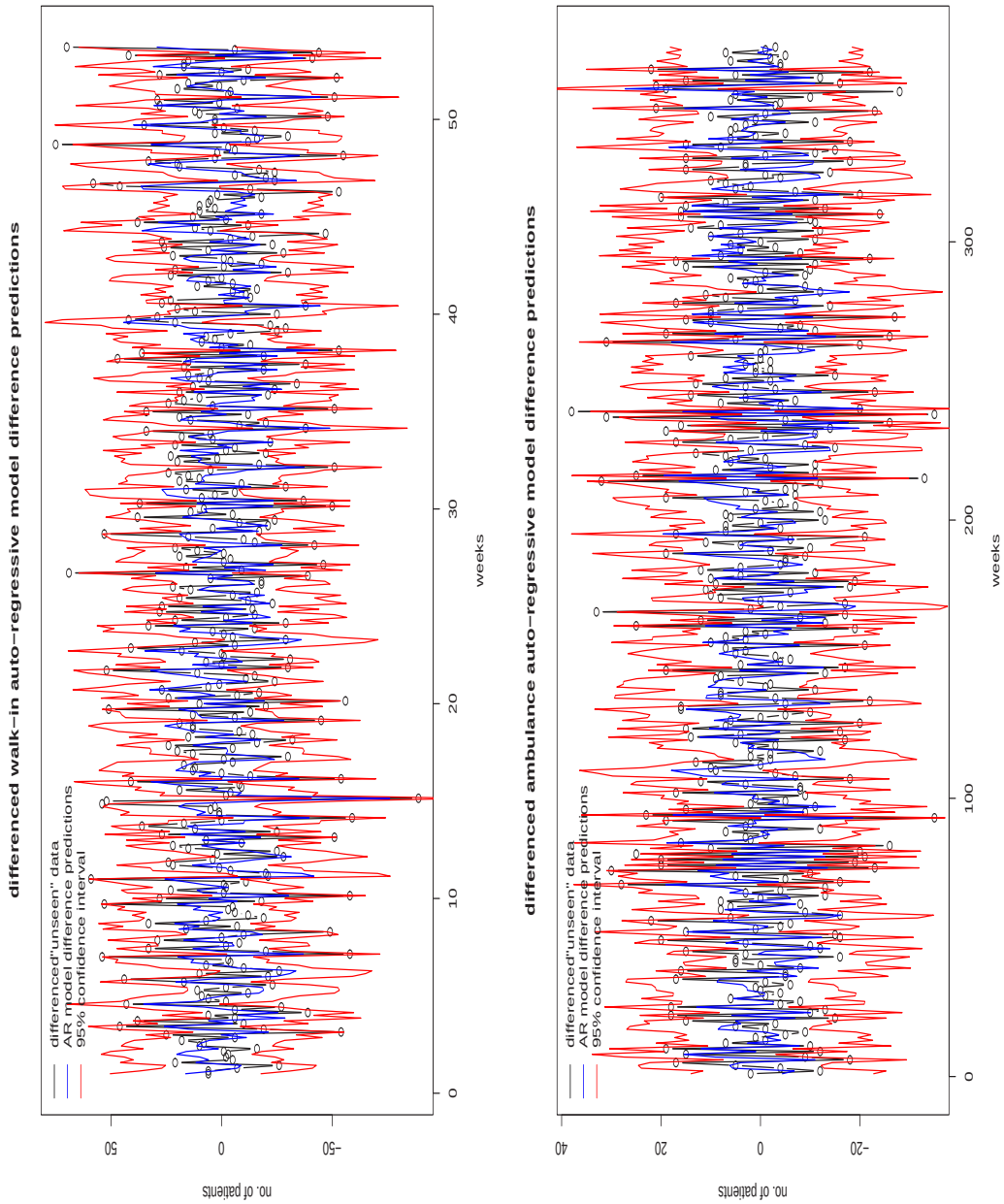


Figure 3.15: The one day ahead AR model difference predictions (in blue) for the walk-in (top) and ambulance (bottom) differences and corresponding 95% confidence intervals (in red), compared with the observed "unseen" differences (in black).

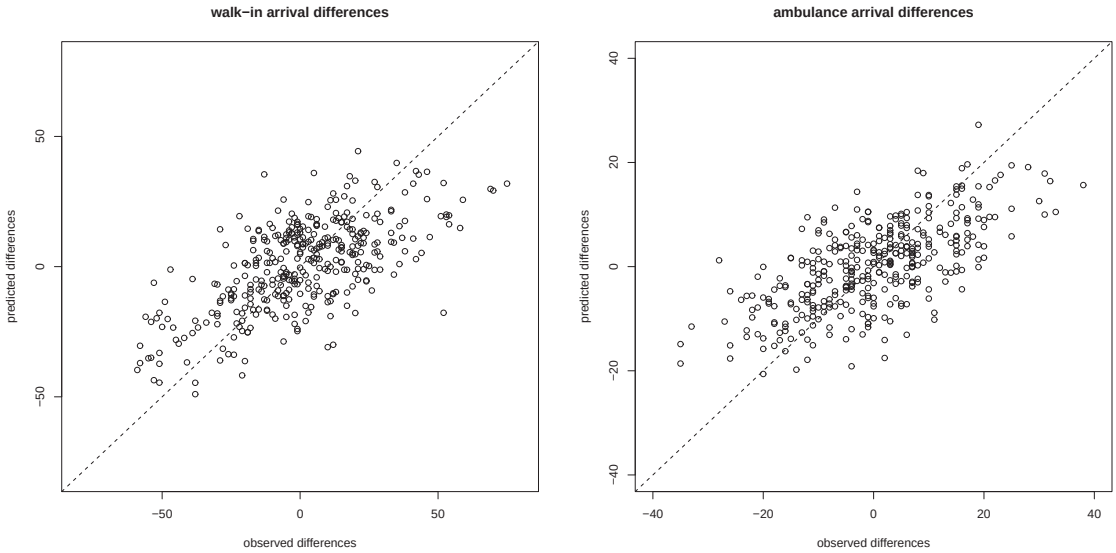


Figure 3.16: Scatterplots comparing the one day ahead difference AR model predictions for walk-in (left) and ambulance (right) differenced arrivals with the "unseen" observed patient arrival differences.

Arrival Predictions

Tables 3.3 and 3.4 present the calculated quality of forecast measures of the one to seven day ahead "undifferenced" AR model predictions. The one day ahead arrival prediction scatter plots and plots are shown in Fig. 3.17 and 3.18. The one week ahead prediction comparison with the "unseen" observed arrivals is presented in Table 3.5 and the corresponding scatterplots are shown in Fig. 3.19.

AR walk-in model					
l	Mean Bias	RMSE	r	95% CI width	p
1	0.1243	18.3640	0.5921	± 34.7199	0.0649
2	0.3012	18.4518	0.5632	± 43.2713	0.0216
3	0.6619	21.4887	0.4994	± 49.5012	0.0163
4	0.7555	19.1508	0.5206	± 53.9496	0
5	1.7516	21.4693	0.5722	± 56.9190	0.0135
6	1.6249	21.2451	0.5364	± 59.5005	0.0164
7	-2.1202	18.5099	0.2704	± 61.3974	0

Table 3.3: Quality of forecast measures of the AR model predictions for walk-in arrivals.

From Tables 3.3 and 3.4 we can see that the AR model forecasts for the walk-in arrivals show mostly a small positive mean bias, while the forecasts for the ambulance arrivals show no trend in the bias. Except for the one day ahead walk-in arrival predictions (where $p = 0.0649$), the p values for both the walk-in and ambulance arrival

AR ambulance model					
l	Mean Bias	RMSE	r	95% CI width	p
1	0.2752	9.3754	0.1863	± 18.4156	0.0486
2	-0.4186	9.2524	0.0836	± 24.4845	0.0108
3	0.0578	9.3764	0.1221	± 28.6854	0.0244
4	-0.6424	9.2733	0.0406	± 31.8526	0
5	0.3283	9.2457	0.2269	± 34.2822	0
6	-0.5308	10.1090	-0.1211	± 36.4353	0
7	0.2966	10.5806	-0.1225	± 37.9654	0

Table 3.4: Quality of forecast measures of the AR model predictions for ambulance arrivals.

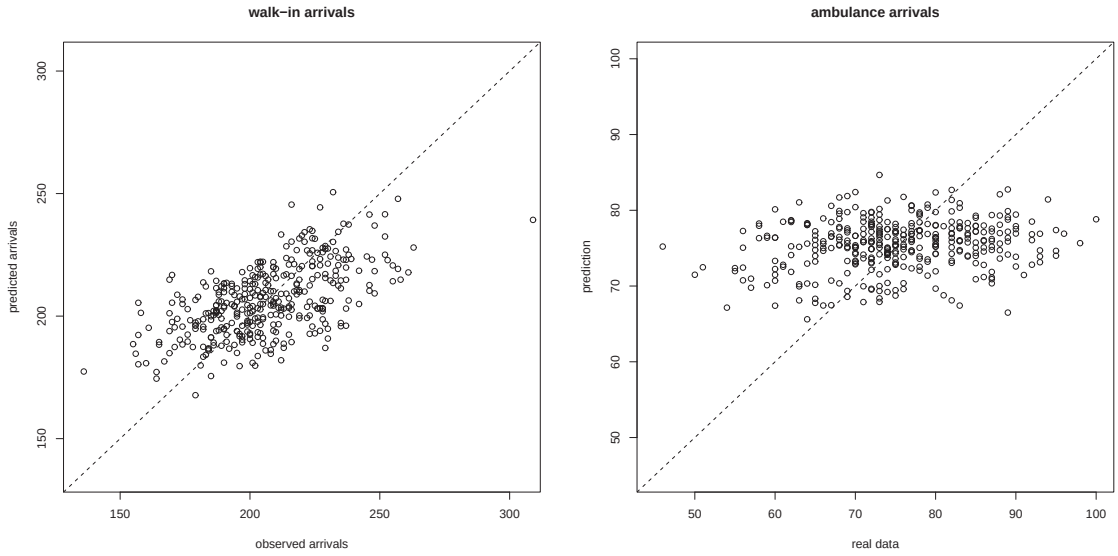


Figure 3.17: Scatterplots comparing the one day ahead AR model predictions for the walk-in (left) and ambulance (right) arrivals with the "unseen" observed patient arrivals.

models are less than 0.05 which indicates that, except for the one day ahead walk-in predictions, at least 95% of the "unseen" observed arrivals are within the calculated 95% confidence intervals for the model predictions. For the one day ahead walk-in arrival predictions 93.51% of the "unseen" observed arrivals are still within the predicted confidence intervals.

Table 3.3 and the scatterplot on the left in Fig. 3.17 show that the AR model of walk-in arrivals perform well, with our one day ahead predictions showing good correlation with the "unseen" data ($r = 0.5921$). As expected, when predicting further ahead the quality of forecast deteriorates for both our walk-in and ambulance arrivals model, with the mean 95% confidence interval widths increasing and r value exhibiting a decreasing

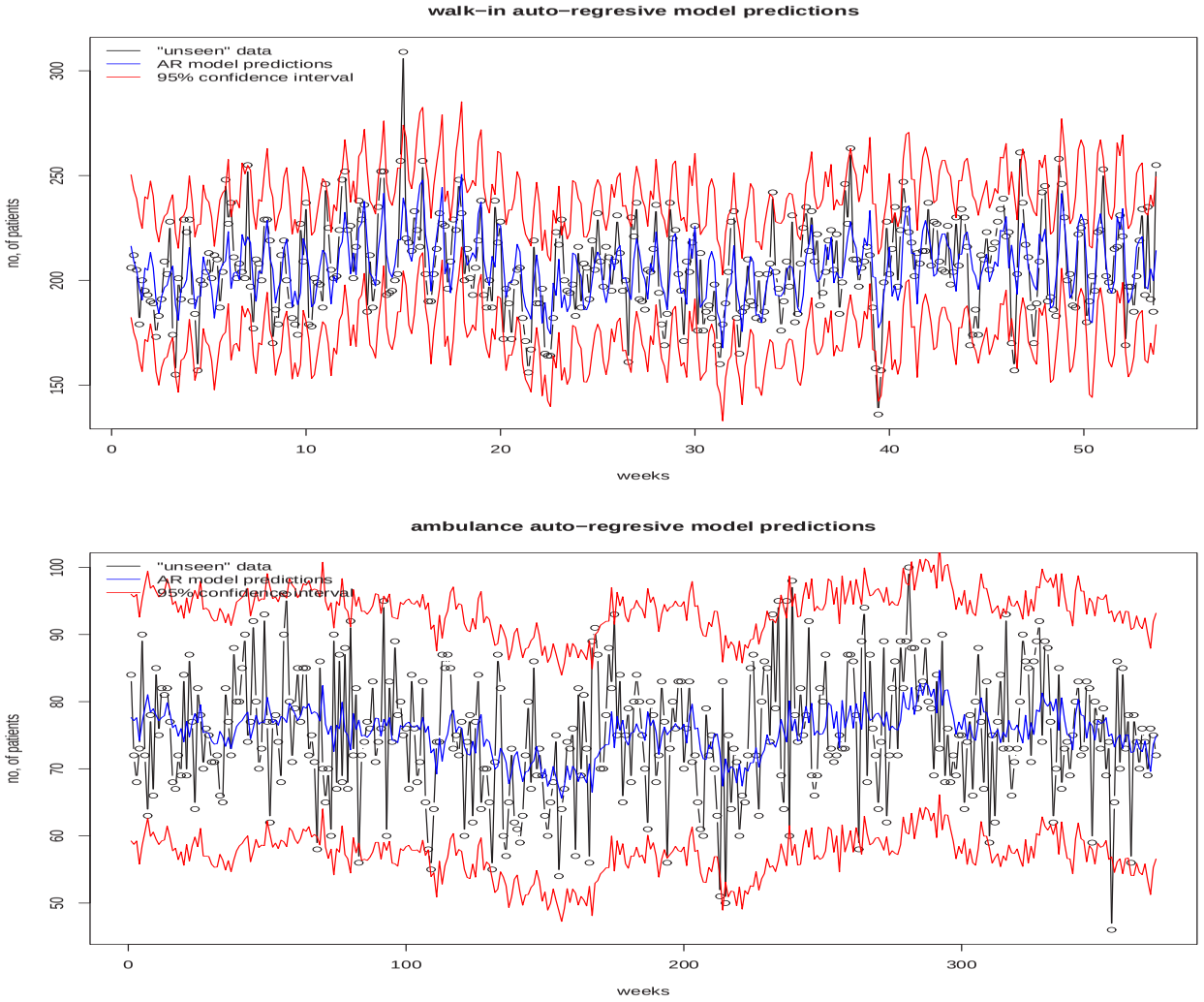


Figure 3.18: The one day ahead AR model predictions (in blue) for walk-in (top) and ambulance (bottom) arrivals and corresponding 95% confidence intervals (in red), compared with the "unseen" observed patient arrivals (in black).

trend for the ambulance model. For the walk-in model this decrease in r value is slow up until forecast horizon $l = 6$ (where $r = 0.5364$), and dips sharply at $l = 7$ (where it decreases to $r = 0.2704$). In contrast with differenced ambulance model (cf. Table 3.2 and Fig. 3.16), the quality of the AR model ambulance arrivals predictions are poor. As shown in Table 3.4, the one day ahead predictions show poor correlation with the "unseen" observed arrivals ($r = 0.1863$); however the correlation is slightly improved for the fifth day ahead prediction ($r = 0.2269$). This reasonable performance for the walk-in arrival model and poor performance by the ambulance arrival model is reinforced by the one day ahead scatterplots shown in Fig. 3.17.

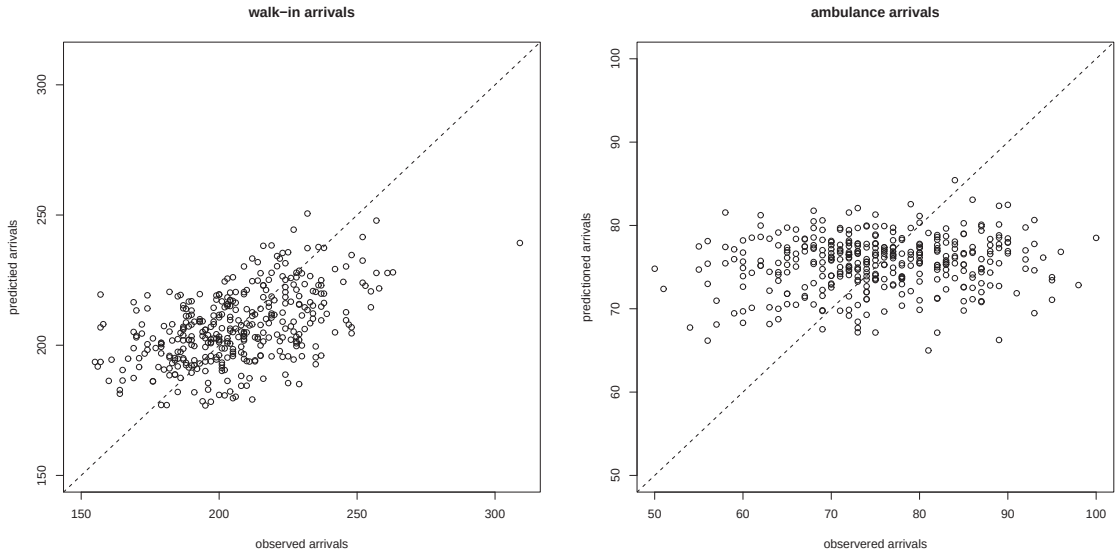


Figure 3.19: Scatterplots comparing the one week ahead AR model predictions for walk-in (left) and ambulance (right) arrivals with the "unseen" observed patient arrivals.

AR walk-in model				
Mean Bias	RMSE	r	95% CI width	p
0.5690	19.8089	0.5049	± 51.2613	0.0243
AR ambulance model				
0.2697	9.5323	0.1360	± 30.2432	0.0108

Table 3.5: Quality of forecast measures of the one week ahead AR model predictions for walk-in and ambulance arrivals.

The quality of forecast measures of the week ahead "undifferenced" AR model predictions for both arrival streams are shown in Table 3.5 and the scatterplots in Fig. 3.19. Again we can see that we have a reasonable fit for the walk-in arrivals but a poor fit for the ambulance arrivals. Both the AR model predictions show a slight positive

mean bias and the RMSE for both the walk-in and ambulance predictions are lower than that for the corresponding RA models. The one week ahead walk-in predictions show a slightly better correlation with the "unseen" data than for the corresponding RA model ($r = 0.5049$ compared with $r = 0.4965$); however, this correlation is worse for the ambulance arrival model ($r = 0.1360$ compared with $r = 0.1608$). The 95% confidence interval widths get much wider as we predict further ahead, and are nearly two times wider for both sets of one week ahead AR model predictions than for the corresponding RA model predictions. This explains the very low p values observed for the one week ahead AR models.

3.5 Structural Times Series Models

3.5.1 Specification

Intuitively a structural time series model [56] can be thought of as a regression model in which the explanatory variables are functions of time and the parameters are time varying. For our models we have used the classical decomposition in which the series is seen as the sum of trend, seasonal and irregular components. Thus:

$$x_t = \mu_t + \gamma_t + \epsilon_t, \quad t = 1, ..., T \tag{3.4}$$

where μ_t is the trend, γ_t is the seasonal component and ϵ_t is the irregular component. All three components are stochastic and the disturbances driving them are mutually uncorrelated. The irregular component is white noise, that is a sequence of uncorrelated random variables with constant mean and variance.

A deterministic linear trend is given by $\mu_t = \alpha + \beta t$. Since μ_t may be obtained recursively from $\mu_t = \mu_{t-1} + \beta$ with $\mu_0 = \alpha$, continuity may be preserved by introducing stochastic terms as follows:

$$\mu_t = \mu_{t-1} + \beta_{t-1} + \eta_t, \quad \beta_t = \beta_{t-1} + \xi_t \tag{3.5}$$

where η_t and ξ_t are mutually uncorrelated white noise disturbances with zero means and with variances σ_η^2 and σ_ξ^2 respectively. The effect of η_t is to allow the level of the trend to shift up and down, while ξ_t allows the slope to change. The larger the variances, the greater the stochastic movements in the trend. If $\sigma_\eta^2 = \sigma_\xi^2 = 0$ then the stochastic trend collapses to the deterministic trend. The forecasts from such a model put more weight on the most recent observations; the faster the level and slope change the more past observations are discounted.

Other components can be added to the model such as a seasonal component. A model of deterministic seasonality has the seasonal effects summing to zero over a year. The seasonal effects can be allowed to change over time by letting their sum over the previous year be equal to a random disturbance term ω_t, with mean zero and variance σ_ω^2. Thus, if s is the number of seasons in the year:

$$\gamma_t = -\sum_{j=1}^{s-1} \gamma_{t-j} + \omega_t \tag{3.6}$$

Fitting of structural time series models is a complex (but readily automated) procedure involving determination of a likelihood function using a Kalman filter (an efficient recursive filter that estimates the state of a dynamical system from previous measurements) and numerical optimisation. Full details of this procedure can be found in Harvey's book [56]. Predictions are made by extrapolating estimated components into the future. For these ST models we incorporated only a weekly periodicity and not the annual periodicity indicated by the power spectra (cf. Fig. 3.2) as short term forecasts will be dominated by the weekly periodicity.

We use the "training" data to fit a structural time series model with a seven day (weekly) periodicity. We then use this model to predict the number of arrivals for the lth day ahead ($l = 1, 2. \ldots, 7$); we then shift ahead into the remaining data by l data points and use the next 1456 data points of observed arrivals to fit a new ST model and again calculate the lth day ahead prediction. This is repeated until we have shifted through the remaining 370 days of data. This set of lth day ahead predictions is then compared to the actual arrivals for the corresponding days in the "unseen" data. We also compute and then compare the entire week ahead predictions with the "unseen"

data in order to make a direct quality of forecast comparison with the rolling average model predictions. The *R* code used to create and fit these ST models and subsequently to perform forecasts, is shown in Appendix A.3.

3.5.2 Structural Times Series Model Fit

The Pearson product-moment correlation coefficient (r) of the "training" data with the corresponding model fit is good for the walk-in model with $r = 0.9535$, which indicates a very good fit. For the ambulance model we have $r = 0.6157$, which indicates a reasonable initial fit.

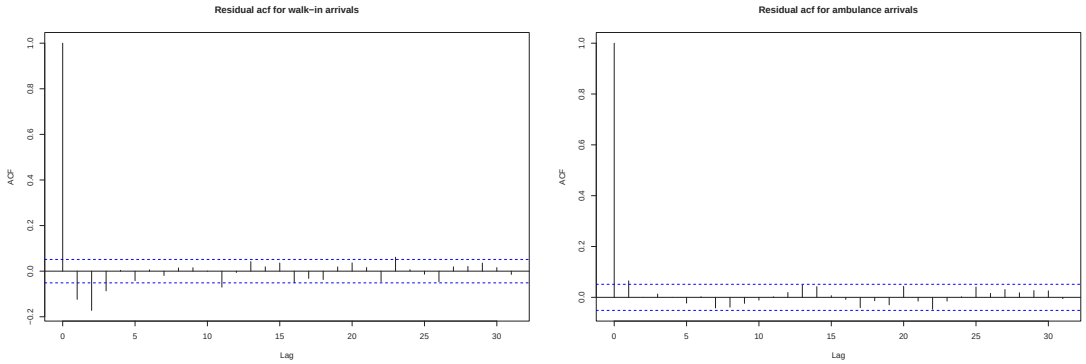


Figure 3.20: The acf of the residuals of the ST models of walk-in (left) and ambulance (right) arrivals.

As for the previous models we check the residuals to see if there is any unexplained structure not captured by the models. The acfs of the residuals are shown in Fig. 3.20. For the walk-in arrivals (shown on the left) there are some peaks crossing the 95% confidence interval – indicating that there may be some further dependencies within the data not yet incorporated into our model; for the ambulance model we see only one significant peak, so it is reasonable to assume that the residuals are uncorrelated. Fig. 3.21 show plots of the initial ST model fit to the "training" data for both the walk-in and ambulance arrivals. Fig. 3.22 shows the corresponding histogram of the residuals and also a superimposed normal density with the same mean and standard deviation as the residuals. The close correspondence for both arrival types indicate that it is reasonable to assume that the residuals are normally distributed around mean zero. Figs. 3.24 and 3.23 and show the corresponding residual plots against time and fitted values respectively. These do not show any clear pattern in mean or variance; however,

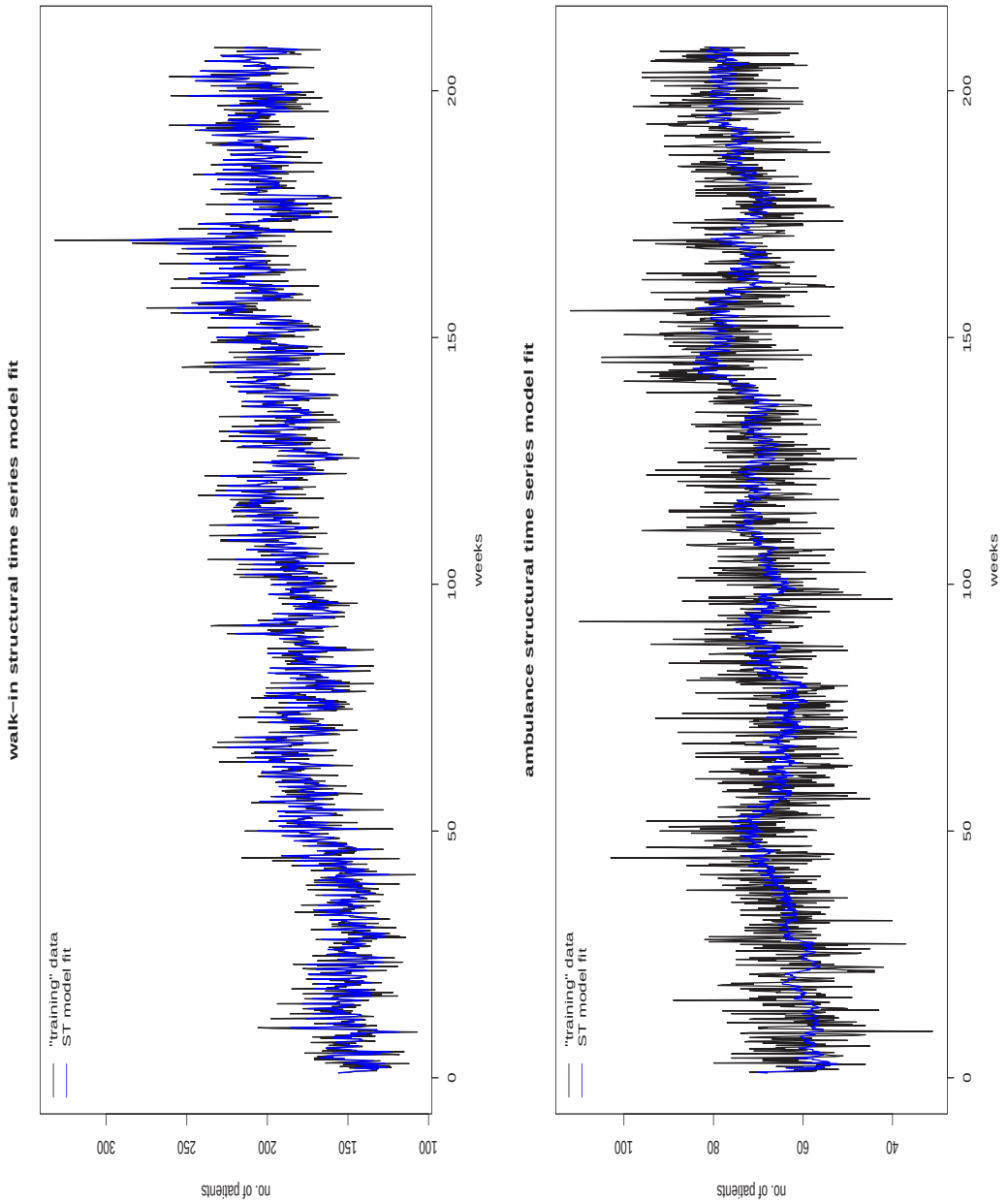


Figure 3.21: The walk-in (top) and ambulance (bottom) arrival ST model fit (in blue) to the "training" data (in black).

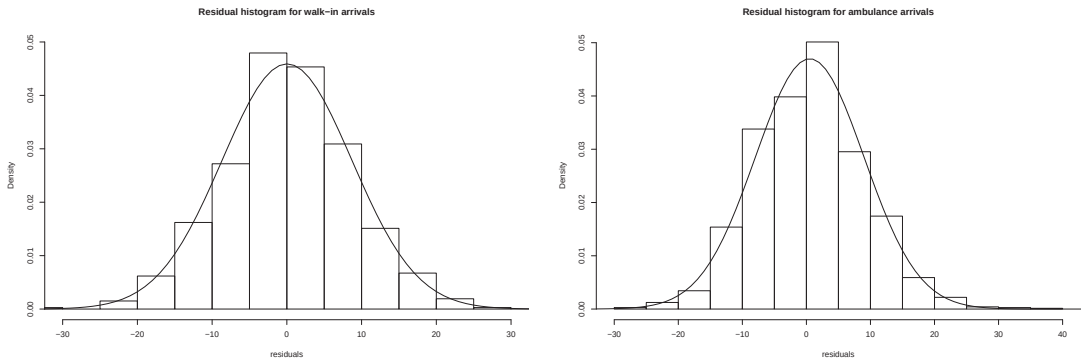


Figure 3.22: The distribution of the residuals of the ST models of walk-in (left) and ambulance (right) arrivals and the corresponding normal distributions.

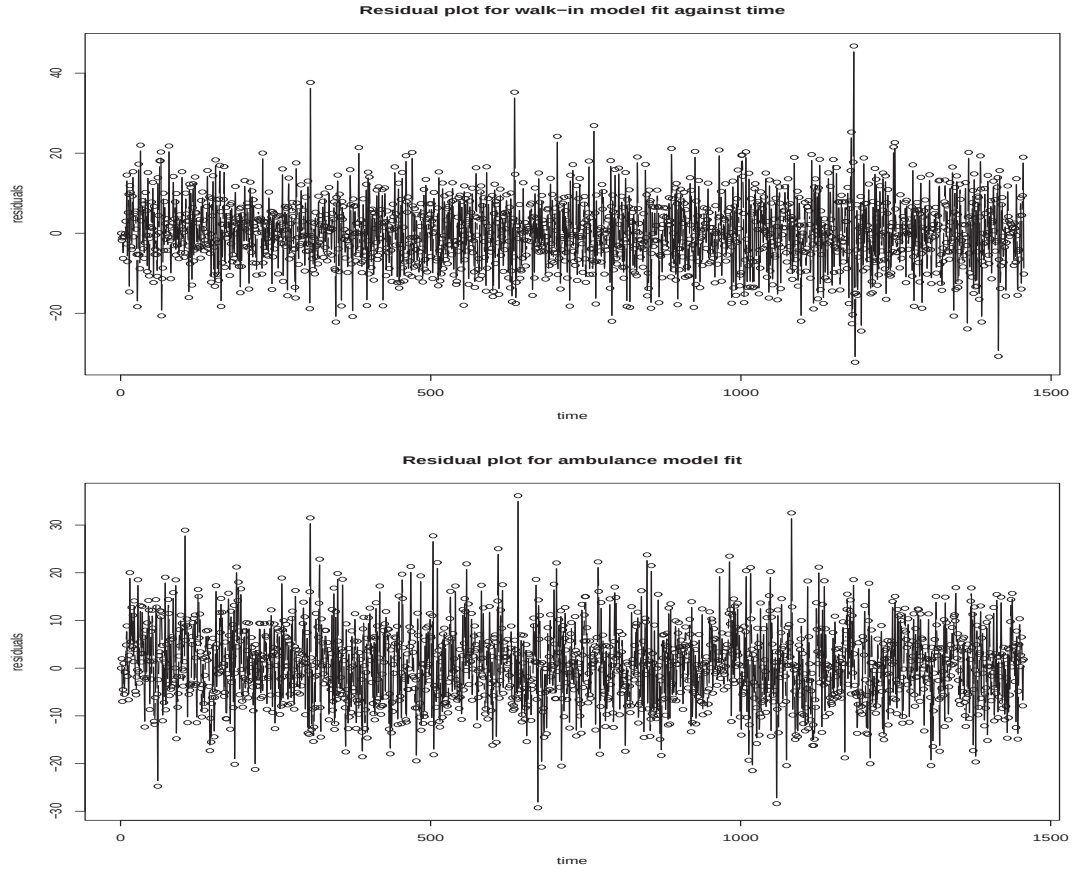


Figure 3.23: The plot of the residuals of the ST models of walk-in (top) and ambulance (bottom) arrivals against time.

the walk-in model residual plots indicate a number of outliers. The Ljung-Box test returned p value $\ll 0.0001$ for the walk-in model residuals and p value $= 0.5316$ for the ambulance model residuals. This means we can reject the null hypothesis that the walk-in model residuals are independent (as was indicated by the corresponding acf), but for the ambulance model residuals we cannot reject the null hypothesis.

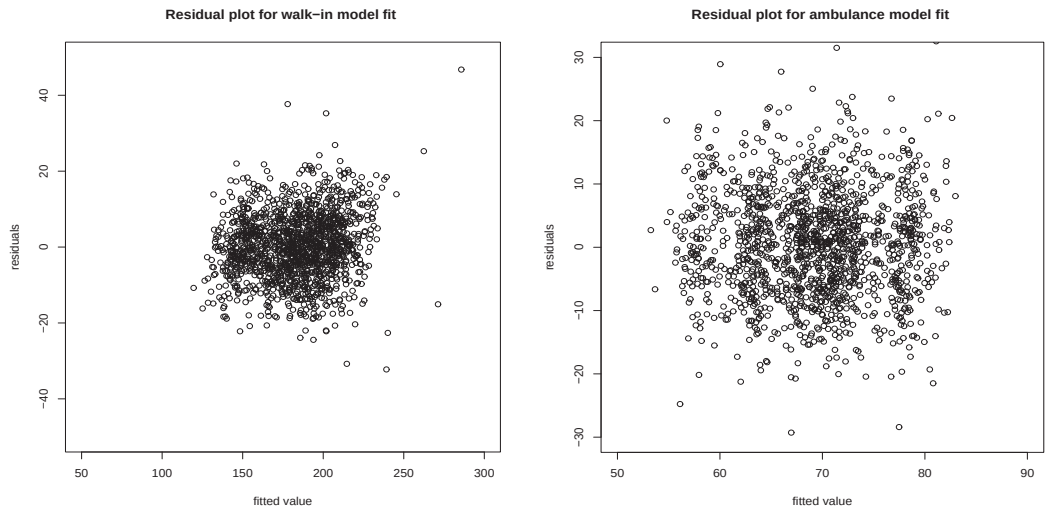


Figure 3.24: Scatterplots of the residuals of the ST models of walk-in (left) and ambulance (right) arrivals against fitted value.

3.5.3 Structural Time Series Model Predictions

Tables 3.6 and 3.7 show the quality of forecast measures of the one to seven day ahead ST model predictions. The one day ahead arrival prediction scatterplots and plots are shown in Figs. 3.25 and 3.26 respectively. The one week ahead prediction comparison with the "unseen" observed arrivals is presented in Table 3.8 and the corresponding scatterplots are shown in Fig. 3.27.

ST walk-in model					
l	Mean Bias	RMSE	r	95% CI width	p
1	0.6257	18.0185	0.6205	± 34.3215	0.0676
2	0.5042	17.7694	0.6095	± 35.2163	0.0541
3	0.9841	20.8653	0.5426	± 36.0673	0.0622
4	1.4213	19.2425	0.5264	± 36.9521	0.0568
5	1.6864	20.6583	0.6114	± 37.9560	0.0595
6	1.4356	21.0688	0.5534	± 38.4447	0.0514
7	-1.1342	19.5209	0.2608	± 55.3970	0.0297

Table 3.6: Quality of forecast measures of the ST model predictions for walk-in arrivals.

From Tables 3.6 and 3.7 it is apparent that the ST model forecasts for the walk-in arrivals show mostly a small positive mean bias, while the forecasts for the ambulance arrivals show no trend in the bias. The p values for the walk-in arrivals tend to be very slightly over 0.05 which indicates that the 95% confidence intervals of the model predictions are slightly too narrow; the largest p value is $p = 0.0676$ (for the one day

ST ambulance model					
l	Mean Bias	RMSE	r	95% CI width	p
1	0.2944	9.0364	0.2951	± 18.3133	0.0324
2	-0.5935	9.0146	0.2030	± 18.3456	0.0297
3	-0.0380	9.2714	0.1873	± 18.4050	0.0270
4	-0.8125	8.7768	0.2318	± 18.4465	0.0297
5	0.2909	8.8092	0.3478	± 18.4977	0.0297
6	-0.0374	10.1434	-0.0374	± 18.5300	0.0351
7	1.0104	10.1481	-0.0443	± 18.6486	0.0405

Table 3.7: Quality of forecast measures of the ST model predictions for ambulance arrival models.

ahead predictions) which indicates that 93.24% of the "unseen" observed arrivals are inside the 95% confidence intervals for that set of predictions. For the ambulance arrivals the p values are less than 0.05 for all forecasts.

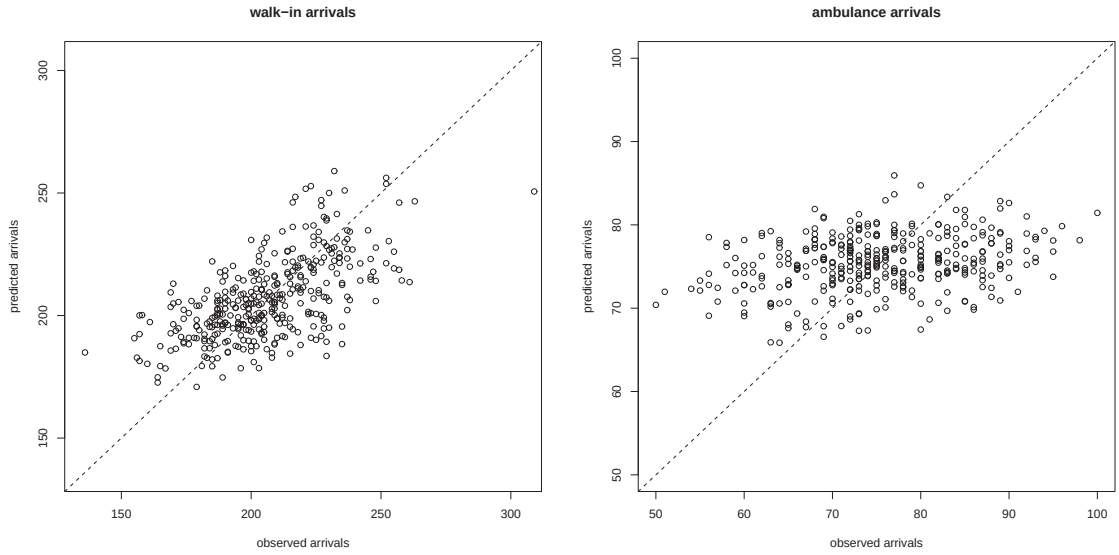


Figure 3.25: Scatterplots comparing the one day ahead ST model predictions for walk-in (left) and ambulance (right) arrivals with the "unseen" observed patient arrivals.

Table 3.6 and the scatterplot on the left in Fig. 3.25 show that the ST model of walk-in arrivals performs well when forecasting, with one day ahead predictions showing good correlation with the "unseen" data ($r = 0.6205$). As expected, when predicting further ahead the quality of forecast deteriorates; however, for up to six days ahead this deterioration is slow, with a trend towards the mean 95% confidence interval widths increasing slightly and r value slowly decreasing. For a seven day forecast horizon, we can see that the r value decreases sharply and the mean confidence interval width increases steeply. This may be related to the strong seven day seasonality in the walk-in ST

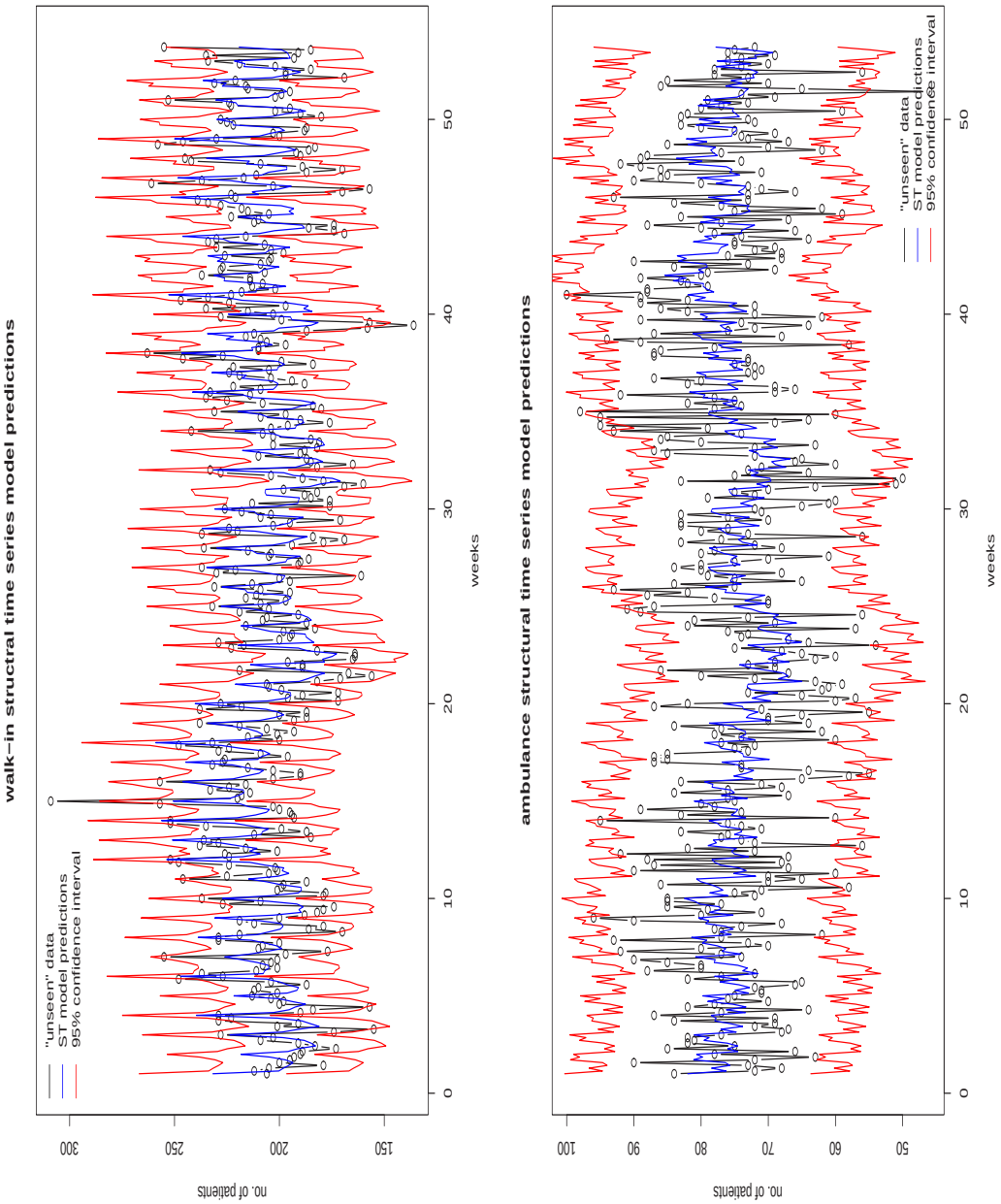


Figure 3.26: The one day ahead ST model predictions (in blue) for the walk-in (top) and ambulance (bottom) arrivals and the corresponding 95% confidence intervals (in red), compared with the observed "unseen" arrivals (in black).

model. From Table 3.7 we see that the quality of the ST ambulance model predictions are not as good, with our one day ahead predictions showing poor correlation with the "unseen" data (r = 0.2951) while the sixth and seventh day ahead forecasts show no correlation to the actual arrivals (where r = -0.0374 and r = -0.0443 respectively). This is reinforced by the one day ahead scatterplot shown on the right in Fig. 3.25.

The quality of forecast measures of the week ahead ST model predictions for both arrival streams are shown in Table 3.8 and the scatterplots in Fig. 3.27.

ST walk-in model				
Mean Bias	RMSE	r	95% CI width	p
4.8328	21.0724	0.5017	± 46.0742	0.0297
ST ambulance model				
0.1350	9.2118	0.2503	± 18.4920	0.0405

Table 3.8: Quality of forecast measures of the one week ahead ST model predictions for walk-in and ambulance arrival models.

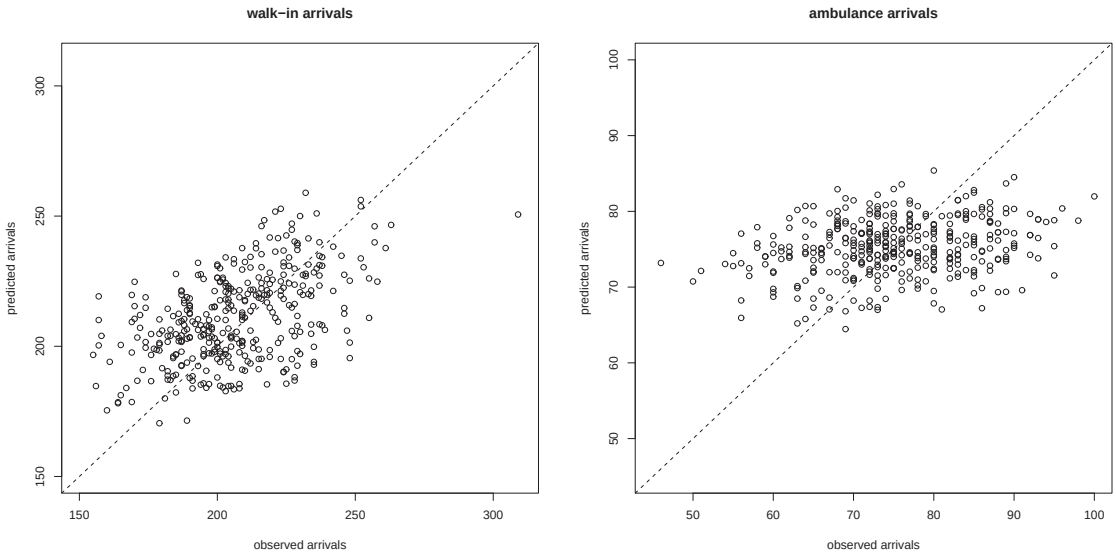


Figure 3.27: Scatterplots comparing the one week ahead ST model predictions for walk-in (left) and ambulance (right) arrivals with the "unseen" observed patient arrivals.

Again we can see that we have a reasonable fit for the walk-in arrivals but a poor fit for the ambulance arrivals. Both sets of the one week ahead ST model predictions show a positive mean bias with the bias of the walk-in predictions much larger. The RMSE for the walk-in predictions is slightly higher than that of the walk-in RA model predictions and slightly lower than that of the ambulance RA model predictions. The one week ahead predictions for both arrival streams show a slightly better correlation with the

"unseen" observed data than for the corresponding RA model ($r = 0.5017$ compared with $r = 0.4965$ for the walk-in predictions and $r = 0.2503$ compared with $r = 0.1608$ for the ambulance predictions). The p values are slightly lower than 0.05 for both sets of arrival model predictions, as would be expected.

3.6 Non-homogeneous Poisson Process Model

Since the time series models in the previous section failed to characterise ambulance arrivals adequately we investigate the possibility that daily ambulance arrivals may be characterised by certain classes of stochastic process.

3.6.1 Poisson Processes

We first explore the possibility that the ambulance arrivals may be characterised by the well known Poisson process. As defined in Section 2.3.2, the Poisson process is a counting process for the number of randomly occurring events observed in a given interval of time and was famously used by Ladislaus von Bortkewitsch in 1898 to describe deaths due to horse kicks in the Prussian cavalry [18].

The Poisson process has a constant arrival rate λ. From Fig 3.28 we can see that the daily ambulance arrivals exhibit an increasing linear trend (shown in red); this means that daily ambulance arrivals have an increasing daily arrival rate. In addition to this the mean and variance of the Poisson process are identical and each equal to λt. The mean and variance of daily ambulance arrivals for the time period 1 April 2002 to 31 March 2007 are 70.7782 and 115.353 respectively, indicating that daily ambulance arrivals are unlikely to be characterised by a Poisson process with constant rate λ. Furthermore when we applied a χ^2 goodness of fit test to test the hypothesis that the ambulance data comes from a Poisson distribution with $\lambda = 70.7782$, we obtain a test statistic of 2974.35. The critical value for this test (a χ^2 distribution with 1825 degrees of freedom) at $p \leq 0.05$ is 1926, further confirming that it is not reasonable to assume that ambulance arrivals are taken from a Poisson distribution.

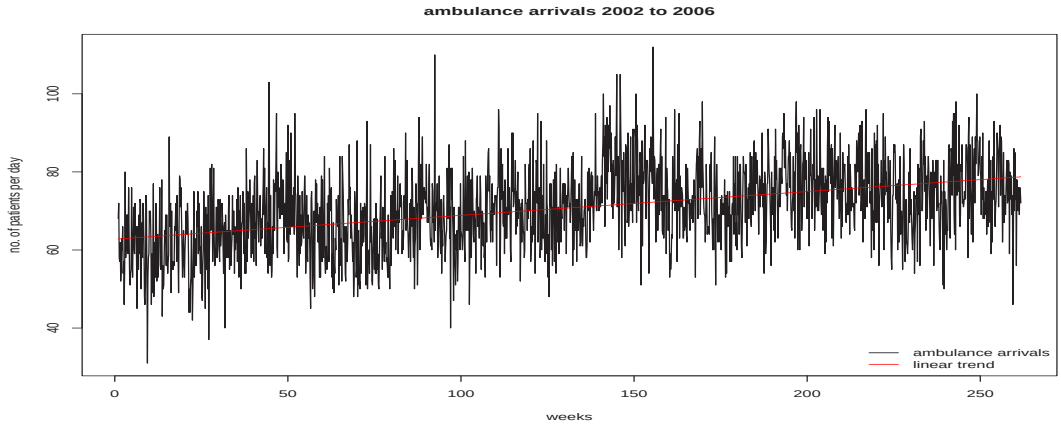


Figure 3.28: Daily ambulance arrivals (in black) and linear trend (in red) for 2002-2007.

3.6.2 Non-homogeneous Poisson Processes

In the previous section we found that ambulance arrivals are not characterised by a Poisson process with constant rate λ. We now consider the non-homogenous Poisson process (NHPP), which is a generalisation of the Poisson process where the rate $\lambda(t)$ is a deterministic function of time. For daily ambulance arrivals we fit a NHPP with a linear rate [72]. In particular we assume that we have a NHPP over the interval $[0, T]$ with arrival rate function:

$$\lambda(t) = a + bt \quad 0 \leq t \leq T \tag{3.7}$$

The overall time interval $(0, T]$ is divided into N measurement subintervals $(\frac{(k-1)T}{N}, \frac{kT}{N}]$, where $1 \leq k \leq N$ and we observe the number of patient arrivals in each. To test the goodness of fit we calculate:

$$U \equiv \sum_{k=1}^{N} \sigma_k^{-2} \Big(Y_k - \sigma_k^2 \frac{T}{N} \Big)^2, \quad \text{where} \quad \sigma_k^2 = (a + bx_k)\frac{T}{N} \tag{3.8}$$

where Y_k is the number of observations in the k^{th} subinterval and $x_k = (k - \frac{1}{2})\frac{T}{N}$. If the Y_k is a linear non-homogeneous Poisson process, U should be approximately χ^2 with $N - 2$ degrees of freedom.

For ambulance arrivals we have five years of daily observations giving us $N = T = 1826$. We fit a linear trend to the ambulance arrivals using linear least squares regression to find the values of a and b giving us: $\lambda(t) = 62.813 + 0.0087t$. From Equation 3.8 we

have $U = 2439.761$ and $N - 2 = 1824$ degrees of freedom. The χ^2 distribution with 1824 degrees of freedom has at $p \leq 0.05$ a critical value of 1924. Our U value is much higher than this, showing that the daily ambulance arrivals has significant departures from the linear non-homogeneous Poisson model.

There are papers which describe the fitting and simulation of non-homogeneous Poisson processes with cyclic or periodic behaviour [65, 66]. This may prove to be a promising future line of enquiry in this context. However, there is no goodness of fit test available as yet and what little software has been developed cannot accommodate the large number of observations in our ambulance arrivals data.

3.7 Hourly Arrivals

We now investigate daily patient arrivals into our case study A&E department by hour. Instead of forecasting hourly arrivals into our department, we will look at the percentage of arrivals that arrive by hour during the day for each arrival stream. Hourly patient arrivals into an American Emergency Department have been previously forecasted [95] but this did not take to account different patient arrival types.

First we plot the percentage of patient arrivals arriving by hour separately for the each day of the week, broken down by year of arrival for the time period 1 April 2002 to 31 March 2007. This is to determine if the hourly arrival patterns alter by year for each arrivals stream. Appendix B.1 shows the plots for walk-in arrivals and Appendix B.2 shows the corresponding plots for the ambulance arrivals. From these plots we can see that the hourly arrival patterns remain relatively unchanged for each of the years of data for both arrival streams. Hence we aggregate the percentage arrivals by hour for each day of the week for all five years of arrivals data; this is shown in Fig. 3.29.

From Fig. 3.29 and the plots in Appendices B.1 and B.2 we can see that the hourly arrival patterns for weekdays differ from weekends for the walk-in and ambulance arrivals, with a higher percentage of arrivals in the early hours for both walk-in and ambulance arrivals during the weekends relative to the weekdays. For this reason we group weekday hourly arrivals and weekend hourly arrivals as shown in Fig. 3.30.

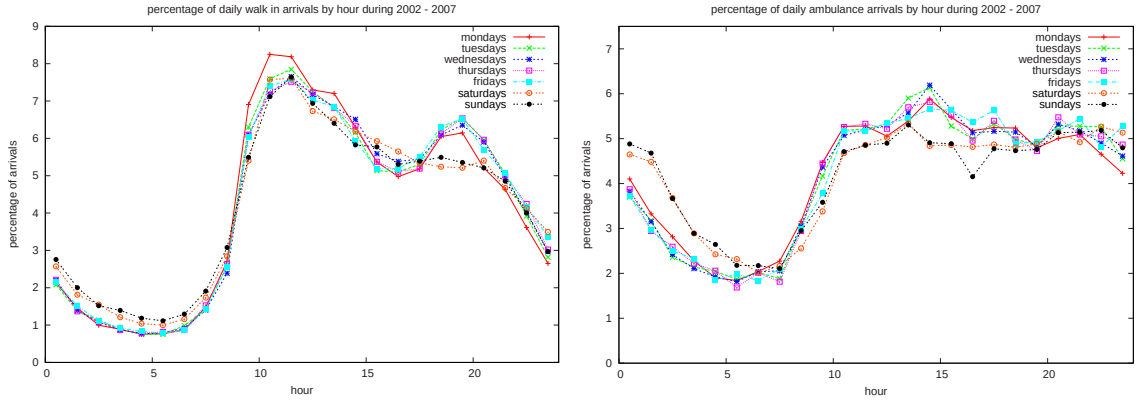


Figure 3.29: Plots of the percentage of walk-in (left) and ambulance (right) arrivals by hour for each day of the week during 2002-2007.

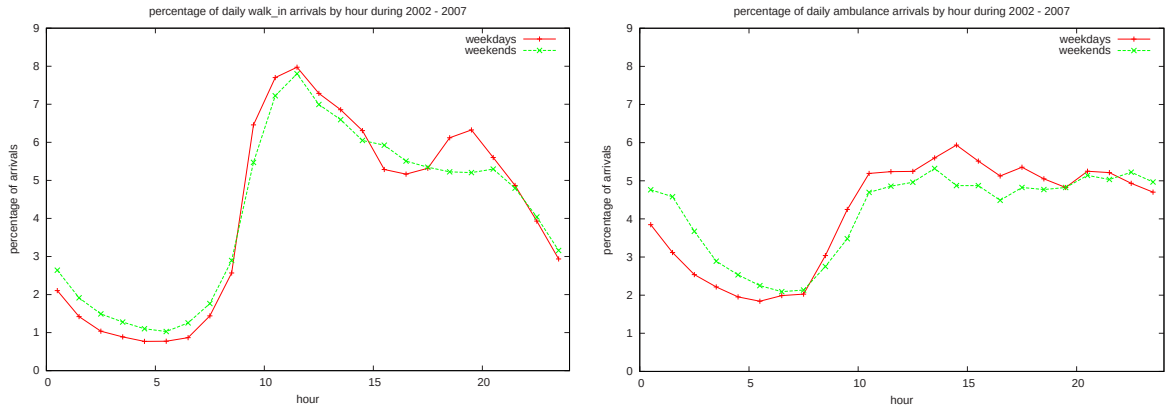


Figure 3.30: Plots of the percentage of walk-in (left) and ambulance (right) arrivals by hour over weekdays and weekends for 2002-2007.

Fig. 3.30 clearly shows that there are more arrivals during the early hours for both walk-in (between 00 and 07 hours) and ambulance (between 00 and 05 hours) arrivals during weekends than on weekdays. We can also see that walk-in arrivals are at a peak between 09 and 14 hours on both weekdays and weekends, followed by a smaller peak between 19 and 21 hours on weekdays but not weekends. Ambulance arrivals are at a peak and fairly constant between 11 and 23 hours on weekdays, and 11 to 01 hours on weekends.

3.8 The Impact of Weather Factors on Patient Arrivals

As described in Section 2.7.1, previous studies of emergency patient arrivals have found a link between weather-related factors such as temperature and rainfall, and the number of emergency service arrivals [37, 26, 60]. For our case study department we similarly

investigated if there was any relationship between a number of weather factors and the number of patient arrivals. First, we obtained five years of daily weather data (from January 2000 to March 2005) from the Met office for the closest weather station to our case study hospital. We then proceeded to fit a number of linear regression models of the maximum, minimum and average daily temperatures, wind chill and daily rainfall to the number of daily patient arrivals. However, we were unable to find any significant relationship between the any of these weather-related parameters and the number of arrivals to our case study A&E department. This may be due in part to the lack of extreme weather conditions here in the UK – unlike the countries of the case studies mentioned – which can result in a rise in the number of people requiring emergency medical care.

3.9 Conclusion

We have shown that using time series models we can characterise walk-in arrivals effectively and that our one to six day ahead forecasts have good predictive power. However, we had less success with our ambulance arrivals models. For one up to six days ahead predictions the structural time series (ST) model forecasts – for both walk-in and ambulance arrivals – perform better than the corresponding auto-regressive (AR) model predictions in terms of higher correlation with the observed data, generally lower root mean square error (RMSE) and narrower 95% confidence intervals. We have also shown that the one week ahead predictions from the auto-regressive and structural time series models perform better for walk-in arrivals than the rolling average (RA) predictions – currently used in our case study department – in terms of higher correlation with the "unseen" data. For ambulance arrival forecasts the ST model one week ahead predictions have a greater correlation with the observed arrivals than the RA model predictions which in turn have a higher r value than the corresponding AR model one week ahead forecasts.

The poor performance of the ambulance arrival time series model forecasts may be because the ambulance arrivals do not exhibit very strong periodicies or other regularity (cf. Fig 3.2). Thus the ambulance arrival stream might not be appropriate for this

method of time series analysis. We investigated characterising ambulance arrivals by a random process, fitting both a Poisson process and a linear non-homogenous Poisson process to the ambulance arrivals, but we found that our "training" data fails the corresponding goodness of fit tests. Despite being only able to characterise and forecast walk-in arrivals effectively, these model forecasts will still be of value to hospital managers as walk-in arrivals will account for the majority of arrivals into an A&E department. These methods may also be useful in characterising and forecasting other forms of hospital arrivals e.g. emergency hospital admissions [59] and non-emergency hospital department arrivals.

In Section 3.7 we showed that the hourly arrival patterns differ for ambulance and walk-in arrivals, but remain constant for each day of the week during each year of our data within each arrival stream. The hourly arrival patterns are similar during weekdays and during weekends for both arrival streams and may be combined together. This hourly breakdown of daily arrivals will be useful to hospital managers when deciding the staffing and resource levels required throughout the day as well as designing workshift and handover patterns that coincide with less busy periods of the day. For instance we have shown that there are proportionally more arrivals in the early hours during weekends for both walk-in and ambulance arrivals than during weekdays. This indicates more staff may be needed during the night on weekends, which currently is not the case in our case study department.

Finally, a study of the impact of weather-related variables on patient arrivals to our case study department found no significant relationships, possibly because of a lack of extreme weather conditions.

Chapter 4

Patient Flow Model

4.1 Introduction

In many complex processing systems with limited resources, fast response times are demanded, but are seldom delivered. Such requirements typically relate not only to mean response times, but also to variability of response times. This is an especially serious problem in healthcare systems providing critical patient care. In particular A&E departments in England are subject to a government set response time target, whereby 98% of patients should spend 4 hours or less in an A&E department from arrival to admission, transfer or discharge. There is therefore a need to develop appropriate performance models of A&E departments in order to assess their ability to meet these response time targets under various resource allocations and patient treatment schemes without having to disrupt the actual system.

As discussed in Section 2.7.2 there have been many models and simulations of A&E departments. There are three main drawbacks of these existing efforts. Firstly, they tend to be very high-level models, so they give a general overview of the system as a whole but are unable to provide any insights at the individual resource level. Secondly, performance measures are often limited to mean values of response times and utilisations; however, more sophisticated measures such as the higher moments and densities of response times can help to understand the variability in a system, and how it should be structured in order to meet response time quantile targets. Finally, these models

and simulations are often parameterised using small quantities of data and frequently either remain unvalidated or are validated against very little actual patient data.

This chapter describes our work on a hierarchical multiclass Markovian queueing network model of patient flow in our case study A&E department. Using patient timing data to help parameterise the model, we solve for moments and probability density functions of patient response time and associated resource utilisations using a discrete-event simulation. We experiment with different patient handling priority schemes and compare the resulting response time moments and densities with real data. We also implement various workload and resource availability scenarios and investigate the subsequent impact on system performance.

The remainder of this chapter is arranged as follows. Section 4.2 describes the patient flow diagrams upon which our queueing network model is based. In Section 4.3 we present the derived multiclass queueing network of patient flow. Section 4.4 describes methods by which we can extract performance measures from our queueing network model. Section 4.5 gives details on the implemented discrete-event simulation. Section 4.6 presents the mean, standard deviation and service time densities of actual patient service times. Section 4.7 presents the simulation results and comparison with actual data for class-based patient priority schemes, with Section 4.8 presenting the corresponding results for time-based priority schemes. Section 4.9 investigates model extensions to take into account impact of the four hour waiting time target. Section 4.10 presents a number of workload and resource/staffing scenarios and the corresponding simulation results. Section 4.11 concludes.

4.2 Preliminaries

Working closely with an A&E consultant at our case study department, we created detailed patient flow diagrams for the following three types of patient arrivals:

- **Self-referred arrivals** - patients that come of their own accord and not using hospital transport (see Appendix C.1),

- **GP-referred arrivals** - patients that come into A&E after being referred by a General Practitioner (see Appendix C.2), and
- **Ambulance arrivals** - patients that arrive via ambulance (see Appendix C.3).

Appendix C.1 additionally illustrates the patient pathways for minors patients – that is, patients with minor illnesses or injuries, the majority of which are self-referred arrivals, while Appendix C.3 additionally shows the patient pathways for majors patients – that is, patients with major illnesses or injuries, the majority of which arrive by ambulance.

4.3 Queueing Network Model

Figs. 4.2 and 4.3 show the simplified multiclass queueing network model of patient flow we have developed from the flow diagrams described in the previous section. Patient timing data from the financial year 2004 to 2005 (1 April 2004 to 31 March 2005) was used to calculate the average arrival rates and routing probabilities. Other parameters not obtainable from the data e.g. estimates of the average service rates, are given by an A&E consultant.

The model takes the form of a hierarchical network of $M/M/m$ queues. Fig. 4.2 shows top-level patient routing with various aggregated servers; their corresponding lower-level expansions are presented in Fig. 4.3. The top-level model has three patient arrival types: walk-in arrivals (whereby patients arrive under their own transport; this includes self-referred and GP-referred arrivals) and two types of ambulance arrivals (whereby the patient arrives by ambulance). Once in the department each patient is categorized as one of four customer classes: patients with minor illnesses or injuries (minors – class 1 in our model), patients with major illnesses or injuries (majors – class 2), patients requiring resuscitation (class 3) and patients that have yet to be classified (assessment – class 4). Customers can change class as they proceed through the system.

4.3.1 Notation

The five different types of nodes used in our model are shown in Fig. 4.1, with the role of each node described in more detail below:

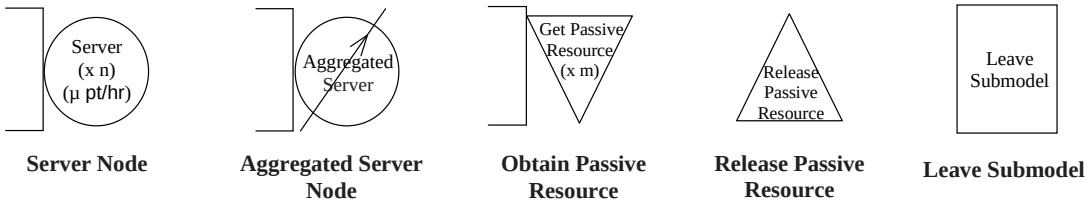


Figure 4.1: Queueing network model notation.

- **Server Node** - service node with n servers, each with mean service rate μ patients per hour (pt/hr). If all n servers are busy, patients queue for service.

- **Aggregated Server Node** - a single node representing the aggregation of a submodel in the top-level model. These are indicated in Fig. 4.2 and are shown in more detail in the lower-level expansion (Fig. 4.3).

- **Obtain Passive Resource** - node at which a patient obtains one of m passive resources (a resource required by a patient before they can progress along a treatment path). Once obtained, the patient retains the resource until released. If all m resources are taken, patients will queue for the resource to become available. There is a configurable added deterministic delay (1 minute in our models) in acquiring a passive resource to account for the time it takes for a patient to move to the resource.

- **Release Passive Resource** - node at which a passive resource is (instantaneously) released, at which point it becomes available to other patients.

- **Leave Submodel** - node at which a patient leaves the aggregated server, moving from the lower-level submodel to the top-level model.

4.3.2 Passive Resources

In many cases a patient needs to obtain a (passive) resource before they can progress along a treatment path. An example is the nurse assessment rooms (of which there are 5 in our A&E department). A patient must wait for one to become free before

entering the room for assessment by a nurse. Once the assessment has been completed, the patient leaves the room, freeing it up for the next patient. Other passive resources include minors cubicles (of which there are 9), majors bays (of which there are 25) and resuscitation beds (of which there are 4).

4.3.3 Walk-in Patients

As shown in the top-level model, the majority of patient arrivals are walk-in patients who enter via the A&E waiting room where they are registered at reception. The receptionists then route each patient into one of three queues: patients with a clear case of minor injury are placed in the minors queue (see AEU Submodel); patients with a clear case of a serious illness or serious injury are sent to the majors queue (see AEU Submodel); all others (including all suspected cases of minor illness) are sent for nurse assessment (see Assess Submodel).

Minors Queue

Patients in the minors queue must first wait for a minors cubicle to become free; the patient then waits there for a minors practitioner (either a minors doctor or a nurse practitioner) to see them. The minors practitioner can decide to:

- Perform investigative tests and/or scans such as blood tests and x-rays, or
- Ask for a specialist opinion, or
- Treat (if necessary) and discharge the patient (to home, their GP or to the pharmacy to pick up medication), or
- Send the patient to be admitted to a (surgical) ward, or to the Medical Assessment Unit (MAU) which assesses the need for medical admissions.

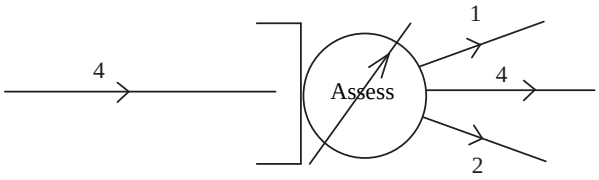
Majors Queue

Patients in the majors queue wait for a bed in a majors bay to become free; once there, a nurse may arrange for a number of tests (e.g. vitals, blood tests, x-ray) so that

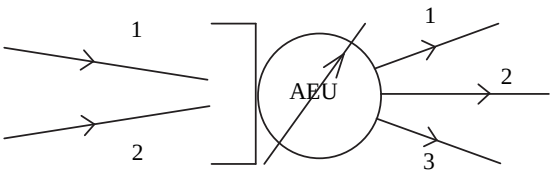
Patient Classes

1. Minors
2. Majors
3. Resuscitation
4. Assessment

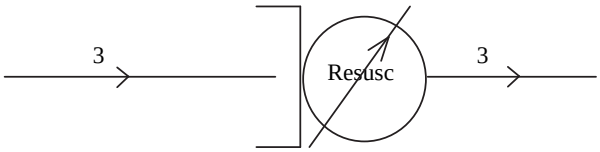
Aggregated Server Assess (Assessment Area)



Aggregated Server AEU (Main A&E Unit)



Aggregated Server Resusc (Resuscitation Unit)



Top Level Model

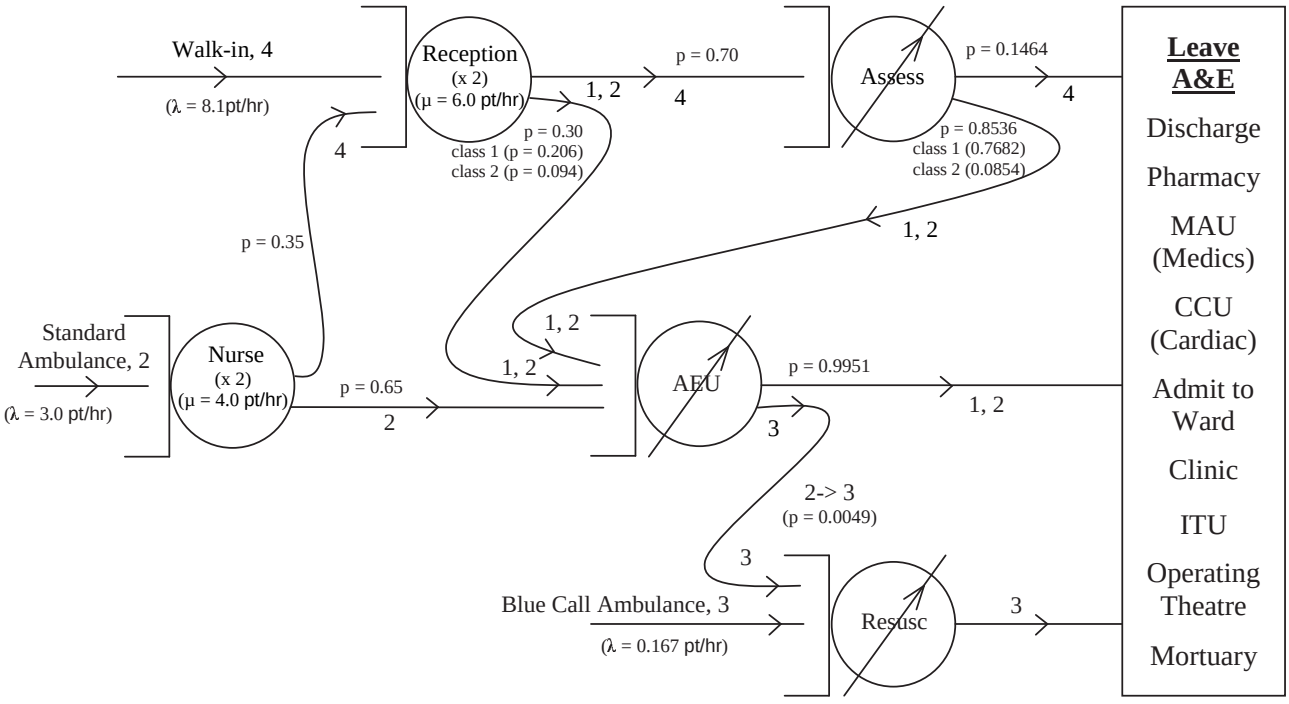
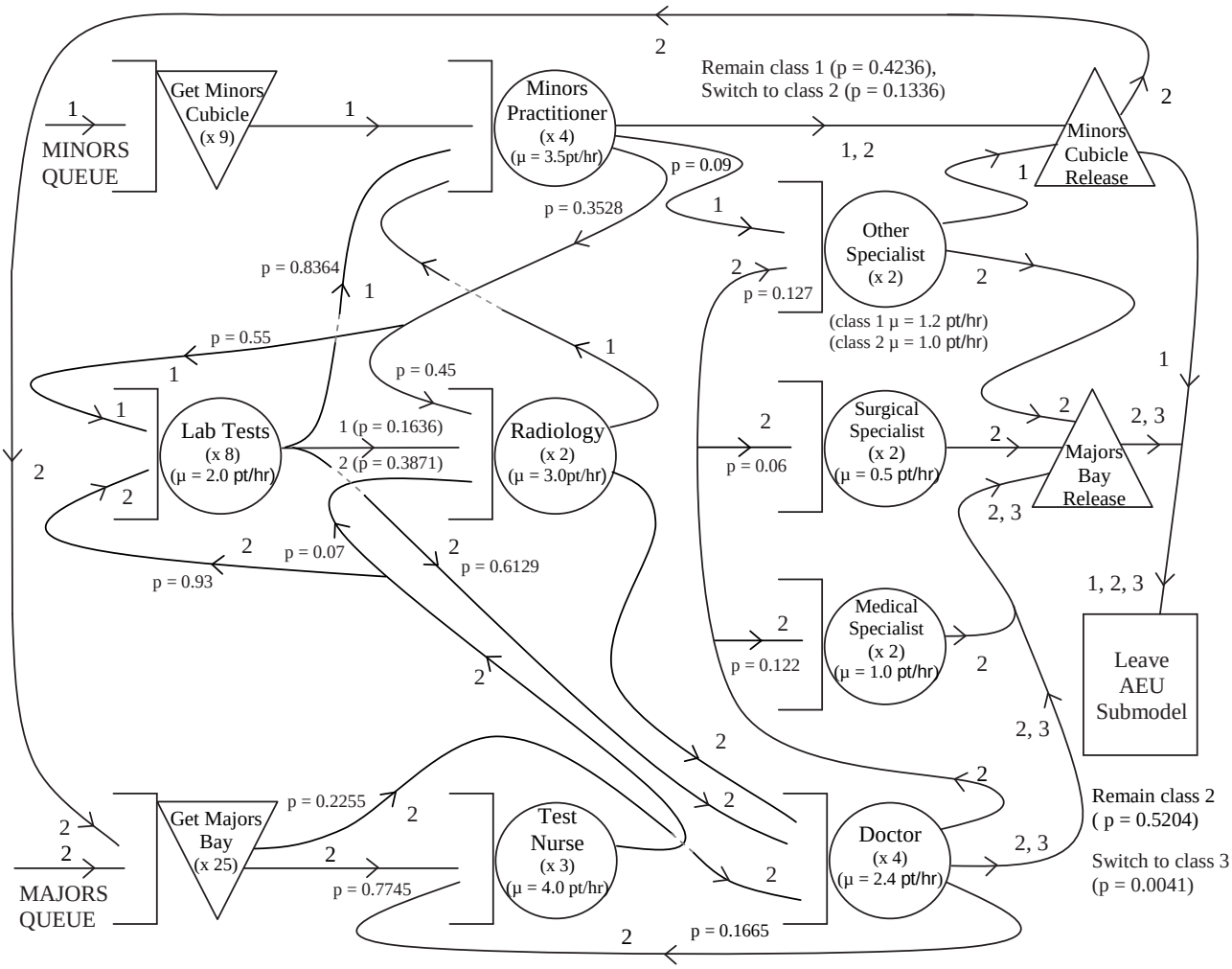
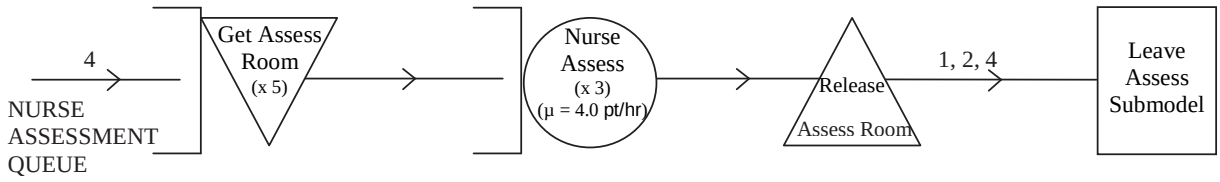


Figure 4.2: Top-level of queueing network model of patient flow.

AEU Submodel



Assess Submodel



Resusc Submodel

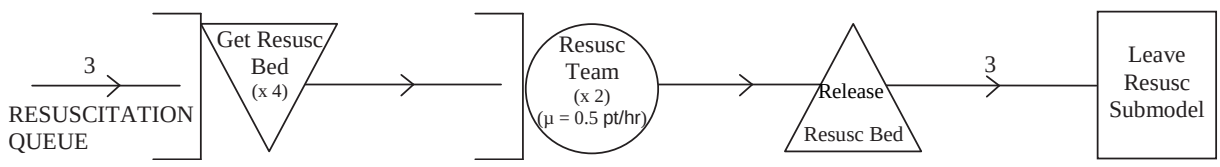


Figure 4.3: Lower-levels of queueing network model of patient flow.

essential information is ready for when a doctor assesses the patient. Tests for both majors and the minors are processed in the same laboratory and radiology facilities. When the doctor has assessed the patient, (s)he may require a specialist opinion, request more tests, or send the patient out of A&E (possibly after treatment) via the routes mentioned above for the minors queue, or, if the patient has a cardiac complaint, send them to the Coronary Care Unit (CCU). Very rarely a patient may suffer a sudden rapid deterioration, in which case the patient is transferred to a resuscitation bay and is attended to by the resuscitation team.

Nurse Assessment

Patients in the nurse assessment queue wait for an assessment room to become available; they then wait there for a nurse to assess the severity of their illness or injury. The nurse can send the patient either to the minors queue, the majors queue or discharge them out of A&E to a specialist clinic, ward, GP etc.

Specialists

A specialist's opinion may be required by a minors practitioner or majors doctor. Minors patients are only referred to "other" specialists which encompass Obstetrics and Gynaecology, ENT (ear, nose and throat) and Orthopaedics. Majors patients may be seen by medical, surgical and other specialists. After assessment, patients are discharged from A&E, either by being sent home, to a clinic for a more thorough investigation, to a ward for admission or to the MAU.

4.3.4 Ambulance Arrivals

As shown in the top-level model, there are two forms of ambulance arrivals: standard ambulance arrivals, which make up the majority of the cases and blue call ambulance arrivals, who require immediate medical attention.

Standard Ambulance Arrivals

These patients (whom we shall refer to from now on as simply ambulance arrivals) are handed over to a nurse from the ambulance. The nurse assesses the patient, and sends them either to reception to be registered (where the patient is routed as for walk-in patients) or straight to a majors bay (where the patient joins the majors queue).

Blue Call Ambulance Arrivals

Blue call ambulance arrivals (whom we shall refer to from now on as blue call arrivals) are very seriously ill or injured patients that require urgent medical attention. Such patients are assigned a resuscitation bay and are attended to by a resuscitation team. Once stable, the patient leaves A&E, being sent either to an operating theatre, to the Intensive Treatment Unit (ITU), to the CCU or to a ward. Patients who cannot be resuscitated are sent to the mortuary.

4.3.5 Complexities not Modelled

There are many complexities not incorporated into our model that would be present in a real life A&E department. These include:

Transient System Parameters In an actual A&E department there are different arrival rates throughout the day (as demonstrated in Chapter 3); staffing and resource levels also vary through the day. However, for simplicity, we use fixed-rate Poisson arrivals and average staffing and resource numbers in our model.

Distributional Assumptions We assume Poisson arrivals and exponential service times. These distributional assumptions are based on mean arrival rates for each arrival type and estimates of mean service times, and so might not reflect the true variability of patient arrivals and staff and resource service time distributions.

Bed Blocking Once it has been decided that a patient should be admitted to a ward or sent to the MAU, if there is no bed/room available for them then the patient stays

in A&E, waiting until a bed becomes available. This a major cause of four hour waiting time breaches.

Hospital Transport Blocking When a patient is discharged, if they require an ambulance to send them home, they stay in A&E until there is one available. Until the patient leaves A&E (s)he is still subject to the four hour waiting time target.

Porters Porters are required to transport patients to various scans and x-rays if the patient is not able to walk. They also transport patients to the different wards or to the MAU for admission.

Parallel Tests and Scans Patients that require both laboratory tests and radiology scans in our model have the tests completed first before going to radiology. In an actual A&E unit, blood and urine etc. samples are first taken and while they are being processed at the laboratory, the patient is sometimes taken to radiology – so the two can potentially occur with some overlapping and not sequentially as modelled.

Treatment We have incorporated the treatment time of a patient into the time spent being seen by either the doctor or minors practitioner. Depending on the treatment this may or may not be the case in an actual A&E unit. Often a doctor treats a patient, leaves them to see to another patient before returning later to see if the treatment has had any effect.

Staff Resources For simplicity in our model we have limited the tasks performed by some members of the staff; for instance we have assigned nurses to only perform the specific task of patient assessment. In an actual A&E unit nurses are trained to perform all required A&E nursing tasks and so act as a pool of resources with the nurses performing whichever tasks are more urgent first. Similarly, doctors working in the minors area (minors practitioners) may help to treat patients in the majors area when it is busy and vice versa.

GP Referrals Some patients arrive in A&E after being referred by a GP for specialist treatment. In some cases, the patients are sent straight to the specialists; for others the patient is first assessed in A&E with perhaps tests being run before being transferred to a specialist.

Specialists Obviously there are many different types of other specialists available in a hospital and not just the Obstetrics and Gynaecology, ENT and Orthopaedic specialists represented in our model.

4.4 Model Solution

There are two main approaches to obtaining performance measures from a model: *simulation* and *analytical* methods. Simulation is used to model systems at arbitrary levels of detail, producing inexact results bounded by confidence intervals. However, there is a high cost and effort involved in constructing accurate models and the length of execution time required to produce reliable results can be very long. Analytical models, on the other hand, make use of formal, abstract models from which exact results can be obtained by generating and solving a set of equations derived from the model.

Using our queueing network model we explored the possibility of solving a closed system analytically using the numerical Laplace transform inversion based method described in Section 2.6, to determine moments and densities of patient service times. However, a major difficulty associated with this kind of modelling is the well known *state space explosion problem* [14, 97], whereby the state space that emerges from complex models becomes intractably large, making it impossible to explore the entire state space within reasonable time using realistic computing memory and power. It is this problem that we encountered when trying to solve our system analytically – our model has over 16 million states in a closed system with only 8 patients.

We therefore implemented a discrete-event simulation of the queueing network model[1]. Discrete-event simulation involves the modelling of a system as it evolves at a dis-

[1]In the next chapter, we return to seek an efficient analytical approximation that avoids the state-space explosion problem.

crete set of points in time. These points in time are the ones at which an event (an instantaneous occurrence that changes the state of the system) occurs. Two integral components of any discrete-event simulation are the *simulation clock* and the *future event list*. The simulation clock is a variable which gives the current value of time and the future event list is a list of the times of occurrences of (pre-determined) future events. This means that the instant an activity begins, its duration is computed (usually this involves drawing a sample from a statistical distribution) and the end event activity (the event that occurs once the current event ends), together with its event time, is placed on the future event list.

When the simulation begins a run, the simulation clock (which is initialised to zero) advances to the time of the first of the events on the future event list and updates the state of the system accordingly (including scheduling new events in the future event list); the clock then advances to the time of the next event. This process is repeated until the system reaches some pre-determined stopping condition. For more details, a thorough discussion of discrete-event simulation can be found in numerous works, for example [12, 69, 81].

4.5 A&E Simulation

We implement the queueing network model via a discrete-event simulation written in Java, using and building on the JINQS Java queuing network simulation library [43]. Using this simulation we obtain various performance measures including moments of patient service time, resource utilisations and patient service time densities for each of the three patient arrival streams (walk-in, ambulance and blue call arrivals). We investigate the performance of the system under different class-based and time-based patient priorities schemes as well as various workload and resource scenarios.

The simulation results presented in the following sections are the average of ten runs. Each run includes a transient period during which 2 000 000 patients move through the system (and during which passage time statistics are not collected), followed by a measurement period which lasts long enough to observe 10 000 passages of blue call arrivals through the system; in this period approximately 485 000 passages of walk-in arrivals

and 180 000 passages of ambulance arrivals are also observed. Each simulation run takes between 20 and 60 minutes wall clock time, depending on the workload/resource parameters, priority scheme implemented and PC cluster workstation used.

4.6 Actual Patient Service Times

Before we present results from our simulation, we first present in Table 4.1 the mean and standard deviation of patient service time and in Fig. 4.4 the service time densities for the three types of patient arrival (i.e. walk-in, ambulance and blue call arrivals) as actually observed in our case study A&E department. Figures are reported over three annual reporting periods (2002/2003, 2003/2004 and 2004/2005), where each reporting period begins on 1 April and ends on 31 March the following year (coinciding with the hospital's financial year). One can readily observe the effect of the introduction and subsequent tightening of patient service time targets – note the peaks corresponding to the four hour target, most evident in the 2004/2005 densities when the four hour target increased from 90% of patients (from March 2003) to 98% of patients (from January 2005) seen within four hours.

	walk-in arrivals		**ambulance arrivals**		**blue call arrivals**	
year	mean	std dev	mean	std dev	mean	std dev
2002/2003	3.22	3.61	5.69	4.84	4.18	5.19
2003/2004	2.46	2.23	4.22	3.12	2.43	2.20
2004/2005	2.04	1.59	3.14	2.12	2.09	1.84

Table 4.1: Observed mean and standard deviation (std dev) of service times (in hours) for different classes of arriving patient.

Our simulation results will be compared with these observed results as we investigate which of the priority schemes described in the upcoming sections returns results that are most consistent with each of the above annual reporting periods.

4.7 Patient Class-based Priority Schemes

We first investigated the performance of the system under various patient class-based priority schemes. The three different patient class-based priority schemes analysed are:

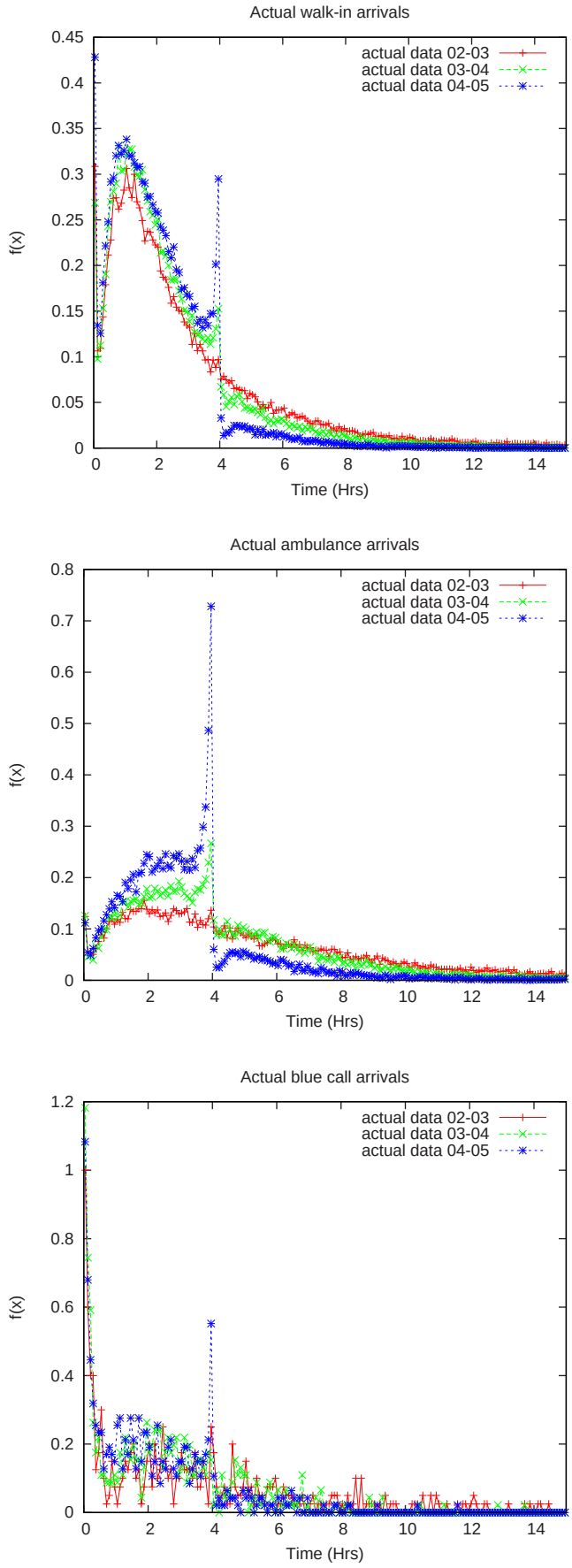


Figure 4.4: Actual service time densities for walk-in (top), ambulance (middle) and blue call (bottom) arrivals for the years 2002 to 2005.

- **Majors Priority** in which majors patients are given priority at the shared resources (lab tests, radiology and other specialist),
- **Minors Priority** in which minors patients are given priority at the shared resources, and
- **No Priority** in which First In First Out (FIFO) queues are implemented at each node.

We incorporated these three class-based priority schemes as we hypothesise that a majors priority scheme most closely represents the priority scheme in place before the introduction of the 4 hour patient response time target, whereby more seriously ill/injured patients are seen/treated first. After the introduction of the target, we believe the system shifted to a scheme more like that of a minors priority system where the emphasis is on processing patients as quickly as possible – since the majority of patients in an A&E system are minors patients who are generally easier and quicker to treat, we assume these are given priority. Finally, we also investigate a no priority system whereby patients are processed on a First In First Out basis at each node.

4.7.1 Numerical Results

Tables 4.2, 4.3 and 4.4 show the mean and standard deviation of patient service time for various types of patient arrival (walk-in, ambulance and blue call arrivals) under different class-based priority schemes and the corresponding 95% confidence interval widths, as calculated using our discrete-event simulation. Table 4.5 shows the utilisations for a selection of staff and resources under the different priority schemes.

From Table 4.2 it can be seen how giving priority to the majors class seriously degrades the waiting time of the walk-in patients (in terms of both mean and standard deviation), which are predominantly minors. By contrast it might appear from Table 4.3 that seriously injured or ill patients arriving by ambulance actually benefit from a minors priority scheme. In fact both ambulance and walk in arrivals under minors priority are seemingly treated quicker than even a no priority system (see Table 4.4).

majors priority	walk-in arrivals	ambulance arrivals	blue call arrivals
mean	4.4391 (± 0.0492)	3.1934 (± 0.0176)	2.0929 (± 0.0051)
std dev	4.6034 (± 0.1116)	3.2386 (± 0.0632)	2.0430 (± 0.0080)

Table 4.2: Mean and standard deviation (std dev) of service times (in hours) and the corresponding 95% confidence interval widths (in brackets) for walk-in, ambulance and blue call arrivals under the majors priority scheme as calculated via simulation.

minors priority	walk-in arrivals	ambulance arrivals	blue call arrivals
mean	2.3926 (± 0.0029)	2.7096 (± 0.0028)	2.0868 (± 0.0059)
std dev	2.0271 (± 0.0032)	2.2101 (± 0.0046)	2.0356 (± 0.0061)

Table 4.3: Mean and standard deviation (std dev) of service times (in hours) and the corresponding 95% confidence interval widths (in brackets) for walk-in, ambulance and blue call arrivals under the minors priority scheme as calculated via simulation.

no priority	walk-in arrivals	ambulance arrivals	blue call arrivals
mean	3.0441 (± 0.0171)	2.7952 (± 0.0069)	2.0871 (± 0.0037)
std dev	2.7149 (± 0.0391)	2.2145 (± 0.0197)	2.0425 (± 0.0053)

Table 4.4: Mean and standard deviation (std dev) of service times (in hours) and the corresponding 95% confidence intervals widths (in brackets) for walk-in, ambulance and blue call arrivals under the no priority scheme as calculated via simulation.

We compare these simulations results to the actual observed means (shown previously in Table 4.1). This is achieved by first writing the simulated majors priority scheme walk-in and ambulance arrival means as a vector $\mathbf{x_1}$ of two components. The 2002/2003 actual walk-in and ambulance arrivals mean service times are also written as a vector $\mathbf{y_1}$. Similarly the corresponding simulated minors priority and no priority scheme results are written as vectors $\mathbf{x_2}$ and $\mathbf{x_3}$ respectively, while the 2003/2004 and 2004/2005 actual results are written as vectors $\mathbf{y_2}$ and $\mathbf{y_3}$ respectively. We then compare each of the $\mathbf{x_i}$ vectors ($i = 1, 2, 3$) representing the simulated means against each of the $\mathbf{y_i}$ vectors representing each year of actual means, by calculating the Euclidian distance (or two-norm) between each[2]. We find that for the 2002 to 2003 actual data our majors priority scheme gives the overall best match. For the 2004 to 2005 actual data (after the introduction of the four hour waiting time target) we find that the minors priority scheme gives the best overall match. For blue call arrivals – which form an almost independent subsystem – we have the same mean and standard deviation for all priority schemes and we find that our simulation mean is very close to the actual

[2]where the Euclidian distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ each of length n denoted by $||\mathbf{x} - \mathbf{y}||_2$ is defined as: $||\mathbf{x} - \mathbf{y}||_2 = \sqrt{((x_1 - y_1)^2 + \ldots + (x_n + y_n)^2)}$

observed mean for the year 2004 to 2005.

Table 4.5 shows the utilisations for the minors cubicles, majors bays, minors practitioners, doctor and the shared resources (other specialists, radiology and labs) under the three different patient class priorities. The minors cubicles, minors practitioners and other specialists are the most highly utilised resources in the system. We can see that the different priority schemes only affect the utilisations of the passive resources (minors cubicles and majors bays) while the utilisations for staff (i.e. minors practitioners, doctors, other specialist, radiology and labs) are insensitive to the priority scheme used. As expected, majors bays are less highly utilised under majors priority, while minors cubicles are less highly utilised under minors priority, with the no priority system having intermediate utilisations for both resources. The reason that priority schemes do not impact the utilisations of the staff (in particular the minors practitioners and doctors) is due to the workload for staff remaining the same regardless of the priority scheme used. However, the priority scheme will have an impact on the patient's length of stay, depending on his/her class, and hence will affect the length of time for which a passive resource is occupied and so affect the utilisation of that resource.

	utilisation		
resource	**majors priority**	**minors priority**	**no priority**
minors cubicle	0.8972 (± 0.0006)	0.7751 (± 0.0004)	0.8448 (± 0.0005)
majors bay	0.3932 (± 0.0002)	0.4581 (± 0.0006)	0.4188 (± 0.0002)
minors practitioner	0.7507 (± 0.0003)	0.7512 (± 0.0003)	0.7511 (± 0.0003)
doctor	0.5951 (± 0.0002)	0.5956 (± 0.0003)	0.5952 (± 0.0003)
other specialist	0.7563 (± 0.0003)	0.7571 (± 0.0004)	0.7571 (± 0.0005)
radiology	0.6660 (± 0.0003)	0.6667 (± 0.0004)	0.6664 (± 0.0004)
labs	0.3971 (± 0.0002)	0.3974 (± 0.0003)	0.3970 (± 0.0002)

Table 4.5: Utilisations of a selection of staff and resources and the corresponding 95% confidence intervals widths (in brackets) under the different patient class-based priority schemes.

4.7.2 Densities of Patient Service Time

Figs. 4.5, 4.6 and 4.7 show the simulated vs. actual patient service time densities for walk-in, ambulance and blue call arrivals respectively; again note the peaks in the 2004/2005 actual service time densities corresponding to the four hour target.

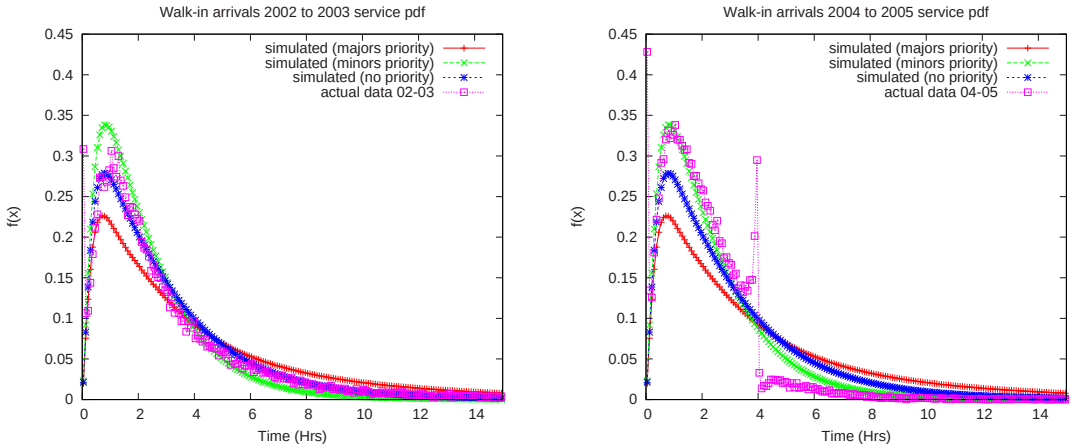


Figure 4.5: Actual and simulated service time density for walk-in arrivals using 2002/2003 data (left) and 2004/2005 data (right).

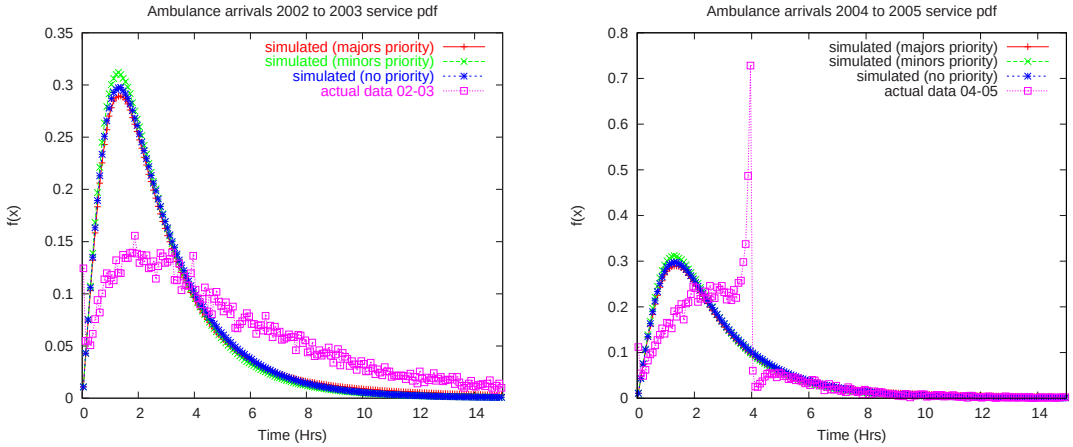


Figure 4.6: Actual and simulated service time density for ambulance arrivals using 2002/2003 data (left) and 2004/2005 data (right).

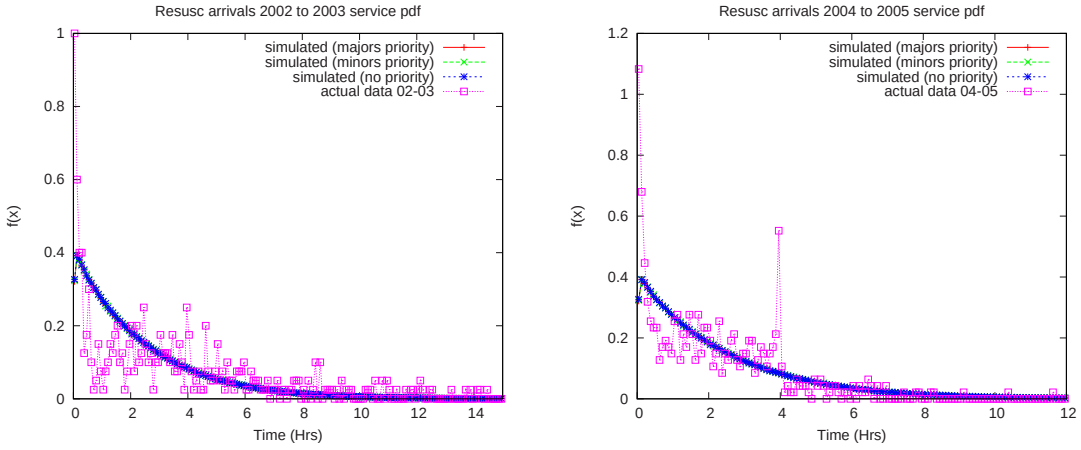


Figure 4.7: Actual and simulated service time density for blue call arrivals using 2002/2003 data (left) and 2004/2005 data (right).

For the walk-in arrivals we can see that the simulation density under no priority is a good fit to the 2002/2003 actual service time density, while the minors priority system provides a good fit (except for the area immediately before and after the peak corresponding to four hours) to the 2004/2005 actual patient service time density. For ambulance arrivals, we can see that the simulation under the three priority schemes give similar service time densities, with none of them adequately fitting the 2002/2003 actual service time density. The fit to the 2004/2005 density is better, but it is difficult to say which priority scheme provides a better fit as the three densities are so similar. For the blue call arrivals the priority schemes have little impact and the simulated densities are the same for each.

4.8 Time-based Priorities

We next investigate the performance of the system under various time-based priority schemes. The two priority schemes are based on the colour coding system used in our case study department, which helps to indicate to staff which patients need to be seen most urgently. Under this system patient details are shown on computer screens in:

- **Red** - if waiting 3 hours and over
- **Yellow** - if waiting 2 to 3 hours
- **Green** - if waiting 1 to 2 hours
- **Blue** - if waiting under 1 hour

The two different patient time-based priority schemes implemented are as follows:

- **Arrival First Priority** in which patients who have been in the department the longest (i.e. have the earliest arrival time) are given priority at the shared resources (lab tests, radiology and other specialist), and
- **Traffic Light Priority** in which patients are allocated higher and higher priority levels at the shared resources as they approach the 4 hour time limit, but are given less priority once the 4 hour mark has been breached. The priority levels are as follows (highest priority first):

- ⋆ **Level 0** - patient has spent between 3 and 4 hours in the department
- ⋆ **Level 1** - patient has spent over 4 hours in the department
- ⋆ **Level 2** - patient has spent between 2 and 3 hours in the department
- ⋆ **Level 3** - patient has spent between 1 and 2 hours in the department
- ⋆ **Level 4** - patient has spent under 1 hour in the department

We experimented with the arrival first priority scheme as we wish to see if treating patients who have been waiting the longest overall first will lead to an overall reduction in waiting time for both patient arrival types. We investigate the traffic light priority system because it mirrors the "rising panic" phenomenon that occurs in real A&E units whereby patients are subject to higher and higher priority treatment as they approach the four hour waiting time target. In this way, we hope to see similar peaks at 4 hours as seen in the observed patient service time densities. In the following sections we only show the results for the walk-in and ambulance arrivals since the blue call results remain the same as before.

4.8.1 Numerical Results

Tables 4.6 and 4.7 show the mean and standard deviation of patient service time for the various types of patient arrival (walk-in, ambulance and blue call arrivals) under the two time-based priority schemes and the corresponding 95% confidence interval widths, as calculated using our discrete-event simulation. Table 4.8 shows the utilisations under the different priority schemes of the minors cubicles, majors bays, minors practitioners, doctors and the shared resources.

arrival first priority	**walk-in arrivals**	**ambulance arrivals**
mean	2.8116 (± 0.0070)	2.7716 (± 0.0035)
std dev	2.2728 (± 0.0114)	2.0512 (± 0.0066)

Table 4.6: Mean and standard deviation (std dev) of service times (in hours) and the corresponding 95% confidence interval widths (in brackets) for walk-in, ambulance and blue call arrivals under the arrival first priority scheme as calculated via simulation.

From Tables 4.6 and 4.7 we can see that for both walk-in and ambulance arrivals the mean response time is lower under the arrival first priority scheme than under the

traffic light priority	walk-in arrivals	ambulance arrivals
mean	2.9884 (± 0.0156)	2.8027 (± 0.0062)
std dev	2.6476 (± 0.0298)	2.2089 (± 0.0153)

Table 4.7: Mean and standard deviation (std dev) of service times (in hours) and the corresponding 95% confidence interval widths (in brackets) for walk-in, ambulance and blue call arrivals under the traffic light priority scheme as calculated via simulation.

traffic light priority scheme. This is to be expected as under traffic light priority those who have waited for over 4 hours have lower priority that those waiting between 3 to 4 hours and so will give higher overall response times than for the arrival first priority scheme. However, both timed priority schemes give higher mean response times for both walk-in and ambulance arrivals than for the system under minors priority. When we compare these two priority scheme results to the actual observed means in terms of Euclidian distance, we find that the arrivals first and traffic light priority schemes return similar values. These two time-based priority schemes are not as good a match as the majors priority scheme with the 2002 to 2003 actual data nor is it as good a match as the minors priority scheme with 2004 to 2005 actual data. However, for the 2003/2004 data, the arrivals first priority scheme provides the best match.

	utilisation	
resource	arrival first priority	traffic light priority
minors cubicle	0.8396 (± 0.0007)	0.8437 (± 0.0007)
majors bay	0.4232 (± 0.0003)	0.4209 (±0.0004)
minors practitioner	0.7509 (± 0.0004)	0.7511 (± 0.0002)
doctor	0.5952 (± 0.0003)	0.5951 (± 0.0003)
other specialist	0.7568 (± 0.0005)	0.7571 (± 0.0008)
radiology	0.6663 (± 0.0004)	0.6664 (± 0.0002)
labs	0.3972 (± 0.0002)	0.3973 (± 0.0002)

Table 4.8: Utilisation of a selection of staff and resources and the corresponding 95% confidence interval widths (in brackets) under the different time-based priority schemes.

Examining the utilisations shown in Table 4.8 we can see that minors cubicles are more highly utilised under both time-based priorities than under the minors priority scheme (cf. Table 4.5). However the majors bay is less utilised for both than under minors priority, but still more highly utilised than when under both no priority and majors priority schemes. The relatively poor performance of the traffic light priority may be because even if patients are given priority at the shared resources, by the time most

patients are processed and leave the A&E unit, they have breached the 4 hour time limit (for instance, even if a patient is given priority by the other specialist, service would still take on average 60 minutes for a majors patient); this then has a knock on effect on those patients who are in the department for between 2 to 3 hours and are assigned two levels lower priority. Similarly for the arrival first priority scheme, by prioritising patients who are in the department the longest, those patients who could be processed faster are penalised while they wait for the higher priority patients to be processed, which has a detrimental effect on the overall mean service times. A priority system similar to the traffic light system whereby patients waiting between 2 and 3 hours in total are given higher priority may give better results in terms of mean response times.

4.8.2 Densities of Patient Service Time

Figs. 4.8 and 4.9 show the simulated vs. actual patient service time densities for walk-in and ambulance arrivals respectively under the arrival first priority and traffic light priority schemes. Fig. 4.10 shows the densities under the two time-based priority schemes together with the actual service time densities for both arrival types.

From Fig. 4.8 we can see that the both the arrival first and traffic light priorities are not as good a match with the 2004/2005 actual walk-in service time density as under the minors priority system (cf. Fig. 4.5). Notice the slight "bump" between 3 and 5 hours on the traffic light priority densities (on the right of Figs. 4.8 and 4.9) corresponding to the impact of giving patients approaching four hours total service time the highest priority level. This relatively small impact on the resulting density around four hours under the traffic light priority scheme indicates that starting to give highest priority to patients one hour before the four hour mark does not give as dramatic an impact in our simulations as seen in the actual system. This together with the plots of the actual 2004/2005 densities (cf. Fig. 4.4) suggests that patients are given highest priority status just before the four hour mark and are then subsequently dealt with much more rapidly than modelled (possible through the deployment of extra resources or routing of patients out of A&E). Fig 4.10 compares both time-based priority systems with the actual service densities, and shows that the arrival first and traffic light priorities give very similar looking densities.

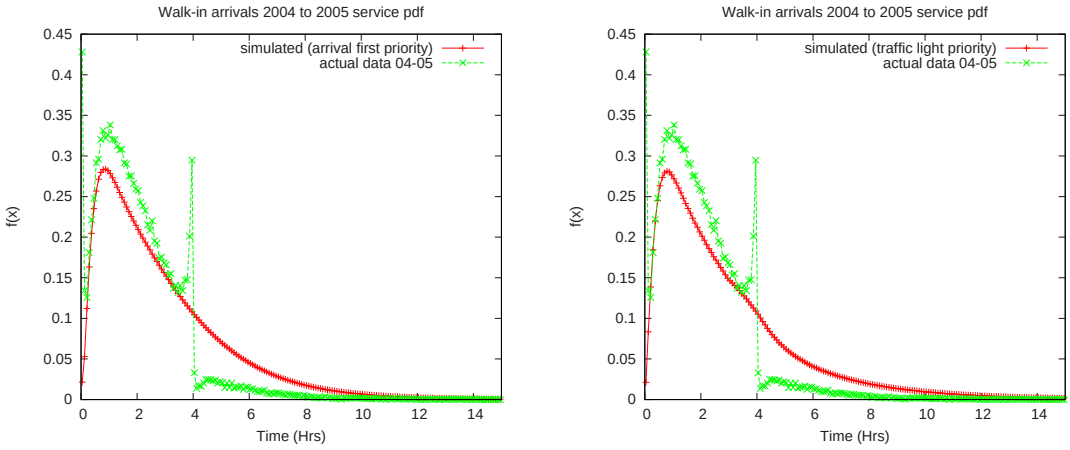


Figure 4.8: Actual and simulated service time density for walk-in arrivals under arrival first priority (left) and traffic light priority (right).

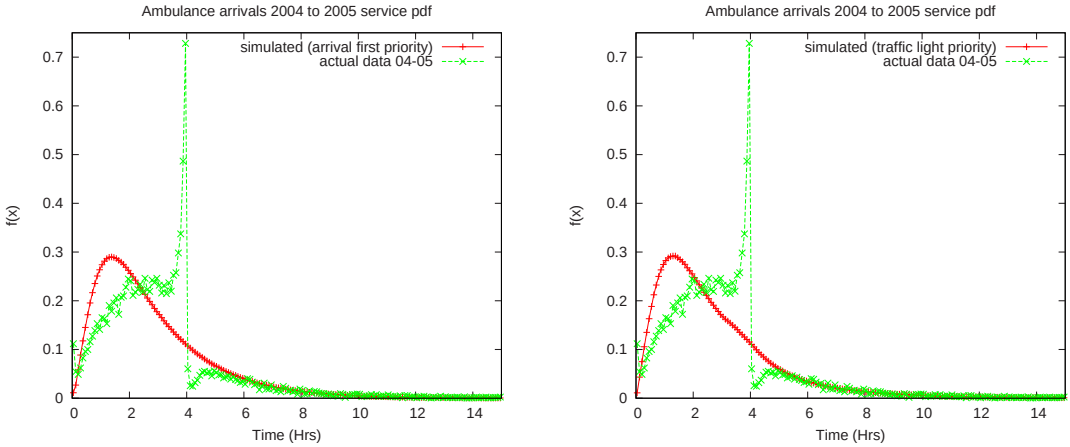


Figure 4.9: Actual and simulated service time density for ambulance arrivals under arrival first priority (left) and traffic light priority (right).

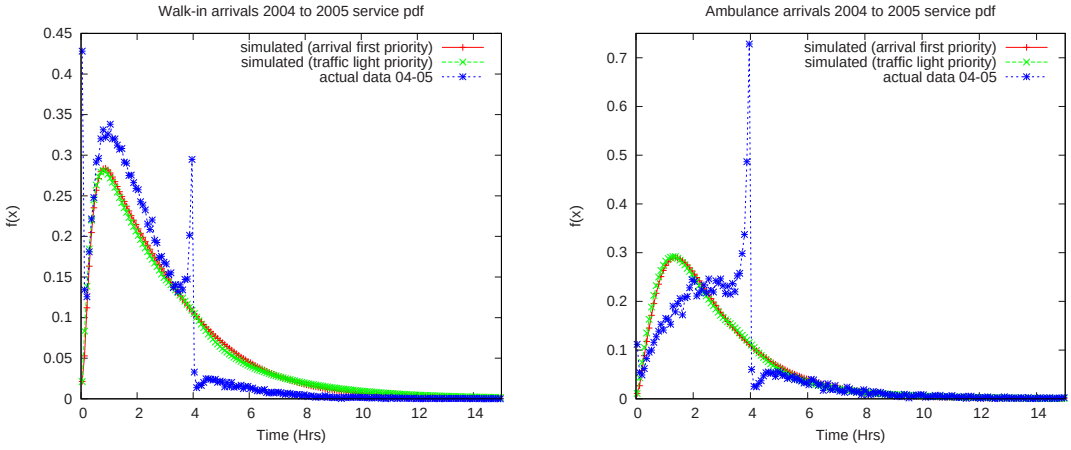


Figure 4.10: Actual and simulated service time density comparing the two time-based priority schemes for walk-in (left) and ambulance (right) arrivals.

4.9 Replicating the Impact of the Four Hour Target

We have seen that of all the priority schemes investigated, the minors priority scheme gives the overall closest match to the actual 2004/2005 system. The biggest discrepancy between the simulated system under minors priority and the actual system is the lack of a spike corresponding to four hours and the height of the simulated density tail immediately after this peak. We now look at ways of adapting the minors priority scheme to include the impact of this target. Having shown that using ten simulation runs for each system scenario provides narrow 95% confidence interval widths, we omit these results in the following sections.

4.9.1 Inserting a CDU Node

First we experiment with the insertion of a Clinical Decision Unit (CDU) which is used by A&E as a holding ward for patients if they are waiting for the return of test/scan results or a specialist opinion and are in danger of breaching the four hour service time target. In our case study department the CDU has 12 beds and an estimated mean holding time of 24 hours.

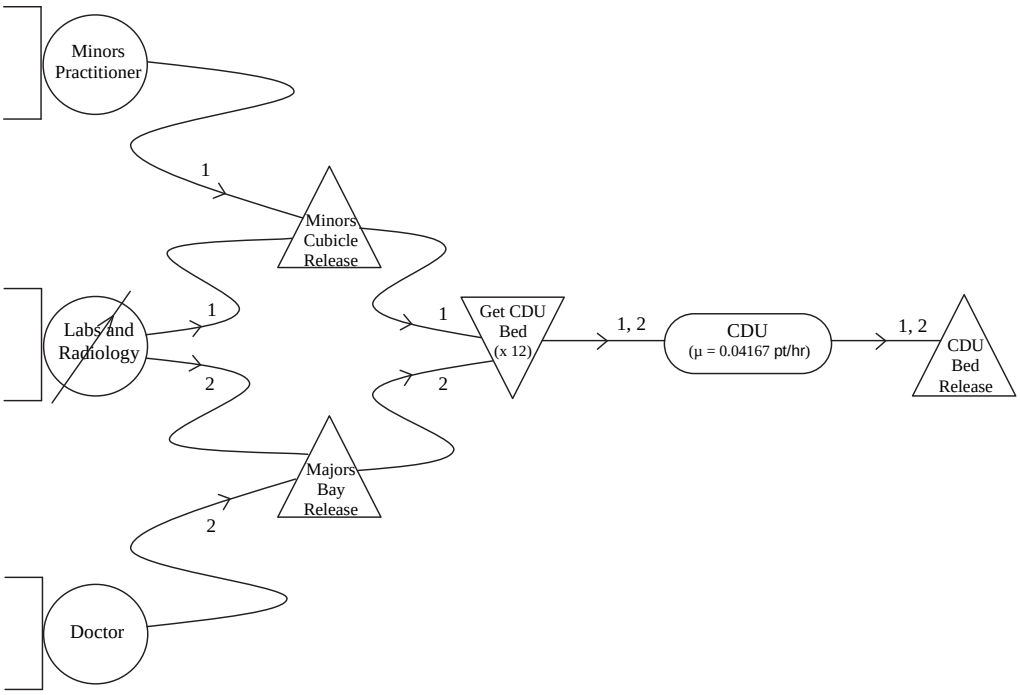


Figure 4.11: Position of the CDU node.

In Fig. 4.11 we show the placement of the CDU node. The node is placed after the laboratory tests and radiology nodes (shown aggregated together as one node) and after the minors practitioner and the doctor servers. This path is only taken if the patient has waited between 3.75 and 4 hours and there is a bed available in the CDU. On taking this path, the patient releases the passive resource currently held (either a minors cubicle or majors bay) and obtains a CDU bed (notice there is no queueing). Once they have a bed the patient is taken to have left the A&E department and the patient timing stops. The patient then holds the CDU bed for a time drawn from the exponential distribution with mean 24 hours before releasing it.

Table 4.9 compares the mean and standard deviation of patient service time in A&E of walk-in and arrivals under minors priority with and without the CDU node. The corresponding service time densities are shown in Fig 4.12.

	walk-in arrivals		**ambulance arrivals**	
priority scheme	mean	std dev	mean	std dev
with CDU node	2.2267	1.8068	2.5626	1.9848
without CDU node	2.3926	2.0271	2.7096	2.2101

Table 4.9: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under minors priority with and without a CDU node.

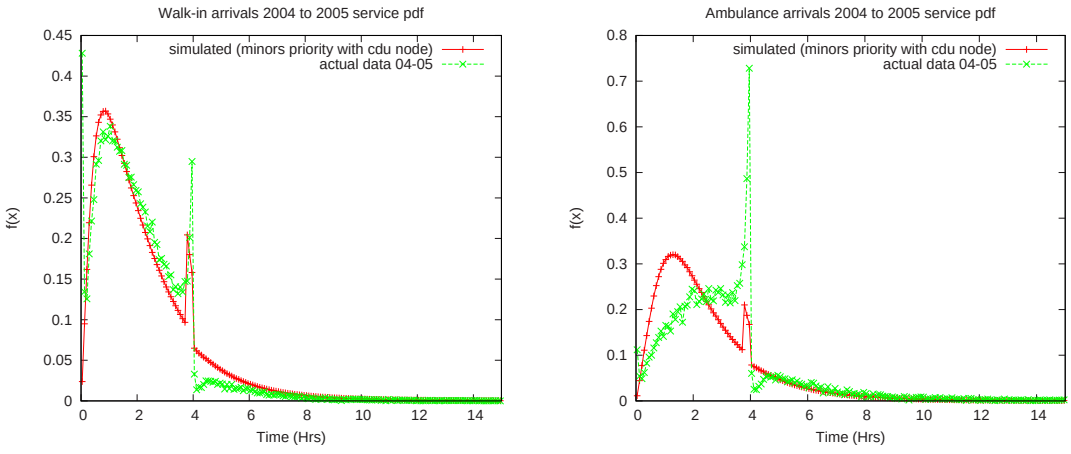


Figure 4.12: Actual and simulated service time density comparing the minors priority with CDU node for walk-in (left) and ambulance (right) arrivals with actual data.

As expected we can see from Table 4.9 that the mean waiting times in A&E for both arrival types are less pronounced under the minors priority with the CDU node. If we now look at the service time densities in Fig. 4.12 we can see that we have a corresponding spike around the four hour mark for both walk-in and ambulance arrivals;

however, both spikes are much lower than for the actual densities, especially for the ambulance arrivals case. We can also see in the actual densities that there is a dip after the four hour spike which is not replicated by the addition of the CDU node.

4.9.2 Probabilistic Adjustment of Patient Service Times

We next experiment with manually altering the response time density under minors priority via a mathematical formula. In particular we reallocate a percentage of patients whose response times lie in the 40 minutes after 4 hours to the 20 minutes before.

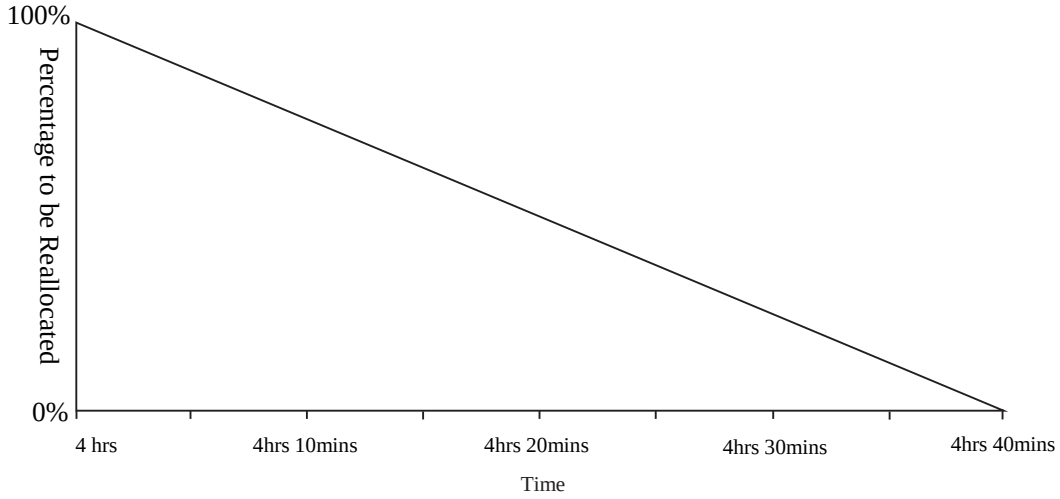


Figure 4.13: Illustration of the percentage of patient service times reallocated from the 40 minutes after 4 hours.

The percentage reallocated $R(t)$ is dependent on the amount of time that has elapsed after the 4 hour mark, as illustrated in Fig. 4.13 and is given by:

$$R(t) = \begin{cases} 100(1 - \frac{3}{2}(t-4)) & \text{if} \quad 4 \leq t \leq 4.6667 \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

When we compute the service time densities, we use five minute "buckets" which means all patient service times that fall into the same 5 minute range are counted together. We then take the mid-point of the range when plotting the densities. When using Equation 4.1 we will take the mid-point to represent the range and reallocate the same percentage for the whole bucket. Thus for instance in the time period 4hrs 30 mins to 4hrs 35 mins, we take the midpoint $t = 4.54167$, which from Equation 4.1 means that 66.67% of the patients falling into this time period will be reallocated.

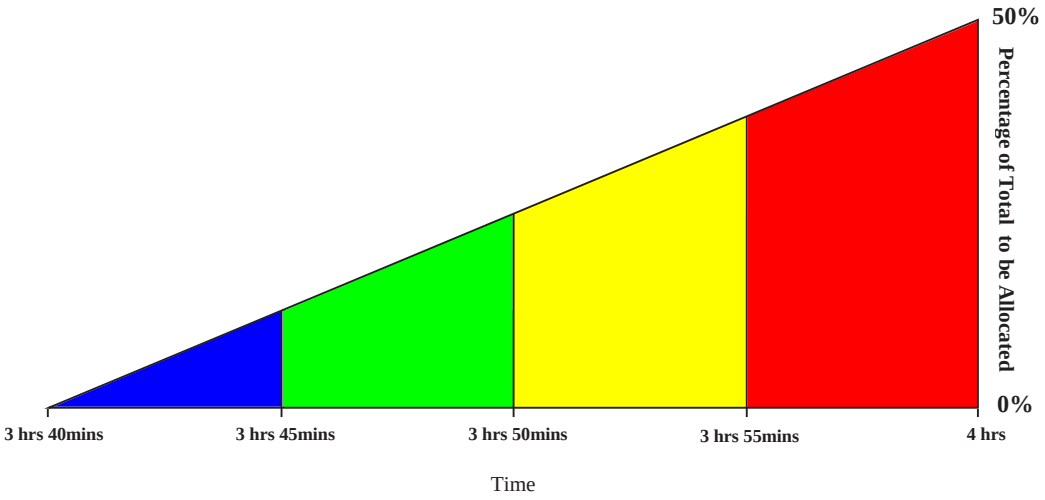


Figure 4.14: Illustration of the percentage of the total patient service times to be added to the service time densities of 20 minutes before 4 hours.

Once we have calculated the amount of patients to be reallocated from each bucket (8 in the time between 4 and 4.6667 hours), we sum to obtain the total number of patients to be reallocated. A percentage of this total sum is then added to each of the 4 buckets that make up the 20 minutes before 4 hours (i.e. the time from 3.6667 to 4 hours). Similar to the reallocation, the percentage of the total to be added $A(t)$ is dependent on the amount of time that has elapsed after the 3.6667 hour mark; this subsequent allocation is illustrated in Fig. 4.14 and is given by:

$$A(t) = \begin{cases} 100(\frac{3}{2}(t - 3.6667)) & \text{if} \quad 3.6667 \leq t \leq 4 \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

As for when using Equation 4.1, when applying Equation 4.2 we will take the mid-point of a bucket to represent the entire range and allocate the same percentage of the total sum to the whole bucket. The allocation is as follows:

- For the time period 3hrs 55mins to 4hrs (shown in red in Fig. 4.14), the midpoint is $t = 3.9583$, which using Equation 4.2 gives the percentage to be allocated to be 43.75%
- For the time period 3hrs 50mins to 3hrs 55mins (shown in yellow in Fig. 4.14), the midpoint is $t = 3.8750$, which using Equation 4.2 gives the percentage to be allocated to be 31.25%

- For the time period 3hrs 45mins to 3hrs 50mins (shown in green in Fig. 4.14), the midpoint is $t = 3.7917$, which using Equation 4.2 gives the percentage to be allocated to be 18.75%
- For the time period 3hrs 40mins to 3hrs 45mins (shown in blue in Fig. 4.14), the midpoint is $t = 3.7083$, which using Equation 4.2 gives the percentage to be allocated to be 6.25%

Notice that the total percentage to be allocated sums up to 100% as would be required.

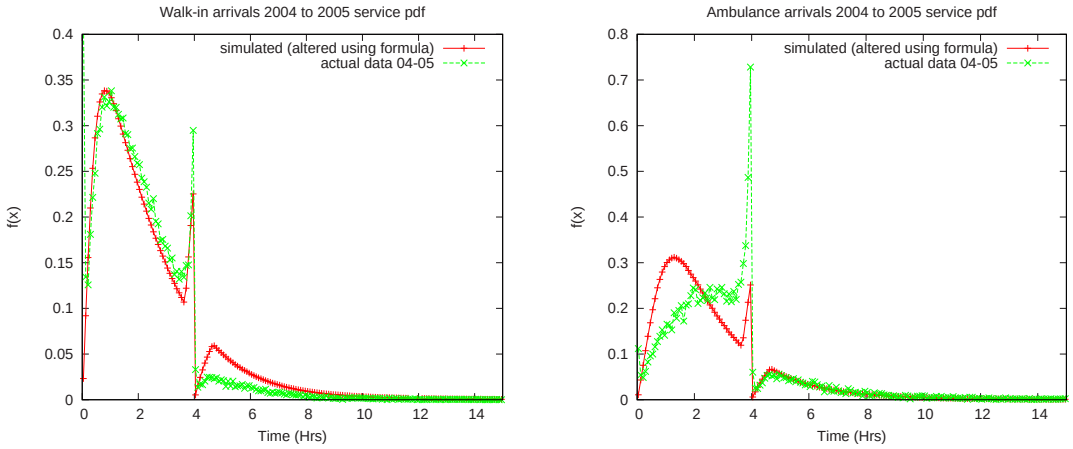


Figure 4.15: Actual and simulated service time density comparing the minors priority (adjusted using the mathematical formula) for walk-in (left) and ambulance (right) arrivals with actual data.

The resulting response time densities are shown in Fig 4.15. We can see that for the walk-in arrival service time density, the height of the spike is closer to the actual one seen in the data but the area after four hours still exhibits a higher tail than seen in the actual data. For the ambulance arrival service time density we can see that though the spike is higher than before (and in the right position) it is still much smaller than actually observed; however, the tail now matches almost exactly.

4.10 Workload and Resource Scenarios

In this section we experiment with differing workloads and resource/staffing scenarios. We run the resource/staffing scenarios using our simulation model under minors priority as we have found this priority scheme to be the closest match to the current real system. From the scenario results we hope to gain insights into how the real life system would

behave under increasing workloads, as well as identifying the resources that have the most impact on patient service times.

4.10.1 Workload Scenarios

We first investigate the workload levels at which each class-based priority scheme is most effective in terms of lower mean walk-in and ambulance arrival patient service times in our A&E model. We vary the rate of walk-in and ambulance arrivals from 25% to 100% of actual workload and compare the resulting service time means under the three class-based priority schemes. Since it is effectively a standalone subsystem, we do not alter the blue call arrival rate, or present any blue call results. Figs. 4.16 and 4.17 show the results for walk-in and ambulance arrivals respectively. The corresponding tables of results for walk-in and ambulance arrivals are shown in Tables D.1 and D.2 respectively in Appendix D.

From Figs. 4.16 and 4.17 we can see that under low loading (between 25% and 50% of workload) the priority scheme has little impact on mean service time for either of the arrival types. For ambulance arrivals, when we have medium loading (50% to 85%) we can see that they perform better under majors priority (as expected) than both no priority and minors priority. However, when we get to slightly higher workloads (85% to 95%) the no priority scheme gives lower mean response times than both majors priority and minors priority schemes. As the system reaches full loading (95% to 100%) the lowest mean response times come under minors priority. From Fig. 4.16 we can see that for the walk-in arrivals, minors priority performs the best once the loading is above 60% with the mean response times much lower under minors priority for higher loads (90% to 100%) with the mean service time for full loading under minors priority being 2.39 hours compared to 4.44 and 3.04 hours for majors priority and no priority schemes respectively.

Next we look at the effect of increasing workload over and above the current level under minors priority. We found that if we increased the workload by more than 15% the system reaches saturation (i.e. the system is not able to move patients out of the system quickly enough, creating very large queues at the critical resources that only grow larger

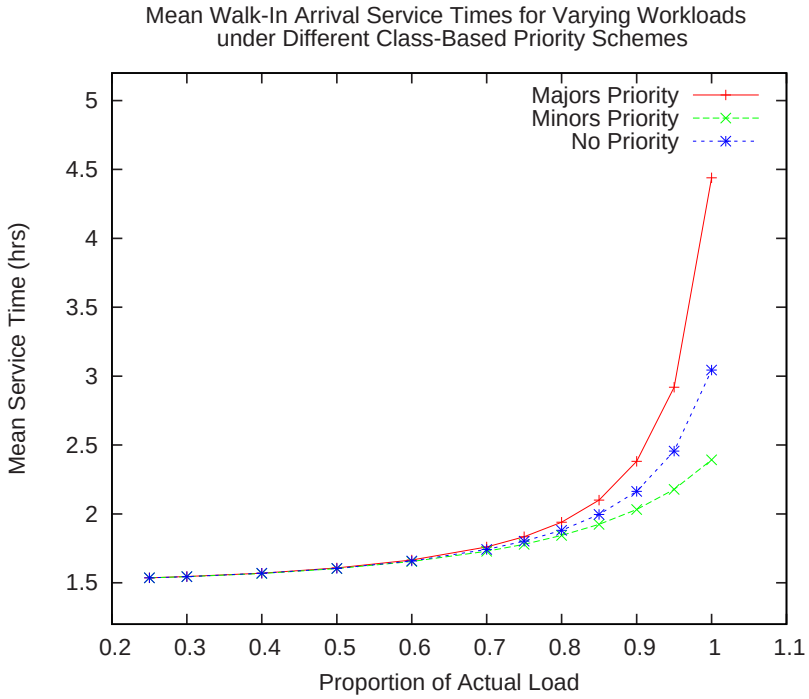


Figure 4.16: Walk-in arrival service time means for varying workloads under the different class-based priority schemes.

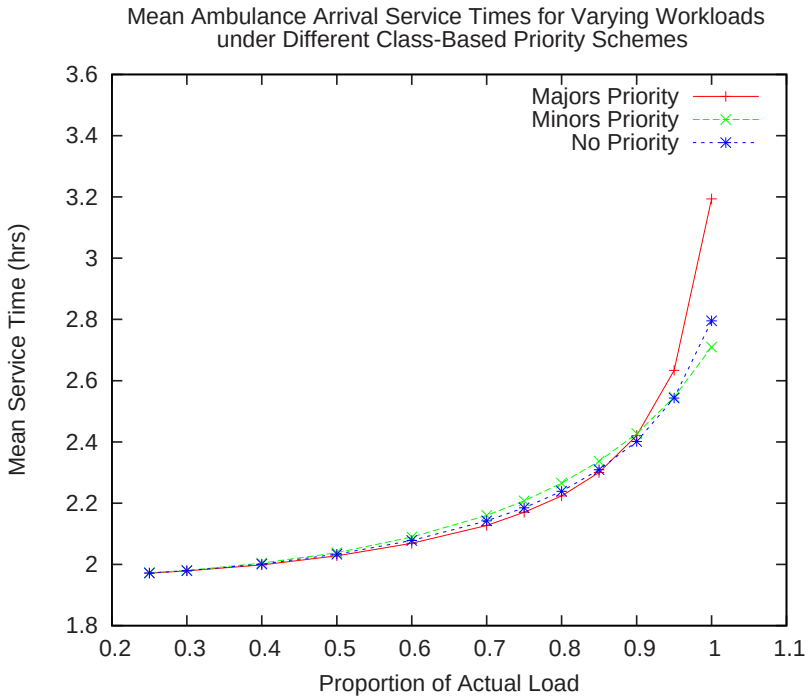


Figure 4.17: Ambulance arrival service time means for varying workloads under the different class-based priority schemes.

with time). The results are also shown in the Tables D.1 and D.2 in Appendix D and in Fig. 4.18.

Fig. 4.18 shows that once workload goes above 110% without corresponding resource and staff increases, the mean service times for both arrival types increases steeply; in addition to this the mean service time for walk-in arrivals starts to go above that for ambulance arrivals indicating that queues mostly build up in the minors area. These results suggest that our case study A&E department is operating close to saturation and that even a small increase in the workload (5%) would lead to a big jump in mean response times for both walk-in (up by 20.5 minutes) and ambulance (up by 14.6 minutes) arrivals.

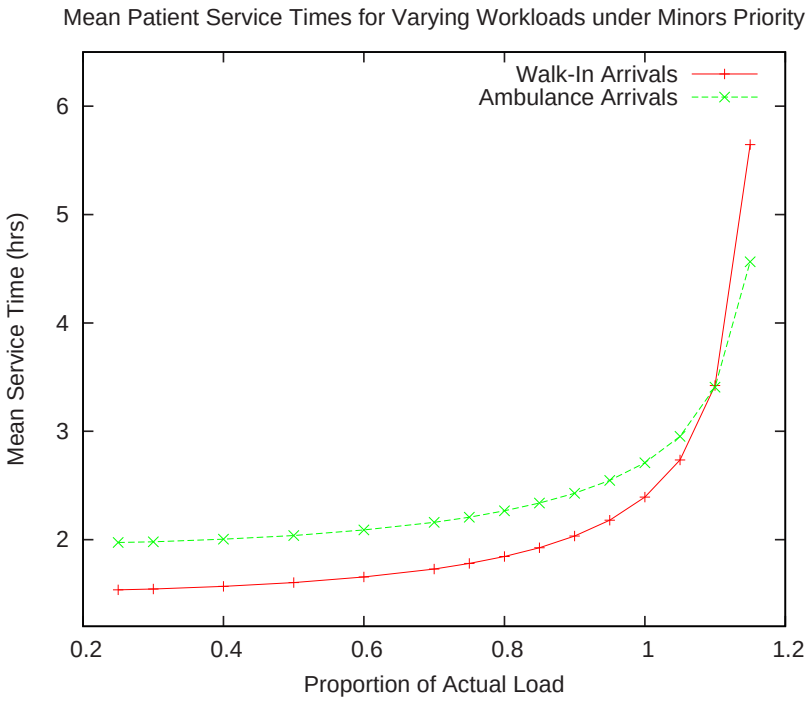


Figure 4.18: Walk-in and ambulance arrival service time means for varying workloads under minors priority.

4.10.2 Resource and Staff Scenarios

In the system under minors priority, the most highly utilised resources in the main A&E unit are the minors cubicles, minors practitioners, other specialists and radiology (see Table 4.5). Servers in the system with the highest utilisations indicate the bottlenecks in the system. We now look at various resource scenarios where we increase the number

of the most highly utilised resources in the system to see the subsequent impact on the mean service times.

First we increase the number of minors cubicles in the system by 1 (from 9 to 10) whilst keeping all other parameters the same and run the simulation. We then restore the number of minors cubicles and increase the number of minors practitioners (from 4 to 5) and then run the simulation again. This is repeated for the other specialist (increasing from 2 to 3) and then for radiology scanners (increasing from 2 to 3).

Table 4.10 compares the mean and standard deviations of the service times for walk-in and ambulance arrivals, with the extra resources and without. Table 4.11 shows the corresponding impact on the utilisations.

	walk-in arrivals		**ambulance arrivals**	
extra resource	mean	std dev	mean	std dev
minors cubicle	2.3008	1.9932	2.6826	2.2148
minors practitioner	2.2063	1.9522	2.6490	2.2018
other specialist	2.1834	1.7191	2.4553	1.7476
radiology scanner	2.2722	1.9146	2.5668	2.1036
none	2.3926	2.0271	2.7096	2.2101

Table 4.10: Impact of extra resources on the mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals.

	extra resource				
	min cubicle	**min prac**	**other spec**	**radiology**	**none**
resource	**utilisation**				
minors cubicle	0.7147	0.7170	0.7400	0.7546	0.7751
majors bay	0.4596	0.4588	0.4078	0.4293	0.4581
minors practitioner	0.7512	0.6010	0.7510	0.7512	0.7512
other specialist	0.7564	0.7569	0.5041	0.7571	0.7571
radiology scanner	0.6667	0.6670	0.6662	0.4442	0.6667

Table 4.11: Utilisation of a selection of staff and resources under a system with extra resources.

We can see from Table 4.10 that the introduction of an extra other specialist has the most impact on the mean service times for both walk-in and ambulance arrivals; this is followed by an extra minors practitioner for walk-in arrivals, but by an extra scanner at radiology for ambulance arrivals. From Table 4.11 we can see that increasing the number of other specialists and servers in radiology decreases the utilisations of both the minors cubicles and majors bays, with the extra other specialist giving the lowest

utilisations for both. This indicates that the other specialist is the major bottleneck for both arrival types; this may be because of the longer service times at this node than for either the minors practitioner or radiology. This is an interesting finding in the light of the current drive towards A&E reorganisation partly led by the shortages in specialists [42, 34, 15].

Next we investigate which resource has the biggest impact when the workload level is increased. We increase the workload to 115% of current workload and at the same time we increase each of the three greatest resource bottlenecks in turn. Tables 4.12, 4.13 and 4.14 show the results for increasing the number of minors practitioners, other specialists and radiology scanners respectively. Figs. 4.19 and 4.20 show the corresponding plots of mean service time for walk-in and ambulance arrivals.

	extra minors practitioner			
	walk-in arrivals		**ambulance arrivals**	
workload	mean	std dev	mean	std dev
1.05	2.4219	2.1747	2.8573	2.5097
1.1	2.7680	2.5441	3.2029	3.0099
1.15	3.4226	3.4103	3.9386	4.2236

Table 4.12: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under increasing workload and an extra minors practitioner in the system.

	extra other specialist			
	walk-in arrivals		**ambulance arrivals**	
workload	mean	std dev	mean	std dev
1.05	2.4073	1.8386	2.5853	1.8323
1.1	2.7653	2.0466	2.7739	1.9670
1.15	3.4915	2.5529	3.1076	2.2693

Table 4.13: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under increasing workload and an extra other specialist in the system.

From Tables 4.12, 4.13 and 4.14 and Figs 4.19 and 4.20, we can see that for workloads up to 110%, increasing the number of other specialists has the greatest impact on lowering the mean service times for both the walk-in and ambulance arrivals. However, as we go over a 10% increase in workload we can see that having an extra minors practitioner has a greater impact for walk-in arrivals. This indicates that as the system workload increases, the minors area will reach saturation first. As before, increasing

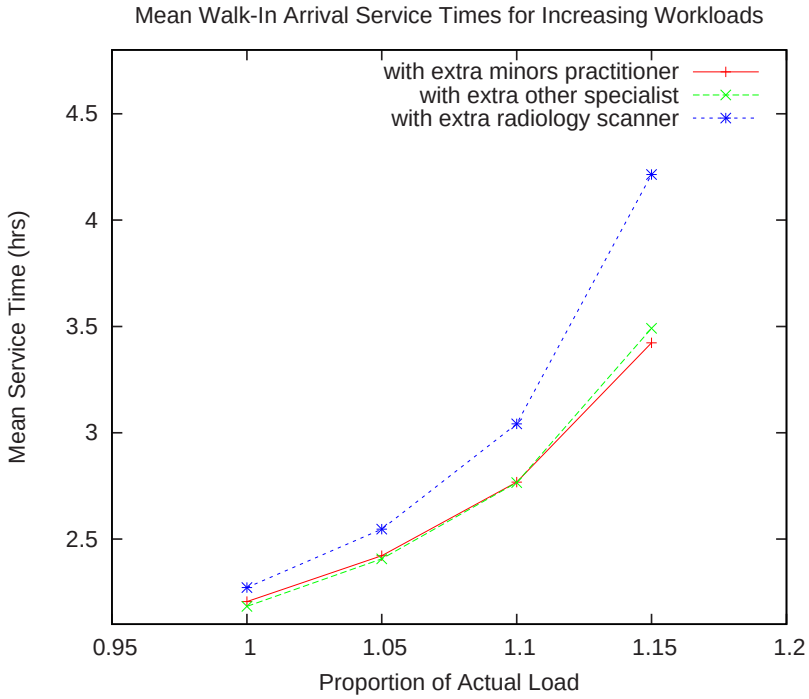


Figure 4.19: Walk-in arrival service time means for increasing workloads with an extra minors practitioner, extra other specialist or extra scanner in radiology.

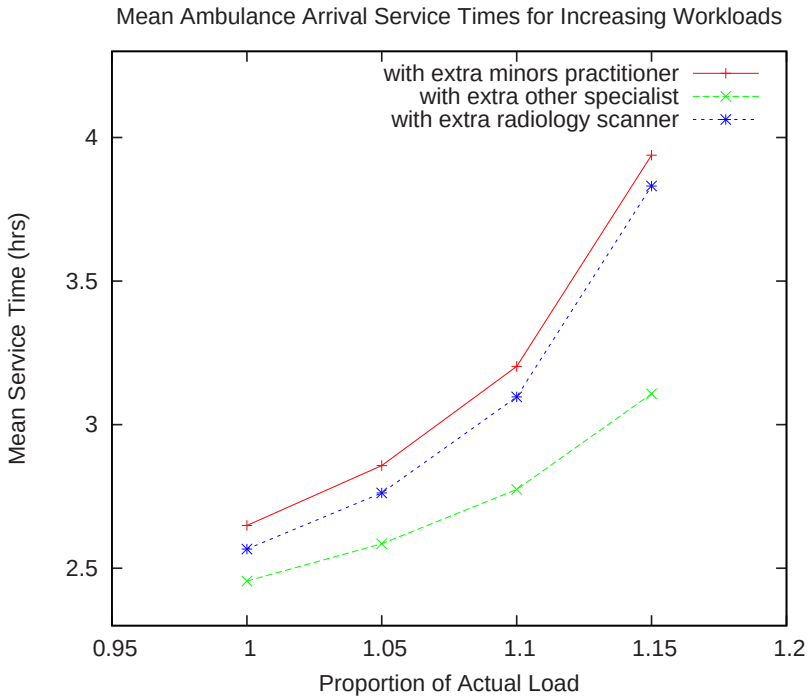


Figure 4.20: Ambulance arrival service time means for increasing workloads with an extra minors practitioner, extra other specialist or extra scanner in radiology.

	extra radiology server			
	walk-in arrivals		ambulance arrivals	
workload	mean	std dev	mean	std dev
1.05	2.5469	2.1431	2.7621	2.3746
1.1	3.0422	2.5791	3.0965	2.8439
1.15	4.2140	3.6948	3.8309	3.8904

Table 4.14: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under increasing workload and an extra server at radiology in the system.

the number of scanners in radiology has a greater impact than increasing the number of minors practitioners for ambulance arrivals, but less so for walk-in arrivals. It is also of interest to note that the standard deviation of service time with the extra other specialist is much lower for both walk-in and ambulance arrivals than when increasing the minors practitioner or radiology scanners – this indicates that the other specialist is the major bottleneck for those patients with higher service times.

4.11 Conclusion

In this chapter we have presented a multiclass queueing network model and compared our subsequent discrete-event simulation results and service time densities of A&E patient flow with those observed in an actual A&E department. We have provided some insights into the effects of different patient priority schemes and the impact of the introduction of the 4 hour waiting time target. The practical effect of this appears to have been to move from a system in which majors patients are given priority, to a system in which minors patients are given priority.

From Table 4.3 it appears ambulance arrivals actually benefit from a system under minors priority. In fact under minors priority both ambulance and walk-in arrivals are treated quicker than in a no priority system. However, this interpretation may be misleading: a significant proportion of ambulance arrivals end up as minors (about 35%) and their benefit outweighs the penalty suffered by the majors that arrive by any means. Conversely, the walk-in majors patients are highly penalised because relatively few walk-in minors patients switch to majors (about 16%). A separate comparison of ambulance arrivals that are treated as majors throughout their stay against walk-

ins that are treated as minors throughout, i.e. neglecting any patients that change class, would reveal the true effects of changing between majors and minors priority. However, it must be remembered that the most important statistics to the individual patient concern their own time spent in hospital, regardless of the class to which they may be assigned.

When comparing mean patient service times from our minors priority simulation model with the observed 2004/2005 figures (shown in Table 4.1), we observe differences of 17.3%, 13.7% and 0.3% for walk-in, ambulance and blue call patients respectively. The disagreement between the simulation and walk-in arrival actual service time may be because the measures taken to process patients in under four hours have not been incorporated into the minors priority scheme, resulting in disparity with the actual service time density around the 4 hour peak (cf. Fig 4.5). The disparity with the ambulance arrival service time mean may be due to the lack of blocking phenomena in our model, which will mostly delay ambulance arrivals. However, considering the many simplifying assumptions we have made the agreement between the simulation and actual arrivals is promising.

In Section 4.8 we investigated two time-based priority schemes and found that these lead to higher mean service times for both walk-in and ambulance arrivals than a straightforward minors priority scheme. We tried implementing a traffic light priority scheme to replicate the effect of the four hour patient waiting time target, but found this priority scheme does not result in the spike seen at four hours in the actual service time density (see Figs. 4.8 and 4.9). In order to try and replicate the impact of the four hour target, we inserted an extra clinical decision unit (CDU) node into the model which patients are sent to if they are close to the four hour waiting time limit. While this method does produce a spike at four hours, this spike is not pronounced enough. We found that by adjusting patient service times via a probabilistic reallocation mechanism we got a much closer fit to the actual service time density, especially for walk-in arrivals.

Finally, in Section 4.10 we provided some insights into how the system behaves when the workload and resource levels are varied. We found that for low to medium workloads, mean service times for ambulance arrivals benefit from a system under majors priority, but under high workloads both arrival types perform better under a minors priority

scheme. We have also shown that the main bottleneck in the system for both arrival types is the other specialist followed by the minors practitioner for the walk-in arrivals and radiology for the ambulance arrivals.

Chapter 5

Approximate Generating Function Analysis (AGFA) Technique

5.1 Introduction

In this chapter we present an approximate analytical technique which provides an efficient way to approximate the mean and standard deviation of response time in networks of multiclass queues with blocking and class-dependent priorities. Support for the latter two phenomena allow the technique to be applied in diverse modelling scenarios and makes this technique especially suited to the analysis of systems in healthcare.

Over many decades, extensive use has been made of queueing networks as an effective modelling abstraction. For certain classes of queueing networks including multiclass queueing networks and queues with blocking [87], Mean Value Analysis (MVA) [92, 91] and a plethora of related techniques (e.g. [6, 86, 98, 88, 101]) provide an efficient and elegant route to mean values of measures of interest (such mean waiting time and throughput), but not higher moments. For closed queueing networks with underlying (semi-)Markov chains, recent much more computationally-intensive methods based on numerical Laplace transform inversion can be applied to determine exact moments and, where tractable, probability distributions, of customer service times (cf. Section 2.6).

However, when applied to our multiclass queueing network model of A&E patient flow, this method suffers from the well-known *state space explosion problem* and so is limited to models with of the order of around 20 million states. Since accurate models of real life systems typically have much larger state spaces, especially when modelling large numbers of customers, performance analysts must often resort to simulation. As illustrated in Chapter 4, while simulation can be used to model complex systems at arbitrary levels of detail, it typically requires a high cost and effort to construct an accurate model and long execution times are often required to produce reliable results that are bounded by narrow confidence intervals.

The search for efficient analytical solutions in diverse modelling scenarios has prompted the development of approximate methods including the approximate generating function analysis (AGFA) technique that is the focus of this chapter. We present and then apply the AGFA technique to our hierarchical multiclass queueing network model of A&E under class-based patient priorities and compare results with our simulation.

The remainder of this chapter is arranged as follows. Section 5.2 presents technical details of the AGFA technique. Section 5.3 describes the alterations made to the queueing network model of A&E before applying the AGFA technique. Section 5.4 presents numerical results and graphs from both the AGFA method and the discrete-event simulation. Section 5.5 concludes.

5.2 Approximate Generating Function Analysis

The AGFA technique comprises of a decomposition method whereby the queueing network is broken up into sub-systems and each sub-system is analysed in isolation. These results are then combined together using a mean value analysis (MVA) extension to higher moments that utilises the general distributional Little's law (cf. Section 2.4).

The essence of the technique is that of Cobham's formula (cf. Section 2.4.6) for calculating mean values of response times in $M/G/1$ queues. This uses the fact that the mean value of a sum of random variables is equal to the sum of the corresponding means, whether or not the variables are independent. Furthermore, given the mean

sojourn time of a low priority customer in a queue, the mean number of higher priority arrivals in that time can be calculated. This analysis is adapted to the calculation of the Laplace transform of response time probability density, which is the expectation of the exponential function of a sum of random variables. Single nodes are analysed in this way, after which sub-networks are solved and aggregated according to the hierarchical MVA approach [92, 44].

5.2.1 Notation

We consider a network with two customer classes and M multi-server nodes, with m_i constant rate exponential servers with rates μ_{ir} at node i for class r ($1 \leq i \leq M, r = 1, 2$). Class 1 has non-pre-emptive priority over class 2. In particular, to enable our response time analysis, we consider the passage of a special "tagged" customer through queueing node i and define the following random variables at equilibrium:

$\mathbf{K}$ $= (K_1, K_2)$ class population vector, i.e. there are K_r customers of class r in the network ($r = 1, 2$);

B_{ir} class r service time of a single server at node i, exponential with parameter μ_{ir};

L_{ir} number of class r customers in the queue waiting to start service;

Q_{ir} time spent by a class r customer waiting to *start* service;

N_{ir} number of class r customers in the queue, including any in service, at a random instant of time (i.e. the class r queue length);

W_{ir} response time of a class r customer, i.e. the sum of queueing time and service time, $Q_{ir} + B_{ir}$;

Let the steady state probability that the (joint) queue length at node i is $\mathbf{n} = (n_1, n_2)$ be $\pi_i(\mathbf{n} \mid \mathbf{k}) = \mathbb{P}(N_{i1} = n_1, N_{i2} = n_2 \mid \mathbf{K} = \mathbf{k})$. We will make use of the probability that an arriving customer has to queue, q_{ir}. In a network with processor sharing servers and no priorities, this is just the probability that the equilibrium queue length is less than m_i when the population is reduced by one in the arriving customer's class (denoted by

$\mathbf{k}^{r-}$), by the arrival theorem (cf. Section 2.4.4). Thus:

$$1 - q_{ir} = \mathbb{P}(N_{i1} + N_{i2} < m_i \mid \mathbf{K} = \mathbf{k}) = \sum_{u=0}^{m_i-1} \sum_{v=0}^{m_i-1-u} \pi_{ir}(u, v \mid \mathbf{k}^{r-})$$

for appropriate $\mathbf{k}, \mathbf{k}^{r-} (r = 1, 2)$.

Letting $E[\cdot]$ and $E[\cdot|\cdot]$ denote the expectation and conditional expectation operators respectively, for a continuous random variable X, we denote its probability distribution function by $X(t) = \mathbb{P}(X \leq t)$ and the Laplace-Stieltjes transform of this distribution (the LSTD) by $X^*(\theta) = E[e^{-\theta X}]$. We denote the density function by $x(t) = X'(t)$, the derivative of the distribution function, with the Laplace transform of the density the same as the LSTD $X^*(\theta)$. We also denote the nth moment of X by $X_{;n} = E[X^n] = (-1)^n X^{*(n)}(0)$ (where the parenthesized superscript denotes differentiation n times). Thus, for example, $S_{2;1}$ is the mean of S_2.

For a discrete random variable Y, we denote its probability generating function (pgf) by $G_Y(z) = E[z^Y]$ and the nth *factorial moment* of Y by:
$Y_{;fn} = E[Y(Y-1)\ldots(Y-n+1)] = G_Y^{(n)}(1)$.

5.2.2 An Approximate MVA Algorithm

Class 1

The high priority class 1 customers are straightforward to handle since the tagged customer only has to wait for those class 1 customers already queueing and the customer in service (of either class), if any. Consider a generic node i in a closed network of M queues. Dropping the subscripts ir for brevity, we have for class 1:

$$Q_1^*(\theta) \quad = \quad E[\, E[e^{-\theta(S_1+\ldots+S_L+UR)} \mid N_1, N_2]\,]$$

where the random variable U is defined by:

$$U \quad = \quad \begin{cases} 1 & \text{if } \; N_1 + N_2 \geq m \\ 0 & \text{if } \; 0 \leq N_1 + N_2 < m \end{cases}$$

The random variables S_l where $l = 1, 2, \ldots, L$ are independent and identically distributed (i.i.d.) as the minimum of the m service time random variables at the individual servers. Therefore each is exponential with parameter $m\mu$ in a single class node. In the multiclass case, they are still exponential but have parameter $m_1\mu_1 + (m - m_1)\mu_2$ when there are m_1 class 1 and $m - m_1$ class 2 customers in service. We make the approximating assumption that, given the class of the tagged customer, the network's population vector $\mathbf{k}$ and the state encountered on arrival $\mathbf{n}$, this rate remains the same throughout the tagged customer's sojourn in the queue, Q_1, viz. $n_1\mu_1 + n_2\mu_2$ if $n_1 + n_2 \leq m$ and $(n_1\mu_1 + n_2\mu_2)m/(n_1 + n_2)$ if not. Of course, this result is exact if the service rate is the same for both classes ($\mu_1 = \mu_2$) and in the single class case ($n_2 = 0$). Note too that we would not have this problem if the priority discipline were pre-emptive, whereupon no class 2 customer could be in service if m or more class 1 customers were present.

The random variable R is the time to the next service completion from the arrival instant of the tagged customer. By the memoryless property, R is distributed as the S_l and is also independent by hypothesis.

Noting that $R^*(\theta U) = R^*(\theta) + (1 - R^*(\theta))(1 - U)$ we therefore obtain (for class 1):

$$\begin{aligned} Q_1^*(\theta) &= E[\, S^*(\theta)^{L_1}\, E[e^{-\theta U R} \mid N_1, N_2]\,] \\ &= E[\, S^*(\theta)^{L_1}\, R^*(\theta U)] \\ &= G_{L_1}(S^*(\theta))R^*(\theta) - (1 - q_1)R^*(\theta) + 1 - q_1 \end{aligned} \tag{5.1}$$

where $G_{L_1}(z) = 1 - q_1 + \sum_{u=m}^{k_1} \sum_{v=m-u}^{k_2} \pi(u, v) z^{u-m_1}$ and m_1 is the number of class 1 customers in service when all servers are busy (recall q_1 is the probability that an arriving class 1 customer has to queue). This can be approximated for non-pre-emptive priority, as above, but in our calculation of moments it comes from the application of Little's law in the extended MVA algorithm. Note that $R^* = S^*$ and, in the calculations of Q_1 and $G_{L_1}(z)$, we assume a population vector $\mathbf{k}$ in which the component corresponding to the class of the arriving customer has been reduced by one in accordance with the arrival theorem. Notice that in the case of a single class $M/M/m$ queue

with constant arrival rate λ, we have $S^*(\theta) = m\mu/(m\mu + \theta)$ and:

$$\begin{aligned} G_L(z) &= 1 - q + (1-\rho)q + (1-\rho)q\sum_{l=1}^{\infty}\rho^l z^l \\ &= 1 - q + \frac{(1-\rho)q}{1-\rho z} \end{aligned}$$

where $\rho = \lambda/(m\mu)$. Consequently, we obtain the single class result:

$$Q_1^*(\theta) = 1 - q + \frac{(1-\rho)qm\mu}{m\mu + \theta - \rho m\mu} = 1 - q + q\frac{(m\mu - \lambda)}{m\mu - \lambda + \theta}$$

The moments of the class 1 queueing time follow by differentiation at $\theta = 0$. For the first few moments this is a straightforward process, but the n-fold differentiation of the term $G_{L_1}(S^*(\theta))$ for arbitrary n leads to ever-increasing complexity. It can be obtained simply using a programming language that supports an appropriate higher-order function – here differentiation with respect to θ – and otherwise using an auxiliary recursive definition [50]. Here we obtain the first two moments explicitly.

Mean queueing time for class 1 Differentiating the class 1 queueing time LSTD given by Equation 5.1, we find:

$$Q_1^{*\prime}(\theta) = G'_{L_1}(S^*(\theta))S^{*\prime}(\theta)S^*(\theta) + G_{L_1}(S^*(\theta))S^{*\prime}(\theta) - (1-q_1)S^{*\prime}(\theta) \quad (5.2)$$

At $\theta = 0$, we obtain:

$$Q_{1;1} = L_{1;1}S_{1;1} + S_{1;1} - (1-q_1)S_{1;1} = (L_{1;1} + q_1)S_{1;1}$$

as could have been obtained by a simple direct argument, whereby the mean queueing time for a class 1 customer is the mean time taken to serve the (on average) $L_{1;1}$ class 1 customers already in the queue plus the mean time taken to wait for a server if all the servers are busy. Now the probability that there is no server free is q_1 and the mean class 1 service time for each server is $S_{1;1}$. Hence the mean queueing time for class 1 customers will simply be the mean class 1 customer queue length plus the probability that the servers are all busy, all multiplied by the mean service time i.e. $Q_{1;1} = (L_{1;1} + q_1)S_{1;1}$ as above.

Second moment of queueing time for class 1 Differentiating Equation 5.2 at $\theta = 0$ we find similarly:

$$Q_{1;2} = L_{1;f2}S_{1;1}^2 + L_{1;1}S_{1;2} + 2L_{1;1}S_{1;1}^2 + S_{1;2} - (1 - q_1)S_{1;2}$$

which simplifies to:

$$Q_{1;2} = (L_{1;2} + L_{1;1})S_{1;1}^2 + (L_{1;1} + q_1)S_{1;2}$$

Class 2

Recall that class 1 customers have non-pre-emptive priority over class 2 customers. Consequently, a class 2 customer has to wait, not only for the service completion of any customer in service at its arrival instant and all class 1 and 2 customers already waiting, but also for all class 1 customers that arrive during its queueing time. As with class 1 customers, we assume that the total service rate remains constant throughout a class 2 customer's sojourn time in the queue, so that service times are the same random variables (S, R) that depend only on the state of the queue on arrival, (n_1, n_2).

Let C be the number of class 1 arrivals during the tagged customer's queueing time Q_2. Since these arrivals are assumed to be Poisson with rate λ_1 (and so have pgf $e^{-\lambda_1 t(1-z)}$ for a time period t), C has pgf defined by:

$$\begin{aligned} G_C(z) &= E[z^C] = E[\,E[z^C \mid Q_2]\,] \\ &= E[e^{-\lambda_1 Q_2(1-z)}] \\ &= Q_2^*(\lambda_1(1-z)) \end{aligned}$$

Writing $H = \max(N_1 + N_2 - m, 0)$, we therefore have:

$$\begin{aligned} Q_2^*(\theta) &= E[\,E[e^{-\theta(S_1+\ldots+S_H+S_{H+1}+\ldots+S_{H+C}+UR)} \mid N_1, N_2, Q_2]] \\ &= E[S^*(\theta)^H R^*(\theta U) E[S^*(\theta)^C \mid Q_2]] \\ &= E[S^*(\theta)^H R^*(\theta U) e^{-\lambda_1 Q_2(1-S^*(\theta))}] \end{aligned} \tag{5.3}$$

Mean queueing time for class 2 Setting out as for class 1, we first differentiate the class 2 queueing time LSTD given in Equation 5.3 to find:

$$\begin{aligned} Q_2^{*\prime}(\theta) &= E[HS^*(\theta)^{H-1}R^*(\theta U)e^{-\lambda_1 Q_2(1-S^*(\theta))}S^{*\prime}(\theta)] + \\ &\quad E[S^*(\theta)^H R^{*\prime}(\theta U)Ue^{-\lambda_1 Q_2(1-S^*(\theta))}] + \\ &\quad E[S^*(\theta)^H R^*(\theta U)\lambda_1 Q_2 e^{-\lambda_1 Q_2(1-S^*(\theta))}S^{*\prime}(\theta)] \\ &= E[S^*(\theta)^{H-1}e^{-\lambda_1 Q_2(1-S^*(\theta))}\big(HR^*(\theta U)S^{*\prime}(\theta) + \\ &\quad US^*(\theta)R^{*\prime}(\theta U) + \lambda_1 Q_2 S^*(\theta)R^*(\theta)S^{*\prime}(\theta)\big)] \end{aligned} \tag{5.4}$$

At $\theta = 0$, we therefore obtain:

$$Q_{2;1} = H_{2;1}S_{2;1} + q_2 R_{2;1} + \lambda_1 Q_{2;1}S_{2;1}$$

since $E[U] = q$. This gives Cobham's familiar result for mean values (see for example [53]):

$$Q_{2;1} = \frac{(H_{2;1} + q_2)S_{2;1}}{1 - \lambda_1 S_{2;1}}$$

The mean values $S_{2;1}, q_2$ and $H_{2;1}$ depend on the state existing just before an arrival instant, as discussed above, and can be computed as part of the standard variable rate MVA algorithm that we use.

Second moment of queueing time for class 2 Although the analysis of mean values is straightforward, not actually needing generating functions at all, the situation is much more complex for higher moments because of the dependence amongst the random variables concerned (Q_i, L_i, U). In particular, this leads to covariance terms in the second moments.

We therefore define the two-variable generating function $A(z, \theta)$ by:

$$\begin{aligned} A(z,\theta) &= E[z^H e^{-\theta Q_2}] = E[E[z^H e^{-\theta Q_2} \mid N_1, N_2]] \\ &= E[z^H S^*(\theta)^H R^*(\theta U)e^{-\lambda_1 Q_2(1-S^*(\theta))}] \end{aligned}$$

by the same reasoning as in the previous section.

Taking the expectation w.r.t. U, we obtain:

$$\begin{aligned} A(z,\theta) &= 1-q_2+R^*(\theta)\left\{E\Big[\big(zS^*(\theta)\big)^H e^{-\lambda_1 Q_2(1-S^*(\theta))}\Big]-(1-q_2)\right\} \\ &= (1-q_2)\big(1-S^*(\theta)\big)+S^*(\theta)\,A\big(zS^*(\theta),\lambda_1(1-S^*(\theta))\big) \end{aligned}$$

Now let $y = zS^*(\theta)$, and $\phi = \lambda_1(1-S^*(\theta))$ so that $y=1$ and $\phi=0$ when $z=1$ and $\theta=0$. Using primes to denote differentiation of a function of a single variable and the facts that $\frac{\partial y}{\partial z}=S^*(\theta), \frac{\partial y}{\partial \theta}=zS^{*\prime}(\theta), \frac{\partial \phi}{\partial z}=0, \frac{\partial \phi}{\partial \theta}=-\lambda_1 S^{*\prime}(\theta)$, so that $\partial/\partial\theta = zS^{*\prime}(\theta)\partial/\partial y - \lambda_1 S^{*\prime}(\theta)\partial/\partial\phi$, we obtain:

$$\frac{\partial A}{\partial \theta} = \big(A(y,\phi)+q_2-1\big)S^{*\prime}(\theta)+S^*(\theta)S^{*\prime}(\theta)\left(z\frac{\partial A}{\partial y}-\lambda_1\frac{\partial A}{\partial \phi}\right) \tag{5.5}$$

Thus, at $z=1,\theta=0$, we obtain $-Q_{2;1} = -q_2 S_{2;1} - S_{2;1}(H_{2;1}+\lambda_1 Q_{2;1})$ so that:

$$Q_{2;1}(1-\lambda_1 S_{2;1}) = H_{2;1}S_{2;1}+q_2 S_{2;1}$$

as obtained already. Differentiating again at $z=1$ and $\theta=0$, omitting the arguments of functions for brevity, where the meaning is clear, and noting that $\frac{\partial z}{\partial y}=1/S^*$, $\frac{\partial z}{\partial \phi}=z/(\lambda_1 S^*)$ so that $z\frac{\partial z}{\partial y}=\lambda_1\frac{\partial z}{\partial \phi}$, and recalling that $H_{2;f2}$ denotes the second factorial moment of H_2, we now find:

$$\begin{aligned} \left.\frac{\partial^2 A}{\partial\theta^2}\right|_{1,0} &= q_2 S_{2;2}+2S_{2;1}^2[H_{2;1}+\lambda_1 Q_{2;1}]+S_{2;2}[H_{2;1}+\lambda_1 Q_{2;1}]- \\ &\quad S_{2;1}^2\left[z\left(z\frac{\partial^2 A}{\partial y^2}-\lambda_1\frac{\partial^2 A}{\partial y\partial\phi}\right)-\lambda_1\left(z\frac{\partial^2 A}{\partial\phi\partial y}-\lambda_1\frac{\partial^2 A}{\partial\phi^2}\right)\right]_{1,0} \\ &= q_2 S_{2;2}+(H_{2;1}+\lambda_1 Q_{2;1})(S_{2;2}+2S_{2;1}^2)+ \\ &\quad S_{2;1}\left[H_{2;f2}S_{2;1}-2\lambda_1 S_{2;1}\frac{\partial^2 A}{\partial y\partial\phi}+\lambda_1^2 S_{2;1}Q_{2;2}\right] \end{aligned} \tag{5.6}$$

We compute the covariance term $\frac{\partial^2 A}{\partial y\partial\phi}$ at $z=1,\theta=0$ as follows. First,

$$\frac{\partial A}{\partial z} = S^*\frac{\partial A}{\partial y}$$

since $\frac{\partial\phi}{\partial z}=0$. Differentiating w.r.t. θ now gives:

$$\frac{\partial^2 A}{\partial z\partial\theta} = S^{*\prime}\frac{\partial A}{\partial y}+S^*\left[\frac{\partial^2 A}{\partial y^2}zS^{*\prime}+\frac{\partial^2 A}{\partial y\partial\phi}(-\lambda_1 S^{*\prime})\right]$$

At $z = y = 1, \theta = \phi = 0$, and noting that $H_{2;2} = H_{2;f2} + H_{2;1}$, this yields:

$$\left.\frac{\partial^2 A}{\partial z \partial \theta}\right|_{1,0} = -\frac{(H_{2;1} + H_{2;f2})S_{2;1}}{1 - \lambda_1 S_{2;1}}$$

Finally, substituting into Equation 5.6 at $z = 1, \theta = 0$, we obtain:

$$\begin{aligned} Q_{2;2} &= \frac{q_2 S_{2;2} + (H_{2;1} + \lambda_1 Q_{2;1})(S_{2;2} + 2S_{2;1}^2) + H_{2;f2} S_{2;1}^2}{1 - \lambda_1^2 S_{2;1}^2} \\ &\quad + \frac{2\lambda_1 S_{2;1}^3 H_{2;2}}{(1 - \lambda_1 S_{2;1})(1 - \lambda_1^2 S_{2;1}^2)} \end{aligned} \tag{5.7}$$

$Q_{2;1}$ was computed in the previous subsection and, again, the expected value $H_{2;2}$ is computed in the MVA-based algorithm, considering the superposition of the two classes. The second moment $S_{2;2}$ is approximated as the average of the square of the service time of a single server, estimated at equilibrium when all servers are busy. This double approximation is a potentially major source of error in our model; however, it is exact when the two classes have identical service time random variables.

5.2.3 The MVA-based hierarchical model

Apart from the aforementioned moments of the time to the next service completion after the arrival instant of the tagged customer, the only state-dependent parameters that are needed for constant-rate, multi-server queues are the queueing probabilities q_1, q_2. In the case of a single server at equilibrium, this is just the utilisation, which is known to be the product of the arrival rate and the mean service time, by the usual steady-state argument or Little's law. However, multiple servers or state-dependent service times require that every (significant) queue length probability be computed in order to find q_1, q_2 and the first two moments of S_1, S_2 – at each queue and for each network population vector in a closed network. This is the main expense of the algorithm. It also goes some way to explaining why the problem has for long been solved for $M/G/1$ queues but remains open for $M/G/m$ for $m > 1$. Notice too the subtle dependence between the random variables involved that arises when considering non-exponential servers that precludes simply setting the moments of S_1, S_2 to those of the residual service time [51].

Network decomposition and aggregate servers In our hierarchical modelling methodology, we successively decompose a queueing network of multi-servers, where each individual server has constant rate, into a collection of sub-networks. This is a common approach to modelling large systems, pioneered to a considerable extent by Woodside and others in their analysis of layered queueing networks; see for example [44]. The sub-networks we identify as most appropriate are each solved, using the AGFA approach described in the previous subsections, for the first two moments of their response time, given each (multi-server) node's service time moments, the network's routing probabilities and the constant populations of its customer-classes. No class transitions are allowed within a sub-network (which constrains the choice of sub-networks, of course). Each node in a sub-network is analysed using the results of the previous section and Little's law (for both the first and second moments of queue length and waiting time) at class populations increasing from 0 to the maximum required. This is done in a straightforward modification of the standard MVA algorithm with state-dependent parameters to yield the required first two moments [53]. At the next level up, these moments are assigned to those of the individual service times at the corresponding multi-server nodes. The number of parallel servers at each node is set to the maximum population specified for each class in the sub-network at the lower-level – recall that no class transitions occur. Hence the class population maxima are preserved all the way up the hierarchy. The higher-level network is then fully parameterised by its routing probabilities, easily obtainable from the initial, flat network's specification.

At the top-level, we analyse an open network of aggregated nodes, at any of which there may be interaction between the classes. In particular, the service rate of each class may depend on the joint population of both classes currently at the node. Such a node is solved by a direct Markov model with state space truncation. This is not excessively expensive for a single node with just two classes, and in fact no more than about 2000 states are needed in practice. Nevertheless, this complicates the already expensive calculation of queue length probabilities which, as already mentioned, constitutes the major share of the computation time of the whole algorithm.

The only remaining quantities needed for the hierarchical algorithm are the mean and second moments of the numbers of visits a task makes to each node. These are derived

in the next section. This whole decomposition is implemented in Mathematica 5.1 [102], the code for which is shown in Appendix E. Clearly a lower-level implementation in a language such as C would be orders of magnitude more efficient numerically, and benefit especially the single node, direct Markov models discussed above.

Moments of visit counts

The MVA algorithms, open or closed, require the same moments of the nodes' visit counts as those required for response time. To this end, let the random variable V_{ir} denote the number (or rate) of visits a task of class r makes to node i and let X_{ir} be the visit count (or rate) of external arrivals, $1 \leq i \leq M, 1 \leq r \leq R$. Then we have:

$$E[z^{V_{ir}}] = E[z^{X_{ir}+\sum_{j,s} N_{js;ir}}]$$

where $N_{js;ir}$ is the number of class s service completions at node j that go to node i as class r. Thus:

$$\begin{aligned} E[z^{V_{ir}}] &= E[E[z^{X_{ir}+\sum_{j,s} N_{js;ir}} \mid V_{js}]] \\ &= E[z^{X_{ir}}] \prod_{j,s} E\left[\big(1 - p_{js;ir}(1-z)\big)^{V_{js}}\right] \end{aligned}$$

since the random variables $N_{js;ir}$ are independent and binomially distributed with parameters $(V_{js}, p_{js;ir})$. Hence we have:

$$G_{V_{ir}} = G_{X_{ir}} \prod_{j=1}^{M} \prod_{s=1}^{R} G_{V_{js}}\big(1 - p_{js;ir}(1-z)\big)$$

Differentiating once, then twice at $z = 1$ then yields:

$$V_{ir;1} = X_{ir;1} + \sum_{j=1}^{M} \sum_{s=1}^{R} p_{js;ir} V_{js;1} \tag{5.8}$$

$$\begin{aligned} V_{ir;f2} &= V_{ir;1}^2 + X_{ir;f2} - X_{ir;1}^2 + \\ &\quad \sum_{j=1}^{M} \sum_{s=1}^{R} p_{js;ir}^2 \big(V_{js;f2} - V_{js;1}^2\big) \end{aligned} \tag{5.9}$$

5.3 Accident and Emergency Model

To illustrate the use of the combined AGFA-MVA technique, which we abbreviate to just AGFA, we apply it to the hierarchical queueing network model of our case study A&E department (as described in Section 4.3). However, before we can apply the AGFA technique, this model needs to be adapted so that the lower-levels form closed sub-networks as described in the next section.

5.3.1 Closed Queueing Network Model

In this model, passive resources and all their associated *active* resources (i.e. those providing a service that actually progresses a patient through the treatment) are aggregated into a single node in the top-level model (see Fig. 5.1). Where these active resources include one shared with another class associated with a different passive resource, the union of the two sets of resources, associated with each passive resource, is aggregated – essentially giving a transitive closure. This leads to the AEU aggregate node (expanded in Fig. 5.2), which includes the minors cubicles, majors bays and all their associated resources.

The lower-levels of the model (see Fig. 5.2), consisting of the submodels (AEU, Assess and Resusc) are now closed sub-networks. Notice that the obtaining and releasing of passive resources are no longer indicated, since we evaluate each sub-network for class populations increasing from zero to the maximum possible (generally this will be dictated by the number of passive resources in the sub-network). Therefore the passive resources are not actually obtained or released, but taken to be always occupied or unoccupied as determined by the class population. Within this new altered model, class changes can only occur in the top-level of the model. When a patient switches from class 1 (minors) to class 2 (majors), this no longer happens within the AEU submodel; instead the patient has to first leave the AEU submodel as a class 1 patient and then re-enter the AEU aggregated node as a class 2 patient. Similarly for the class 2 (majors) patients switching to class 3 (resuscitation), the patient must leave the AEU submodel before they can switch to class 3 and then enter the resuscitation area.

Altered Top-Level Model

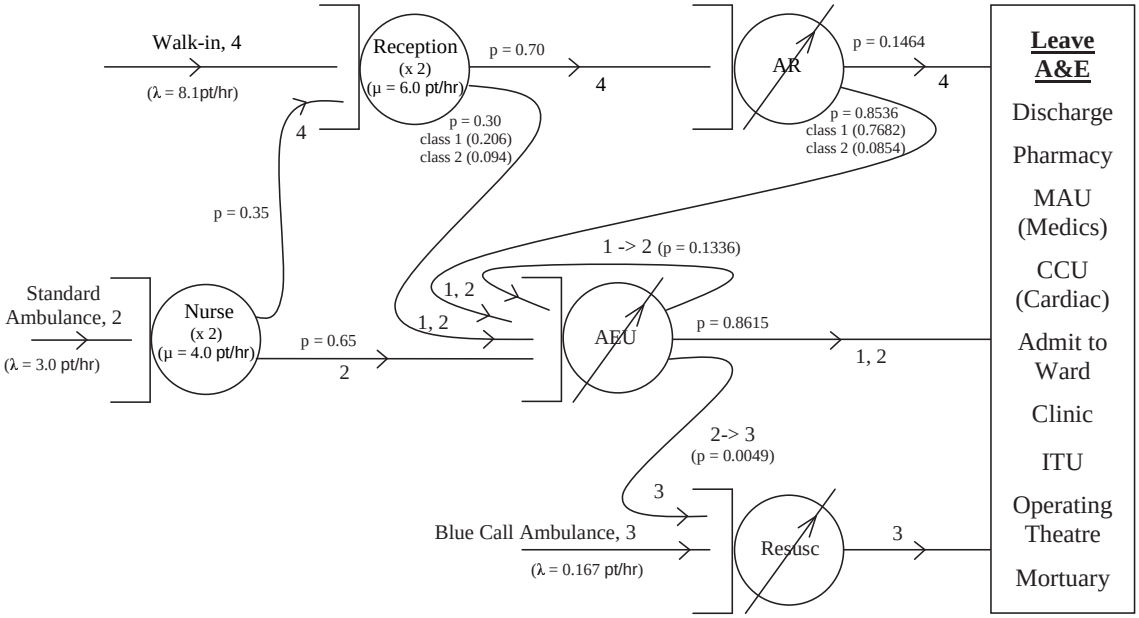


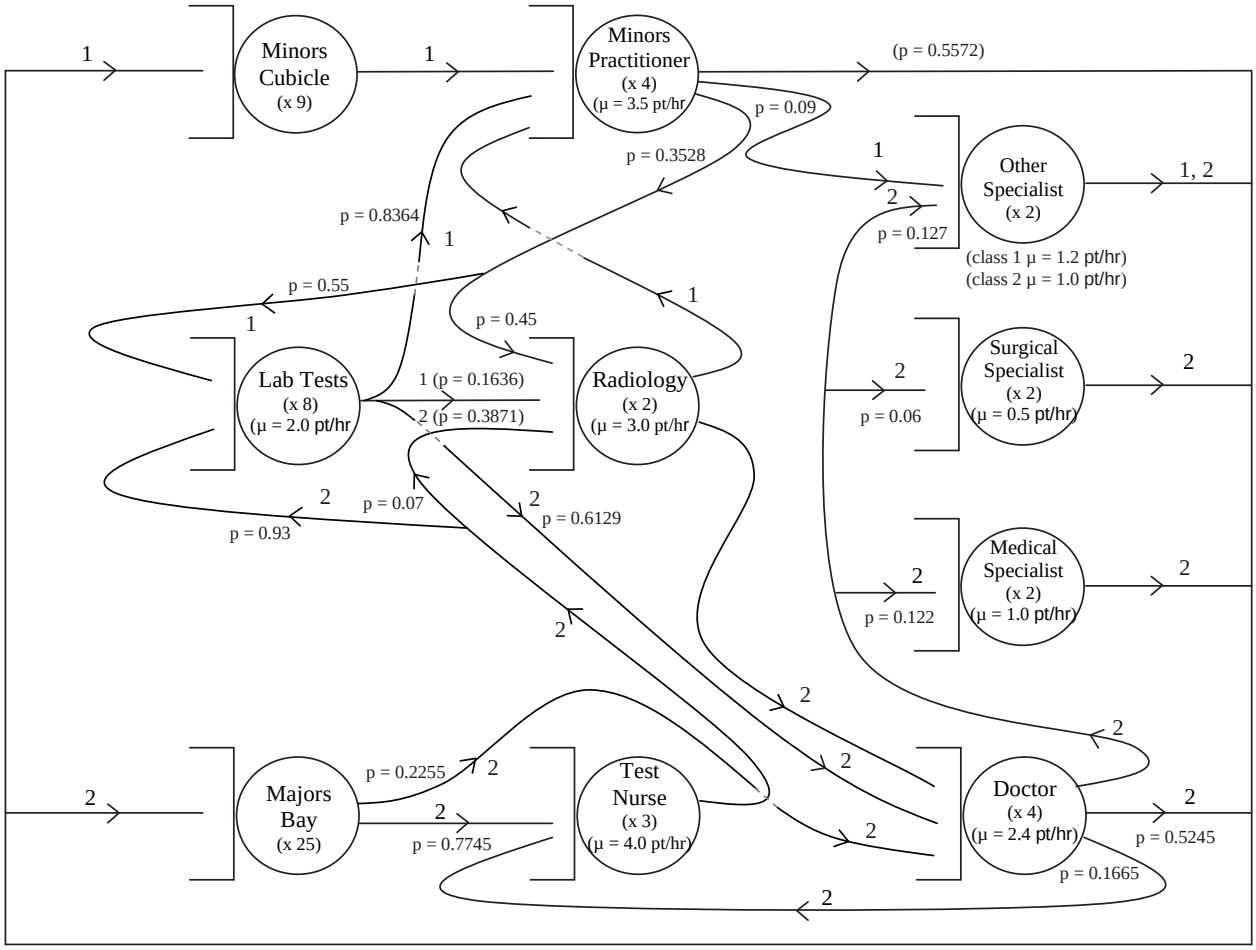
Figure 5.1: Altered top-level of queueing network model of patient flow.

5.3.2 Class-based Priority Schemes

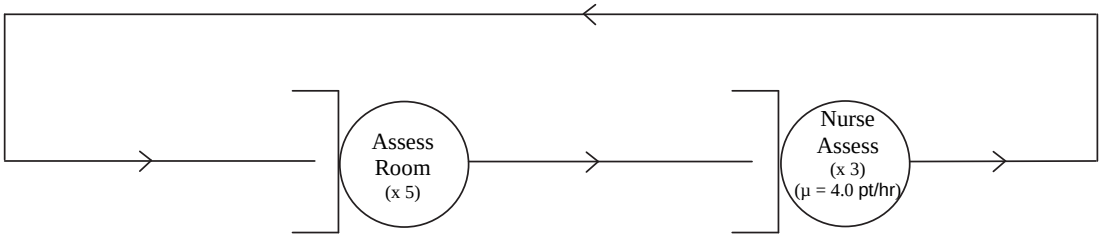
We investigate various patient class-based priority schemes, utilising the AGFA support for class dependent priorities. The three different patient class-based priority schemes analysed are:

- **No Priority** in which First In First Out (FIFO) queues are implemented at each node,
- **Majors Priority** in which majors patients are given priority at the shared resources (lab tests, radiology and "other" specialist), and
- **Minors Priority** in which minors patients are given priority at the shared resources.

Closed AEU Submodel



Closed Assess Submodel



Closed Resusc Submodel

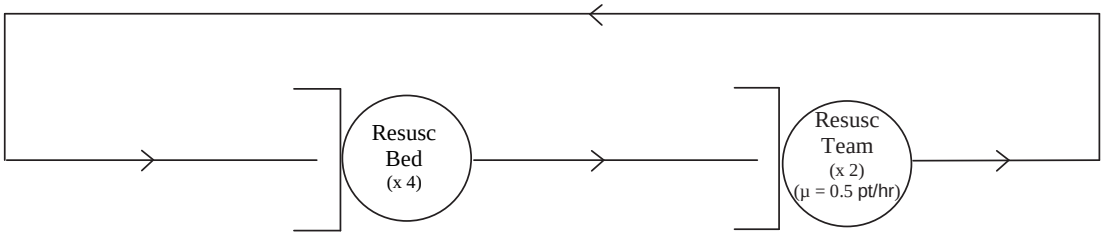


Figure 5.2: Lower-levels of closed queueing network model of patient flow.

5.4 Numerical Results

We compare the mean and standard deviation of patient service time under the three class-based priority schemes and workloads as calculated by our discrete-event simulation and the AGFA technique.

5.4.1 Mean and Standard Deviation of Patient Response Time

First we compare the AGFA and simulation mean and standard deviation of patient service time of the A&E model under full workload. Tables 5.1, 5.2 and 5.3 compare the results for no priority, majors priority and minors priority schemes respectively.

	walk-in arrivals		**ambulance arrivals**		**blue call arrivals**	
no priority	mean	std dev	mean	std dev	mean	std dev
simulation	3.0441	2.7149	2.7952	2.2145	2.0871	2.0425
AGFA	3.5183	3.2703	3.5589	2.9775	2.0917	2.0475

Table 5.1: Mean and standard deviation (std dev) of response times for walk-in, ambulance and blue call arrivals under no priority system, as calculated by the AGFA technique and simulation.

	walk-in arrivals		**ambulance arrivals**		**blue call arrivals**	
majors priority	mean	std dev	mean	std dev	mean	std dev
simulation	4.4391	4.6034	3.1934	3.2386	2.0929	2.0430
AGFA	4.2218	4.2426	4.0598	3.4293	2.0915	2.0474

Table 5.2: Mean and standard deviation (std dev) of response times for walk-in, ambulance and blue call arrivals under majors priority system, as calculated by the AGFA technique and simulation.

	walk-in arrivals		**ambulance arrivals**		**blue call arrivals**	
minors priority	mean	std dev	mean	std dev	mean	std dev
simulation	2.3926	2.0271	2.7096	2.2101	2.0868	2.0356
AGFA	2.4418	3.4517	2.7880	4.7862	2.0917	2.0475

Table 5.3: Mean and standard deviation (std dev) of response times for walk-in, ambulance and blue call arrivals under minors priority system, as calculated by the AGFA technique and simulation.

From Tables 5.1, 5.2 and 5.3 we can see that the AGFA and simulation results for the blue call arrivals under all priority schemes are very close (within 0.6%) for both the mean and standard deviation, as would be expected since the resusc system is essentially

an $M/M/1$ queue with no class-based priorities. Under a no priority scheme (see Table 5.1) we see that for the walk-in and ambulance arrivals the agreement between the simulation and AGFA is relatively poor with the means disagreeing by 15.6% for walk-in and 27.3% for ambulance arrivals. As expected the standard deviations are even further out, by 30.5% for walk-in and 34.5% for ambulance arrivals. When we look at the performance under majors priority (shown in Table 5.2) we can see that the agreement for walk-in arrival mean and standard deviation are good (within 4.9% and 7.8% respectively), while the ambulance arrival mean is fairly poor disagreeing by 27.1%; however, the standard deviation shows good agreement (within 5.9%). Finally, Table 5.3 shows that under a minors priority scheme we get good agreement for mean service times for all arrival types with the simulation and AGFA means within 2.1% for the walk-in arrivals and 2.9% for ambulance arrivals. However, the standard deviation agreement is very poor with the AGFA standard deviations much higher than that obtained by simulation (disagreeing by over 100% for the ambulance arrivals).

5.4.2 Workload Variations

In order to try and understand the circumstances under which the AGFA technique gives good agreement with our simulation, we vary the workload to the system to investigate how the level of system load affects our results. We vary the walk-in and ambulance arrivals from 25% to 95% of full workload whilst keeping the blue call arrivals constant. Since they effectively make up a separate system, further results for blue call arrivals are not presented.

Tables 5.4, 5.5, and 5.6 show the mean and standard deviation of patient response time for a selection of workloads as calculated using the AGFA technique and the corresponding simulation results for walk-in and ambulance arrivals. The full AGFA tables showing all the results for workloads (ranging from 25% to 100% of full workloadc) are shown in Section D.2 of Appendix D. Figs. 5.3, 5.4 and 5.5 display graphically the AGFA and simulation results shown in these tables.

Figs. 5.3, 5.4 and 5.5 illustrate clearly the loading levels at which simulation and AGFA patient response times start to disagree as the workload increases and the department

	no priority							
	walk-in arrivals				ambulance arrivals			
	AGFA		sim		AGFA		sim	
load	mean	std dev	mean	std dev	mean	std dev	mean	std dev
0.25	1.5148	1.5517	1.5369	1.3862	1.9238	1.7725	1.9721	1.4616
0.5	1.5878	1.6130	1.6053	1.4312	1.9881	1.8300	2.0331	1.5000
0.75	1.7910	1.7855	1.8030	1.5632	2.1456	1.9822	2.1848	1.6001
0.85	1.9818	1.9438	1.9967	1.7024	2.2720	2.1077	2.3093	1.6892
0.95	2.5461	2.4498	2.4569	2.1020	2.6717	2.4424	2.5433	1.8979

Table 5.4: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under no priority for varying workloads as calculated using the AGFA technique and simulation.

	majors priority							
	walk-in arrivals				ambulance arrivals			
	AGFA		sim		AGFA		sim	
load	mean	std dev	mean	std dev	mean	std dev	mean	std dev
0.25	1.5151	1.5517	1.5367	1.3876	1.9231	1.7716	1.9712	1.4617
0.5	1.5905	1.6136	1.6080	1.4355	1.9815	1.8202	2.0278	1.4948
0.75	1.8116	1.7992	1.8349	1.6082	2.1178	1.9373	2.1704	1.5845
0.85	2.0518	2.0048	2.1005	1.8453	2.2405	2.0459	2.3003	1.6926
0.95	3.1581	3.0696	2.9197	2.7274	3.0824	2.5904	2.6338	2.1283

Table 5.5: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under majors priority for varying workloads as calculated using the AGFA technique and simulation.

	minors priority							
	walk-in arrivals				ambulance arrivals			
	AGFA		sim		AGFA		sim	
load	mean	std dev	mean	std dev	mean	std dev	mean	std dev
0.25	1.5145	1.5517	1.5367	1.3876	1.9244	1.7734	1.9712	1.4617
0.5	1.5850	1.6125	1.6031	1.4322	1.9951	1.8412	2.0382	1.5088
0.75	1.7718	1.7910	1.7798	1.5590	2.1787	2.0648	2.2074	1.6557
0.85	1.9259	1.9701	1.9247	1.6667	2.3243	2.3185	2.3370	1.7860
0.95	2.2004	2.4935	2.1788	1.8628	2.5723	3.1652	2.5466	2.0223

Table 5.6: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under minors priority for varying workloads as calculated using the AGFA technique and simulation.

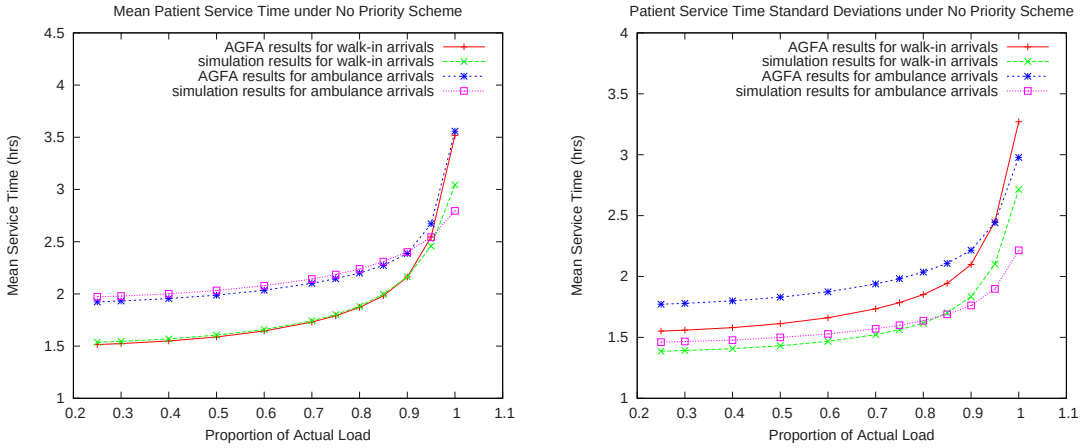


Figure 5.3: AGFA and simulation mean (left) and standard deviation (right) of the service time for walk-in and ambulance arrivals for differing workloads under the no priority system.

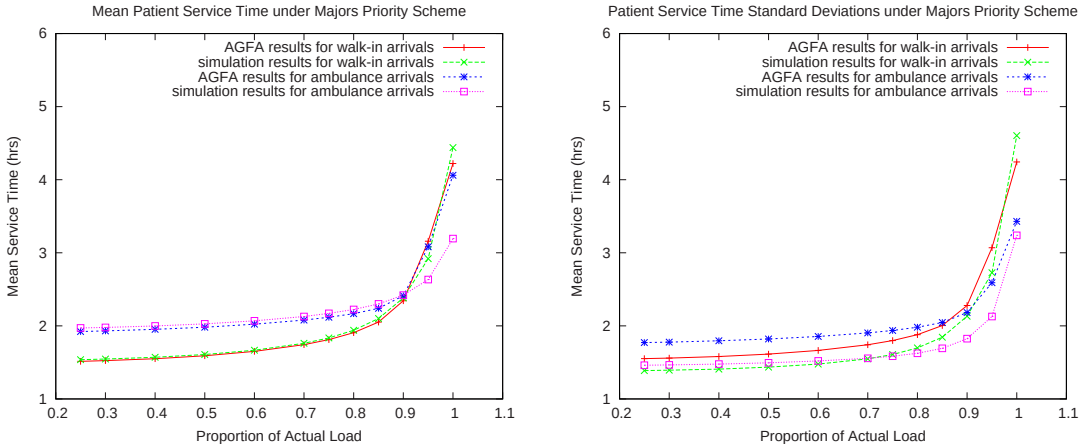


Figure 5.4: AGFA and simulation mean (left) and standard deviation (right) of the service time for walk-in and ambulance arrivals for differing workloads under the majors priority system.

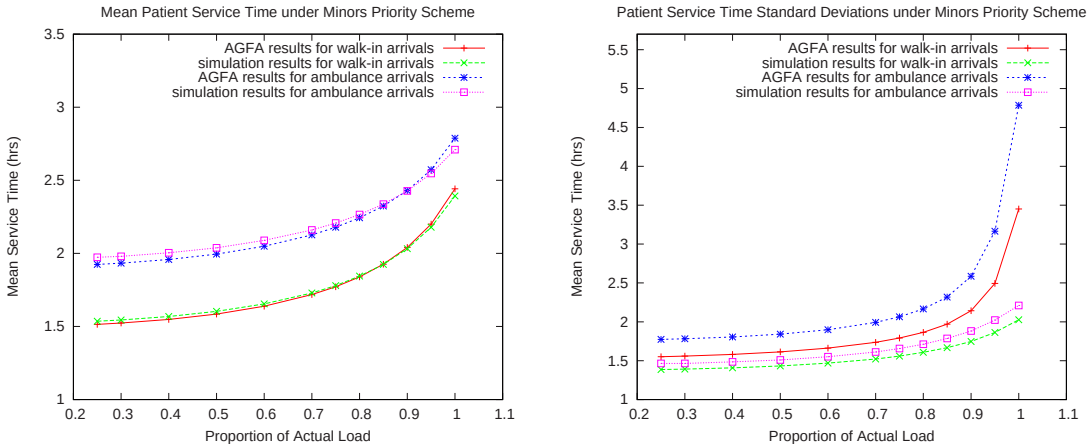


Figure 5.5: AGFA and simulation mean (left) and standard deviation (right) of the service time for walk-in and ambulance arrivals for differing workloads under the minors priority system.

approaches saturation. We can see from Figs. 5.3 and 5.4 that under no priority and majors priority schemes the agreement between the simulation and AGFA means are good up to 90% load whereupon they start to diverge. Under minors priority shown in Fig. 5.5, there is good agreement for mean service times up to full workload, which may be due to the system being less saturated under a minors priority system. For the standard deviations we can see that the AGFA and simulation agreement is good for all system loads under the majors priority scheme. Under the no priority and minor priority schemes we see closer agreement under lower loads (up to 95% and 90% of load for no priority and minors priority schemes respectively) but a big divergence in the standard deviations as the system reaches full load.

5.5 Conclusion

From our results, we see that the mean values obtained via the AGFA technique show good agreement with our simulation results, especially under minors priority and for workloads up to 90% under the other priority systems. It seems that the mean values diverge as the system becomes more highly utilised, as is illustrated by the results under high workloads where patients remain in the department for longer, resulting in greater saturation. It is well known that both approximate analytical methods and simulations tend to suffer from loss of accuracy in saturated systems.

As expected, the AGFA results for the standard deviations are generally not as good a match against the simulation. This is partly because, although aggregation can be shown to preserve many expected values of random variables associated with the queueing processes concerned, the same cannot be said for higher moments. Furthermore, the approximations pointed out in the AGFA analysis of Section 5.2 become more significant at higher moments.

In terms of run times, each simulation run required approximately 20–60 minutes wall clock time (depending on the workload parameters, priority scheme and the PC cluster workstation used), with results for each priority scheme and workload combination being averaged over 10 runs to obtain confidence intervals on the means. By contrast, AGFA required between 30 seconds and 20 minutes (in the saturated majors priority

case) wall clock time for each (single-run) priority scheme and workload combination.

Although still needing improvement in terms of the standard deviation approximation, the good agreement with simulation for the mean values is encouraging. Another advantage of the AGFA technique is that once the approximate Laplace transform has been calculated, all higher moments may be subsequently derived.

Chapter 6

Conclusion

6.1 Summary of Thesis Achievements

This thesis has presented techniques and tools to characterise and forecast patient arrivals, to model patient flow, and to assess the response-time impact of different resource allocations, patient treatment schemes and workload scenarios. We have also presented an efficient approximate generating function analysis (AGFA) technique for determining moments of customer response time in networks of multiclass queues with blocking and class-dependent priorities.

6.1.1 Time Series Models of Patient Arrivals

Being able to predict future arrivals into an A&E department is an important tool for hospital managers. In Chapter 3 we applied time series analysis to model and forecast A&E patient arrivals. We found that walk-in and ambulance patient arrivals have very different characteristics; this may be because walk-in arrivals (the majority of which will have minor illnesses/injuries) have more of a choice when deciding the most convenient time to go into A&E, while ambulance arrivals will tend to call an ambulance as and when needed (due to the more serious nature of their illness/injury). This observation is consistent with other studies [93, 29]. Thus these two arrival streams require to be modelled separately.

We demonstrated that the walk-in arrivals exhibit a strong seven day seasonality that is best modelled with a structural time series model. A structural time series model provides one to six day ahead forecasts with good predictive power. However, we had less success with our ambulance arrivals models. This may be because the ambulance arrivals do not exhibit any strong periodicies or other regularity. Thus the ambulance arrival stream might not be appropriate for this method of time series analysis. We have also investigated characterising ambulance arrivals by a linear non-homogeneous Poisson process [66, 72] but we found that our "training" data fails the corresponding goodness of fit tests. Despite being only able to characterise and forecast walk-in arrivals effectively, these forecasts will still be of value to hospital managers as walk-in arrivals typically account for the majority of arrivals into an A&E department.

We have also demonstrated that arrivals into an A&E department by hour varies predictably, with weekdays exhibiting similar hourly arrival patterns as do weekends. This hourly breakdown of daily arrivals will be useful to hospital managers when deciding the staffing and resource levels required throughout the day as well as designing workshift and handover patterns that coincide with less busy periods of the day. Finally, a study of the impact of weather-related factors – including daily temperature and rainfall, on the number of patient arrivals to our case study department, found no significant relationships. This is possibly because of a lack of extreme weather conditions in this country.

6.1.2 Patient Flow Modelling

A key contribution of this thesis is the development of a detailed model of patient flow in an A&E department, parameterised using actual patient data, as presented in Chapter 4. This model facilitates the calculation of sophisticated performance measures including the higher moments and the densities of patient service time in addition to performance measures at the individual resource level.

The multiclass queueing network model of patient flow was implemented in a discrete-event simulation written in Java. The simulation results and service time densities were compared with those observed in the actual A&E. Having investigated the effects of

different patient priority schemes, we found the impact of the introduction of the 4 hour waiting time target has been similar to a move from a system in which majors patients are given priority, to a system in which minors patients are given priority treatment. This (seemingly socially unacceptable) prioritisation of treatment for minors patients over majors patients leads to the counter-intuitive outcome that mean service times for ambulance arrivals are not adversely affected (in fact they are slightly improved), while mean service times (and corresponding variances) for walk-in arrivals are dramatically lower. This is a particularly interesting result in light of UK government waiting time targets, which has led to the prioritisation of minors patients.

We also gained some insights into how the system behaves when the workload and resource levels are varied. We found that for low to medium workloads, mean service times for ambulance arrivals benefit from a system under majors priority, but under high workloads both arrival types perform better under a minors priority scheme. The main bottleneck in the system for both arrival types was found to be the other specialist followed by the minors practitioner for the walk-in arrivals and radiology for the ambulance arrivals.

6.1.3 Efficient Approximate Response Time Analysis

Whilst seeking an efficient analytical method – that avoids the state-space explosion problem – to solve our model of patient flow we developed the approximate generating function analysis (AGFA) technique presented in Chapter 5. In this technique single nodes of networks of multiclass $M/M/m$ queues with blocking and class-dependent priorities are analysed individually, adapting Cobham's formula to approximate the Laplace transform of response time probability density. The single nodes are aggregated together to form sub-networks, which are solved using a hierarchical MVA-like approach [92, 44].

This technique was applied to a closed network adaptation of the model of patient flow and the results compared against those obtained via the discrete-event simulation. We showed that the technique works well for mean response times under a number of different priority schemes although discrepancies were noted when the system starts

to become saturated under high workloads. The corresponding standard deviations – equivalent to second moments – show generally adequate agreement but were (not atypically) less accurate. This is because higher moments lack the linearity properties of first moments (means) and so greater care and precision is required in their analysis. Although the AGFA method provides this well in open queues, its approximation becomes worse when it is applied in closed systems with constrained class populations at individual nodes.

Although still needing improvement in terms of the standard deviation approximation, the good agreement with simulation for the mean values is encouraging. Another advantage of the AGFA technique is that once the approximate Laplace transform has been calculated, all higher moments may be subsequently derived.

6.2 Applications

The techniques and methods used in this thesis to model and forecast patient arrivals modelling may be easily applied to not only other A&E departments, but also to a range of other healthcare systems, including ambulance services, hospital admissions and out-patient clinics. Similarly the model and discrete-event simulation of our case study department patient flow can be easily adapted to other A&E departments and other healthcare systems where response time targets are in place. Another interesting application of the patient flow model could be to adapt it to model disease progression. With appropriate quantities of data, probability distributions of the time to the next stage of disease can be fitted and combined to form a complete model of disease pathways. In this way useful disease progression time quantiles such as: 95% of patients over 60 with n-stage diabetes on a certain medication will need cataract surgery in x number of years, can be obtained.

As mentioned in the previous section, the AGFA technique can be applied to any multiclass $M/M/m$ queueing system with priorities and blocking to obtain the higher moments of response time. Another application of this technique is to optimise complex queueing network models where the mean and standard deviations of customer response time is optimised by finding the optimal resource allocation. Any optimisation would in

general involve applying an optimisation algorithm in which various resource allocations are investigated. The AGFA technique would be employed at this point to find the impact of these resource allocations; depending on the results obtained, a new set resource allocations are returned by the optimisation algorithm and the subsequent impact again investigated via the AGFA method. This process is repeated until an optimal set of resource allocations is found. A prime example would be to find the optimal staff and resource mix in order to minimise the mean and standard deviation of patient treatment times in the case study A&E model (which we will discuss further in the next section).

6.3 Future Work

Future work may involve identifying improvements and possible extensions to both the patient arrival and patient flow models. The AGFA technique can also be refined and the accuracy at the higher moments improved. Another future area of research could be the use of the patient flow model and the AGFA technique to perform optimisation.

The patient arrivals models can be improved by further characterising ambulance arrivals using other methods, possibly including non-homogeneous Poisson processes with cyclic or periodic behaviour [65, 66]. Better models for walk-in arrivals can be fitted as we gather more data; this may involve incorporating both a weekly and an annual seasonality by fitting a multi-seasonal structural time series model [49] and including calendar events such bank holidays into our models.

As mentioned in Section 4.3.5 there are a number of complexities not incorporated into our patient flow model. As future work the main additions/changes we believe would make the biggest improvements to the model accuracy are:

Staff and Resource Service Time Distributions Given sufficient staff and resource timing data, we can calculate the first four moments of service time. It has been shown that by fitting a *Generalised Lamda Distribution* [45, 61] to the first four moments of a probability distribution, a good approximation to the actual probability

distribution can be obtained [7, 8]. In this way we hope to obtain much more accurate service time distributions with which to parameterise our patient flow model.

Bed Blocking As mentioned in Section 4.11, the disparity between the actual and simulated ambulance arrival service time means may be due to the lack of blocking phenomena in our model, which will mostly delay ambulance arrivals. This will require incorporating the various different wards in the hospital plus the numbers of beds available in each. The rate of admission (both from A&E and other departments) and discharge to and from each ward will also have to be approximated.

Parallel Tests and Scans It is important to model the laboratory and radiology area accurately because a large number of patients are routed through these nodes; consequently, as we have shown, they are highly utilised with radiology being one of the bottlenecks in the system. Overlapping testing and scanning can be incorporated into the model by using a fork-join queue [17, 70].

Patient Arrivals As shown in Chapter 3, the arrivals process at an A&E department is non-stationary and is inadequately modelled by a Poisson arrivals stream. Although we have shown that our patient arrival models can be used to effectively characterise patient arrivals by hour and by day, we decided not to incorporate the patient arrival models into our model of patient flow for two reasons. Firstly this was done for ease of implementation and secondly to facilitate the application of the AGFA technique to our model of patient flow. However, if we are only looking to simply improve the results of our discrete-event simulation, incorporating realistic patient arrivals would be a first step.

Future work should also involve refining the AGFA technique in order to get better agreement at greater loads and higher moments. Other improvement to this technique involve adapting it to incorporate more complex queueing disciplines such as time-based queueing priorities (i.e. queues with ageing).

Finally, as mentioned in the previous section, another promising line of future research could be to utilise both the AGFA technique and our detailed patient flow model to

optimise our case study A&E department, whereby the optimal staff and resource mix is found in order to minimise the mean and standard deviation of patient service times. Optimisation has been widely used in the healthcare sector, generally to find the optimum allocation of resources and to optimise staff scheduling in terms of cost [83, 40, 22, 94]. Currently the AGFA technique when applied to the full patient flow model is too slow to facilitate any optimisation in reasonable time. This is due to optimisation methods potentially requiring the application of the AGFA technique hundreds if not thousands of times. A first step in any optimisation would therefore be to first further simplify the model of patient flow to make any optimisation tractable. Once this has been achieved, the optimisation to be made can be cast into a non-linear integer programming problem; a number of optimisation techniques may then be applied to obtain optimal values. These optimal values can then be verified using the discrete-event simulation of the full model and the corresponding response time densities obtained.

Glossary of Medical Terms and Abbreviations

Notation	Description
A&E	Accident and Emergency.
Ambulance arrivals	Patients that come into A&E via ambulance.
Blue call arrivals	Very seriously ill/injured patients that require resuscitation.
CCU	Coronary Care Unit – a specialist unit that specialises in cardiac conditions.
CDU	Clinical Decision Unit – short stay ward for patients that require observation.
COPD	Chronic Obstructive Pulmonary Disease.
CT scan	Computerised Tomography scan – medical imaging method.
DOA	Dead On Arrival.
ENT	Ear, Nose and Throat.
GP	General Practitioner.
GP-referred arrivals	Patients that come into A&E after first consulting a GP.
ITU	Intensive Treatment Unit.
Majors patients	Patients with major illness or injury.
MAU	Medical Assessment Unit – assesses if a patient requires admission to a medical ward.
Minors patients	Patients with minor illness or injury.

Notation	Description
Minors Practitioner	Doctor or nurse trained to assess and treat minor illnesses or injuries.
Radiology	Department where all scans such as x-rays are performed.
Resusc	Resuscitation.
Resuscitation patients	Patients that require urgent medical attention.
Review clinic	Consultant run follow-up clinic for patients who were not admitted.
Self-referred arrivals	Patients who come into A&E of their own accord and transport.
Suspended patient	Resuscitation patient that shows no vital signs.
Theatre	Operating theatre where surgery is performed.
Walk-in arrivals	Patients that come into A&E via their own transport.
Ward	Hospital room or block with beds for patients that require similar care.

Appendix A

R Code for Fitting Time Series Models

A.1 Rolling Average Models

```
#When using R, any sentence preceded by  # indicates a comment.

#read in all the data in this case we use the daily ambulance arrivals data
ambulance_data <- scan("amb_02-07.txt", list("",""))

#set data as a time series with 7 day frequency
series<- ts(as.numeric(ambulance_data[[2]]),frequency =7)

#take first 1456 values use last 370 to evaluate predictions
npred <- 1456

#truncate the data to get just the training data
training_data <- ts(series[1:npred],frequency=7)

#keep the rest of the data as the unseen data
unseen_data<-ts(series[(npred+1):length(series)], frequency =7)

#fit a rolling average model to the training data
series_RA_fit <- data.frame()
series_RA_current<- data.frame()

#6 week rolling average fit
fit_index <- (6*7+1): length(training_data)

for (i in fit_index)

{

        #find the current rolling average (RA) value
        series_RA_current<- (training_data[(i-7)]/6 + training_data[(i-14)]/6 + training_data[(i-21)]/6
        + training_data[(i-28)]/6 + training_data[(i-36)]/6 +training_data[(i-42)]/6)

        #save RA value
        series_RA_fit <- c(series_RA_fit,as.numeric(series_RA_current))

}

#create a time series of the RA fits
series_RA_fit<- ts(as.numeric(series_RA_fit), frequency =7 )

#work out the residuals
series_RA_resid <- ts(training_data[43:length(training_data)]-series_RA_fit, frequency=7)

#find the 95% confidence intervals from the residuals
ci<-1.96*sd(series_RA_resid)

#find the correlation between the model fit and the actual data
cor.test(training_data[43:1456],series_RA_fit)

#plot the autocorrelation function for the residuals
acf(series_RA_resid, main="Residual acf for ambulance rolling average model")

#fit a normal distribution to the residuals
x_series<- -60:60
sd_series <- sd(series_RA_resid)
mean_series <- mean(series_RA_resid)
y_series <- dnorm(x_series,mean_series,sd_series)
```

```
#histogram of residuals with associated normal distribution
hist(series_RA_resid,xlab="residuals", main="Residual histogram for ambulance rolling average model", prob=T,
ylim=c(0,0.04), xlim=c(-40,40))
lines(x_series,y_series)

#Ljung-Box test for independence of the residuals. testing over 30 lags
Box.test(series_RA_resid, lag=30, type= "Ljung")

#take off the inital 42 values of the data series that was used to fit the initial model
zero<-rep(NA, times=42)
RA_fit_plot<-ts(c(zero,series_RA_fit), frequency =7)

#plot the model fit with the actual data
plot(training_data, type ='b',ylab="no. of patients",xlab="weeks", main=("ambulance rolling average model fit"))
lines(RA_fit_plot,col="blue")
legend("topleft",c("\"training\" data ","RA model fit"), col= c("black","blue"),bty="n",lty = 1)

#now use this model to predict the unseen data
series_RA_pred <- data.frame()
series_RA_current<- data.frame()

#index of how far along the data we are
pred_index <- (npred+1): length(series)

for (i in pred_index)

{

        #predict ahead by one day
        series_RA_current<- (series[(i-7)]/6 + series[(i-14)]/6 + series[(i-21)]/6 + series[(i-28)]/6
        + series[(i-36)]/6 +series[(i-42)]/6)

        #save RA prediction
        series_RA_pred <- c(series_RA_pred,as.numeric(series_RA_current))

}

#create a time series of the RA forecasts
series_RA_pred<- ts(as.numeric(series_RA_pred), frequency =7 )

#find the percentage of predictions inside the 95% confidence interval
conf <- series_RA_pred[abs(unseen_data - series_RA_pred) > (ci)  ]
length(conf)/length(series_RA_pred) * 100

#find the correlation between the model forecast and the unseen data
cor.test(unseen_data,series_RA_pred)

#calculate the mean bias
bias<- ts(series_RA_pred-unseen_data, frequency = 7)
mean(bias)

#calculate the sum of squares
sum_squares<-0
count <- 1: (length(series_RA_pred))
for (i in count)

{
        sum_squares<- (sum_squares + (series_RA_pred[i] - unseen_data[i])*(series_RA_pred[i] - unseen_data[i]))
}
```

```
#calculate the root mean square error
rmse<- sqrt(sum_squares/length(series_RA_pred))

#plot model forecast
plot(unseen_data, type ='b',ylab="no. of patients",xlab="weeks", main=("ambulance rolling average model prediction"))
lines(series_RA_pred,col="blue")
lines(series_RA_pred - (ci),col='red')
lines(series_RA_pred + (ci),col='red')
legend("topleft",c("\"unseen\" data","RA model predictions","95% confidence interval"),
col= c("black","blue","red"),bty="n",lty = 1)

#create scatterplot
plot(unseen_data, series_RA_pred,ylab="predicted arrivals",xlab="observed arrivals", main="ambulance arrivals",
xlim=c(40,100),ylim=c(40,100))
abline(0,1,lty=2)
```

A.2 Auto-regressive Models

```
#read in all the data here we use the daily walk-in arrivals data
series_data <- scan("walk_in_02-07.txt", list("",""))

#set data as a time series
series_ts<- ts(as.numeric(series_data[[2]]),frequency =1)

#keep the last 370 data points as the unseen data
actual_unseen_data <- ts( series_ts[1457:(length(series_ts))],frequency = 7)

#test time series for stationarity
kpss.test(series_ts[1:1456])

#not stationary, take the first difference to create a differenced series
diff_series <- diff(series_ts)

series<- ts(diff_series, frequency = 1)

#test differenced time series for stationarity
kpss.test(series)

#take first 1455 differences use last 370 to evaluate predictions
npred <- 1455

#truncate the data to get just the training data
training_data <- ts(series[1:npred],frequency=7)

#keep the rest of the differences as the unseen difference data
unseen_data <- ts(series[(npred+1):(length(series))], frequency = 7)

#fit a auto-regressive (AR) model to the training data
series_ar <- ar(training_data,aic=T,method="yw",var.method=2)

#remember the order of the model fit
order<-series_ar$order

#AR model fit
series_fitted<-ts(training_data[-c(1:series_ar$order)]-series_ar$resid[-c(1:series_ar$order)],frequency=1)

#take off the values of the data series that was used to fit the initial model
training_data_fit<-ts(training_data[-c(1:series_ar$order)],frequency = 1)

#plot the AR model fit
plot(ts(training_data[-c(1:series_ar$order)],frequency = 7), main="differenced walk-in auto-regressive model fit",
ylab="no. of patients", xlab="weeks")
lines(ts(training_data[-c(1:series_ar$order)]-series_ar$resid[-c(1:series_ar$order)], frequency=7),col="blue")
legend("topleft",c("differenced \"training\" data ","AR model fit"), col= c("black","blue"),bty="n",lty = 1)

#find the correlation between the model fit and the differenced data
cor.test(training_data, fit,series_fitted)

#plot the autocorrelation function for the residuals
acf(training_data, main="acf for differenced walk-in arrivals")

#fit a normal distribution to the residuals
x_series<- -60:60
sd_series <- sd(series_ar$resid,na.rm=T)
mean_series <- mean(series_ar$resid,na.rm=T)
y_series <- dnorm(x_series,mean_series,sd_series)
```

```
#histogram of residuals with associated normal distribution
hist(series_ar$resid,xlab="residuals", main="Residual histogram for walk-ins", prob=T, ylim=c(0,0.03), xlim=c(-60,60))
lines(x_series,y_series)

#plot the residuals against time
t<-(order+1):npred
plot(t, series_ar$resid[(order+1):npred], type="b", main="Residual plot for walk-in difference model fit against time",
xlab="time", ylab="residuals")

#plot the residuals against fitted values
plot(series_fitted,series,ar$resid[(order+1):npred], main="Residual plot for walk-in difference model fit", xlab="fitted
value", ylab="residuals")


#Ljung-Box test for independence of the residuals. testing over 30 lags
Box.test(series_ar$resid[-c(1:series_ar$order)], lag=30, type= "Ljung")

#calculate one day ahead differenced predictions
series.pred <- data.frame()
series.se <- data.frame()

#index of how far along the data we are
index <- 1: (length(series)-npred)
for (i in index)
{
        #work out data to fit model with
        current_time<-ts(series[i:(npred+i-1)], frequency = 1)
        series_ar <- ar(current_time,aic=F, order.max=order,method="yw",var.method=2)

        # make prediction
        model.pred <- predict(series_ar,n.ahead=1)

        #remember predicted values
        series.pred <- c(series.pred,as.numeric(model.pred$pred))

        # remember standard error of predicted values
        series.se <- c(series.se,as.numeric(model.pred$se))
}

#create a time series of the AR forecast
series.pred  <- ts(as.numeric(series.pred[1:(length(series)-npred)]), frequency =7)

#create a time series of standard error in the ST forecast used to find the 95% confidence interval
series.se  <- ts(as.numeric(series.se[1:(length(series)-npred)]), frequency =7)

#calculate the AR model predictions of the arrivals using the predicted differences
series.actual.pred  <- ts(series_ts[(npred+1):(length(series))] + series.pred, frequency =7)

#find the percentage of predictions inside the 95% confidence interval
c95 <- 1.96
conf <- series.pred[abs(unseen_data - series.pred) > (c95 * series.se)  ]
length(conf)/length(unseen_data) * 100

#forecast measures for difference predictions

#find the correlation between the predicted differences and the differenced actual data
cor.test(unseen_data,series.pred)

#calculate the mean 95% confidence interval
```

```
#calculate the mean bias of the difference predictions
bias<- ts(series.pred-unseen_data, frequency = 1)
mean(bias)

#calculate the sum of squares
sum_squares<-0

count <- 1: (length(series.pred))
for (i in count)
{
        sum_squares<- (sum_squares + (series.pred[i] - unseen_data[i])*(series.pred[i] - unseen_data[i]))
}

#calculate the root mean square error
rmse<- sqrt(sum_squares/length(series.pred))

#plot model forecast of differences
plot (unseen_data,type='b',xlab='weeks',ylab='no. of patients', main = 'differenced walk-in autoregressive model difference
predictions')
lines(series.pred,col='blue')
lines(series.pred - (c95 * series.se),col='red')
lines(series.pred + (c95 * series.se),col='red')
legend("topleft",c("differenced\"unseen\" data","AR model difference predictions","95% confidence interval"), col=
c("black","blue","red"),bty="n",lty = 1)

#create scatterplot of the differences
plot(unseen_data,series.pred,xlab='observed differences',ylab='predicted differences',main='walk-in arrival differences',
ylim=c(-80,80), xlim=c(-80,80))
abline(0,1,lty=2)

#forecast measures for arrival predictions

#find the correlation between the predicted arrivals and the actual data
cor.test(actual_unseen_data,series.actual.pred)

#calculate the mean bias of the arrival predictions
bias<- ts(series.actual.pred-actual_unseen_data, frequency = 1)
mean(bias)

#calculate the sum of squares
sum_squares<-0
count <- 1: (length(series.actual.pred))
for (i in count)
{
        sum_squares<- (sum_squares + (series.actual.pred[i] - actual_unseen_data[i])*(series.actual.pred[i]
        - actual_unseen_data[i]))
}

#calculate the root mean square error
rmse<- sqrt(sum_squares/length(series.actual.pred))

#plot model forecast of arrivals
plot (actual_unseen_data,type='b',xlab='weeks',ylab='no, of patients', main = 'walk-in autoregresive model predictions')
lines(series.actual.pred,col='blue')
lines(series.actual.pred - (c95 * series.se),col='red')
lines(series.actual.pred + (c95 * series.se),col='red')
legend("topleft",c("\"unseen\" data","AR model predictions","95% confidence interval"),
col= c("black","blue","red"),bty="n",lty = 1)
```

```
#create scatterplot of arrival predictions
plot(actual_unseen_data,series.actual.pred,xlab='observed arrivals',ylab='predicted arrivals',main='walk-in arrivals',
ylim=c(135,305), xlim=c(135,305))
abline(0,1,lty=2)

#calculate predictions for more than one day ahead
#current lag value (varies from 2 to 7)
lag<-6

lag_series.pred <- data.frame()
lag_series.se <- data.frame()
lag_actual_series.pred <- data.frame()
lag_only_actual_series.pred <- data.frame()
lag_only_actual_unseen_data <- data.frame()
lag_only_unseen_data <- data.frame()
lag_only_series.pred <- data.frame()

#index of how far along the data we are
lag_index <- 0:((length(series)-npred)%/%lag)
for (j in lag_index)

{
        #work out data to fit model to
        i=j*lag
        current_time<-ts(series[(i+1):(npred+i)], frequency = 1)

        series_ar <- ar(current_time,aic=F, order.max=series_ar$order,method="yw",var.method=2)

        # make differenced prediction for the lag value days ahead
        lag_model.pred <- predict(series_ar,n.ahead=lag)

        # remember differenced values of the lag difference
        lag_series.pred <- c(lag_series.pred,as.numeric(lag_model.pred$pred[lag]))

        # remember standard error of predicted differences for the lag difference
        lag_series.se <- c(lag_series.se,as.numeric(lag_model.pred$se[lag]))

        #work out the corresponding difference in the unseen data
        lag_only_unseen_data<-c(lag_only_unseen_data, as.numeric(series[(npred+i+lag)]))

        #work out actual preditions

        act_index<-1:lag
        current_sum<-0

        for (k in act_index)
        {
                current_sum<- current_sum+as.numeric(lag_model.pred$pred[k])

                lag_actual_series.pred<-c(lag_actual_series.pred, (as.numeric(current_sum) + series_ts[npred+1+i]))
        }

      # remember predicted arrival for the lag value day
    lag_only_actual_series.pred<-c(lag_only_actual_series.pred, as.numeric(lag_actual_series.pred[i+lag]))

      #work out the corresponding day in the unseen data of acutal arrivals
    lag_only_actual_unseen_data<-c(lag_only_actual_unseen_data, as.numeric(series_ts[(npred+1+i+lag)]))
}
```

```
#create a time series of the AR forecast standard errors
lag_series.se  <- ts(as.numeric(lag_series.se[1:((length(series)-npred)%/%lag)]), frequency =1)

#create a time series of the corresponding days in the unseen differenced data
lag_only_actual_unseen_data<-ts(as.numeric(lag_only_actual_unseen_data[1:((length(series)-npred)%/%lag)]),
frequency=1)

#create a time series of the AR arrivals forecast
lag_only_actual_series.pred<-ts(as.numeric(lag_only_actual_series.pred[1:((length(series)-npred)%/%lag)]), frequency=1)

#find the percentage of predictions inside the 95% confidence interval
c95 <- 1.96
conf <- lag_only_actual_series.pred[abs(lag_only_actual_unseen_data - lag_only_actual_series.pred) > (c95 *
lag_series.se)  ]
length(conf)/length(lag_only_actual_unseen_data) * 100

#forecast measures for actual arrivals

#find the correlation between the predicted arrivals and the actual data
cor.test(lag_only_actual_unseen_data,lag_only_actual_series.pred)

#calculate the mean 95% confidence interval
mean(c95 * lag_series.se)

#calculate the mean bias of the arrivals predictions
lag_bias<- ts(lag_only_actual_series.pred-lag_only_actual_unseen_data, frequency =1)
mean(lag_bias)

#calculate the sum of squares
sum_squares<-0

count <- 1:(length(lag_only_actual_series.pred))
for (i in count)
{
        sum_squares<- (sum_squares + (lag_only_actual_series.pred[i] - lag_only_actual_unseen_data[i])^2)
}

#calculate the root mean square error
rmse<- sqrt(sum_squares/length(lag_only_actual_series.pred))

#calculate entire week ahead predictions
week_series.pred <- data.frame()
week_series.se <- data.frame()
week_actual_series.pred <- data.frame()
week_only_actual_series.pred <- data.frame()

#index of how far along the data we are
lag_index <- 0:52
for (j in lag_index)

{
        #work out data to fit model to
        i=j*7
        current_time<-ts(series[(i+1):(npred+i)], frequency = 1)

        series_ar <- ar(current_time,aic=F, order.max=series_ar$order,method="yw",var.method=2)

        # make differenced prediction for one entire week ahead
        week_model.pred <- predict(series_ar,n.ahead=7)
```

```
        # remember differenced values for the week
        week_series.pred <- c(week_series.pred,as.numeric(week_model.pred$pred))

        # remember standard error of predicted differences for the week
        week_series.se <- c(week_series.se,as.numeric(week_model.pred$se))

        #work out actual preditions
        act_index<-1:7
        current_sum<-0

        for (k in act_index)
        {
                current_sum<- current_sum+as.numeric(week_model.pred$pred[k])

                week_actual_series.pred<-c(week_actual_series.pred, (as.numeric(current_sum) + series_ts[npred+1+i]))
        }
}

#create a time series of the AR forecast differences
week_series.pred  <- ts(as.numeric(week_series.pred[1:(length(series)-npred)]), frequency =7)

#create a time series of the AR forecast standard errors
week_series.se  <- ts(as.numeric(week_series.se[1:(length(series)-npred)]), frequency =7)

#create a time series of the AR arrivals forecastl
week_actual_series.pred<-ts(as.numeric(week_actual_series.pred[1:(length(series)-npred)]), frequency =7)

#find the percentage of predictions inside the 95% confidence interval
c95 <- 1.96
conf <- week_actual_series.pred[abs(actual_unseen_data - week_actual_series.pred) > (c95 * week_series.se)  ]
length(conf)/length(actual_unseen_data) * 100

#forecast measures for actual arrivals

#find the correlation between the predicted arrivals and the actual data
cor.test(actual_unseen_data,week_actual_series.pred)

#calculate the mean 95% confidence interval
mean(c95 * week_series.se)

#calculate the mean bias of the arrivals predictions
week_bias<- ts(week_actual_series.pred-actual_unseen_data, frequency =7)
mean(week_bias)

#calculate the sum of squares
sum_squares<-0

count <- 1:(length(week_actual_series.pred))
for (i in count)
{
        sum_squares<- (sum_squares + (week_actual_series.pred[i] - actual_unseen_data[i])^2)
}

#calculate the root mean square error
rmse<- sqrt(sum_squares/length(week_actual_series.pred))
```

A.3 Structural Time Series Models

```
#read in all the data here we use the daily ambulance arrivals data
series_data <- scan("amb_02-07.txt", list("",""))

#set data as a time series with 7 day frequency
series<- ts(as.numeric(series_data[[2]]),frequency =7)

#take first 1456 values use last 370 to evaluate predictions
npred <- 1456

#truncate the data to get just the training data
training_data <- ts(series[1:npred],frequency=7)

#keep the rest of the data as the unseen data
unseen_data<-ts(series[(npred+1):length(series)], frequency =7)

#fit a ST model to the training data
series_st <- StructTS(training_data,type="BSM")

#ST model fit
series_st_fit <- ts(apply(series_st$fitted[,c(1,3)],1,sum),frequency=7)

#ST model residuals
series_st_resid <- ts(training_data-series_st_fit, frequency=1)

#plot the ST model data fit
plot(training_data,ylab="no. of patients", xlab="weeks",main=("ambulance structural time series model fit"))
lines(series_st_fit,col="blue")
legend("topleft",c("\"training\" data ","ST model fit"), col= c("black","blue"),bty="n",lty = 1)

#find the correlation between the model fit and the actual data
cor.test(training_data,series_st_fit)

#plot the autocorrelation function for the residuals
acf(series_st_resid, main="Residual acf for ambulance arrivals")

#fit a normal distribution to the residuals
x_series<- -60:60
sd_series <- sd(series_st_resid)
mean_series <- mean(series_st_resid)
y_series <- dnorm(x_series,mean_series,sd_series)

#histogram of residuals with associated normal distribution
hist(series_st_resid,xlab="residuals", main="Residual histogram for ambulance arrivals", prob=T, ylim=c(0,0.05),
xlim=c(-30,30))
lines(x_series,y_series)

#plot the residuals against time
t<-1:npred
plot(t, series_st_resid, type="b", main="Residual plot for ambulance model fit against time", xlab="time", ylab="residuals"

#plot the residuals against fitted value
plot(series_st$fit[1:npred], series_st_resid[1:npred],  main="Residual plot for ambulance model fit ", xlab="fitted value",
ylab="residuals")


#Ljung-Box test for independence of the residuals. testing over 30 lags
Box.test(series_st_resid, lag=30, type= "Ljung")

#calculate one day ahead predictions
series.pred <- data.frame()
```

```
#index of how far along the data we are
index <- 1: (length(series)-npred)
for (i in index)

{
        #work out data to fit model with
        current_time<-ts(series[(ntrim+i-1):(npred+i - 1)], frequency = 7)
        series_st <- StructTS(current_time,type = "BSM")

        # make prediction
        model.pred <- predict(series_st,n.ahead=1)

        # remember predicted values
        series.pred <- c(series.pred,as.numeric(model.pred$pred))

        # remember standard error of predicted values
        series.se <- c(series.se,as.numeric(model.pred$se))
}

#create a time series of the ST forecast
series.pred  <- ts(as.numeric(series.pred), frequency =7)

#create a time series of standard error in the ST forecast used to find the 95% confidence interval
series.se  <- ts(as.numeric(series.se), frequency =7)

#find the percentage of predictions inside the 95% confidence interval
c95 <- 1.96
conf <- series.pred[abs(unseen_data - series.pred) > (c95 * series.se)  ]
length(conf)/length(unseen_data) * 100

#find the correlation between the model forecast and the unseen data
cor.test(unseen_data,series.pred)

#calculate the mean 95% confidence interval
mean(c95*series.se)

#calculate the mean bias
bias<- ts(series.pred-unseen_data, frequency = 7)
mean(bias)

#calculate the sum of squares
sum_squares<-0
count <- 1: (length(series.pred))
for (i in count)
{
        sum_squares<- (sum_squares + (series.pred[i] - unseen_data[i])*(series.pred[i] - unseen_data[i]))
}

#calculate the root mean square error
rmse<- sqrt(sum_squares/length(series.pred))

#plot model forecast
plot (unseen_data,type='b',xlab='weeks',ylab='no. of patients', main = 'ambulance structural time series model predictions')
lines(series.pred,col='blue')
lines(series.pred - (c95 * series.se),col='red')
lines(series.pred + (c95 * series.se),col='red')
legend("topleft",c("\"unseen\" data","ST model predictions","95% confidence interval"),
col= c("black","blue","red"),bty="n",lty = 1)
```

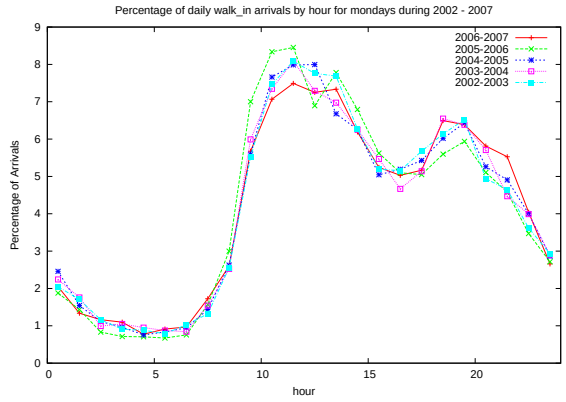
```
#create scatterplot
plot(unseen_data,series.pred,xlab='observed arrivals',ylab='predicted arrivals',main='ambulance arrivals', ylim=c(135,305),
xlim=c(135,305))
abline(0,1,lty=2)

#calculate predictions for more than one day ahead
#current lag value (varies from 2 to 7)
lag<-7

lag_series.pred <- data.frame()
lag_series.se <- data.frame()
lag_future_series<-data.frame()

#index of how far along the data we are
lag_index <- 0:((length(series)-npred)%/%lag -1)
for (j in lag_index)

{
        #work out data to fit model to
        i=j*lag
        current_time<-ts(series[(ntrim+i):(npred+i)], frequency = 7)

        series_st <- StructTS(current_time,type = "BSM")

        # make prediction for the lag value days ahead
        lag_model.pred <- predict(series_st,n.ahead=lag)

        # remember predicted values for the lag value day
        lag_series.pred <- c(lag_series.pred,as.numeric(lag_model.pred$pred[lag]))

        # remember standard error of predicted values
        lag_series.se <- c(lag_series.se,as.numeric(lag_model.pred$se[lag]))

        #work out the corresponding day in the unseen data
        lag_future_series <- c(lag_future_series, series[(npred+i+lag)])
}

#create a time series of the ST forecast
lag_series.pred  <- ts(as.numeric(lag_series.pred[1:(370%/%lag)]))

#create a time series of the standard error of the ST forecast
lag_series.se  <- ts(as.numeric(lag_series.se[1:(370%/%lag)]))

#create a time series of the corresponding days in the unseen data
lag_future_series<-ts(as.numeric(lag_future_series[1:(370%/%lag)]))

#find the percentage of predictions inside the 95% confidence interval
c95 <- 1.96
conf <- lag_series.pred[abs(lag_future_series - lag_series.pred) > (c95 * lag_series.se)  ]
length(conf)/length(lag_future_series) * 100

#find the correlation between the model forecast and the unseen data
cor.test(lag_future_series,lag_series.pred)

#calculate the mean 95% confidence interval
mean(c95 * lag_series.se)

#calculate the mean bias
lag_bias<- ts(lag_series.pred-lag_future_series)
mean(lag_bias)
```

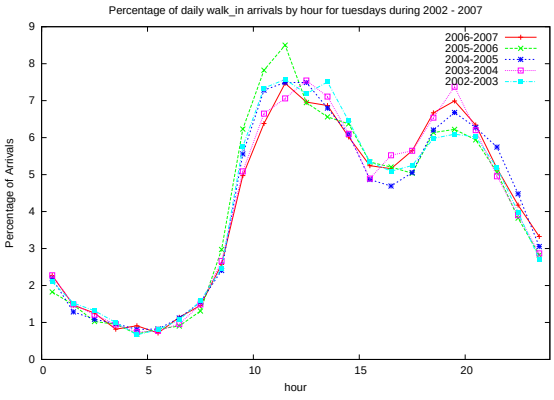
```
#calculate the sum of squares
sum_squares<-0
count <- 1:(length(lag_series.pred))
for (i in count)
{       sum_squares<- (sum_squares + (lag_series.pred[i] - lag_future_series[i])^2) }

#calculate the root mean square error
rmse<- sqrt(sum_squares/length(lag_series.pred))

#calculate entire week ahead predictions
week_series.pred <- data.frame()
week_series.se <- data.frame()
lag_index <- 0:52
for (j in lag_index)
{
        i=j*7
        current_time<-ts(series[(ntrim+i):(npred+i)], frequency = 7)

        series_st <- StructTS(current_time,type = "BSM")

        # make a week ahead prediction
        week_model.pred <- predict(series_st,n.ahead=7)

        # remember predicted values for the entire week ahead and the standard errors
        week_series.pred <- c(week_series.pred,as.numeric(week_model.pred$pred))
        week_series.se <- c(week_series.se,as.numeric(week_model.pred$se))
}

#create a time series of the week ahead ST forecast
week_series.pred  <- ts(as.numeric(week_series.pred[1:370]), frequency = 7)

#create a time series of the standard error of the week ahead ST forecast
week_series.se  <- ts(as.numeric(week_series.se[1:370]), frequency = 7)

#find the percentage of predictions inside the 95% confidence interval
c95 <- 1.96
conf <- week_series.pred[abs(future_series - week_series.pred) > (c95 * week_series.se)  ]
length(conf)/length(future_series) * 100

#find the correlation between the model forecast and the unseen data
cor.test(future_series,week_series.pred)

#calculate the mean 95% confidence interval
mean(c95 * week_series.se)

#calculate the mean bias
week_bias<- ts(week_series.pred-future_series)
mean(week_bias)

#calculate the sum of squares
sum_squares<-0
count <- 1:(length(week_series.pred))
for (i in count)
{
        sum_squares<- (sum_squares + (week_series.pred[i] - future_series[i])^2)
}

#calculate the root mean square error
rmse<- sqrt(sum_squares/length(week_series.pred))
```

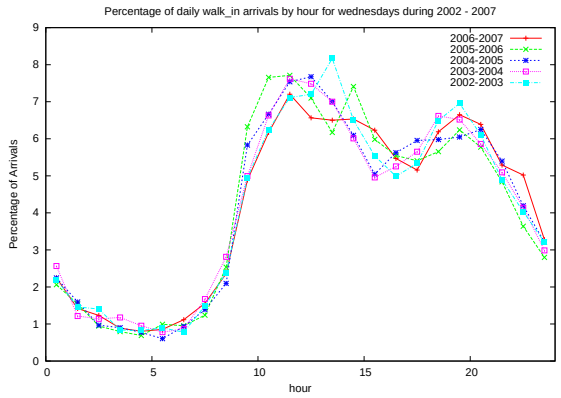
Appendix B

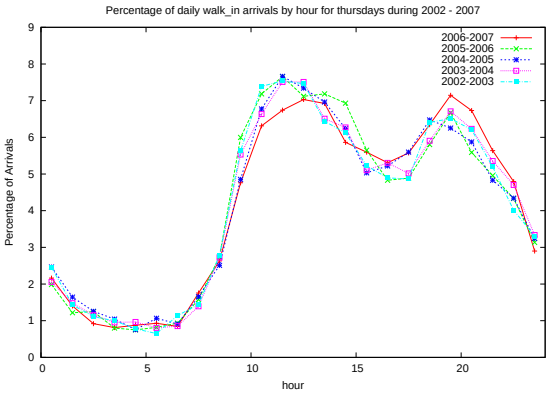
Hourly Patient Arrivals for Each Day of Week

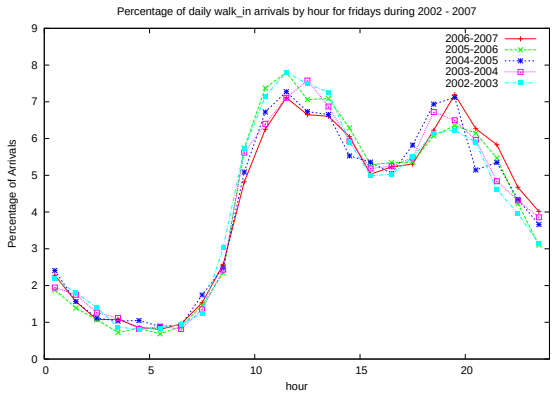
B.1 Walk-In Arrivals

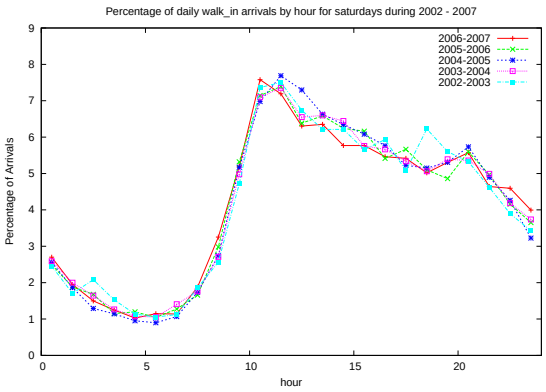


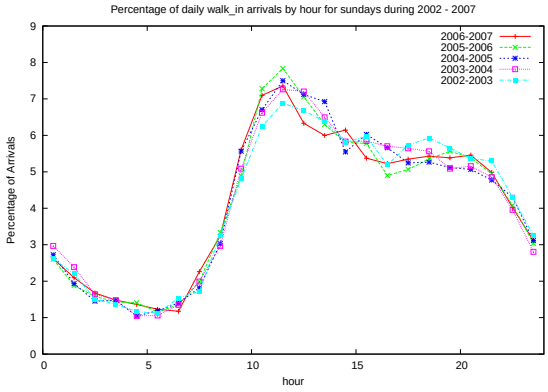




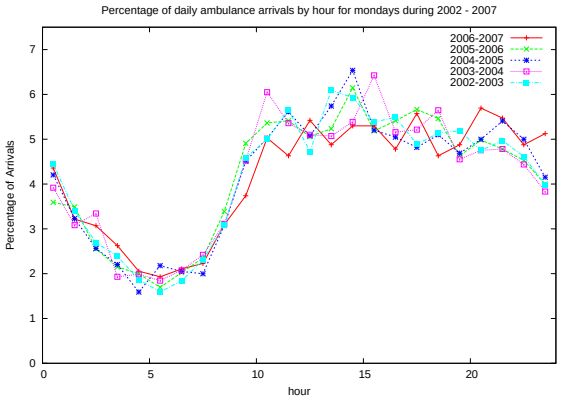


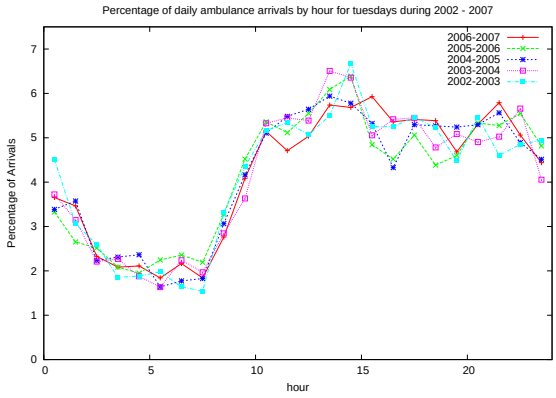


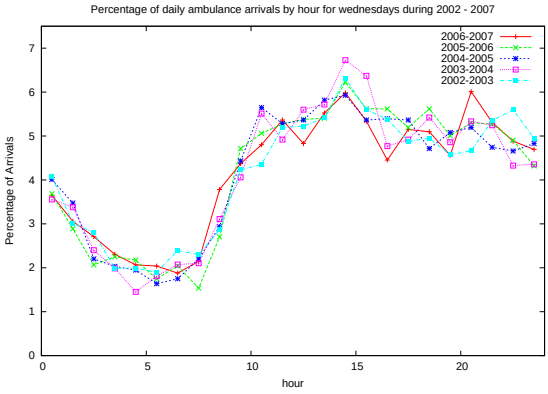


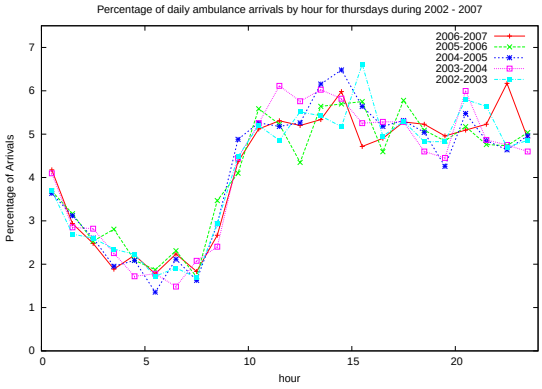


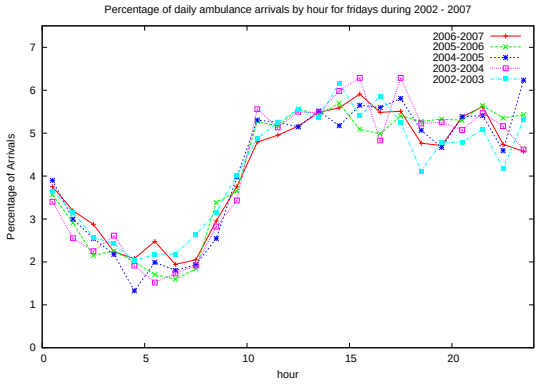
B.2 Ambulance Arrivals

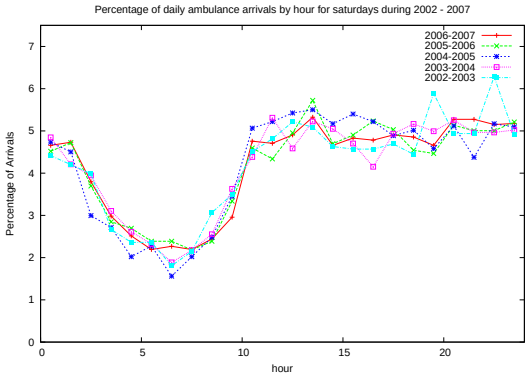


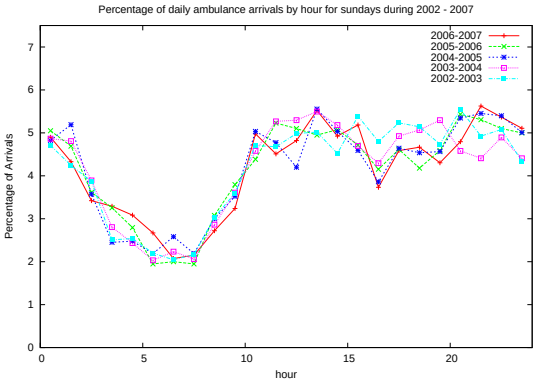








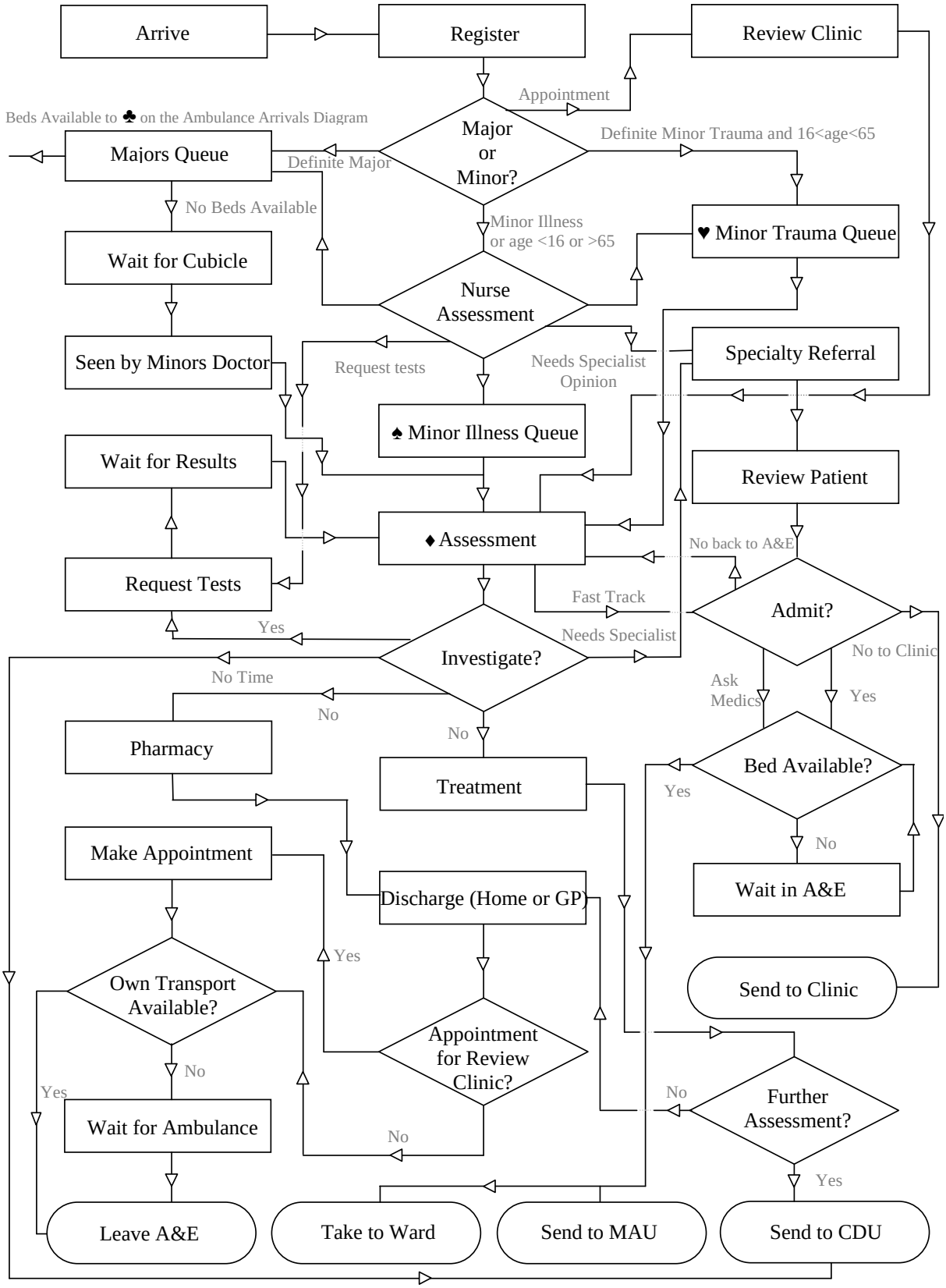




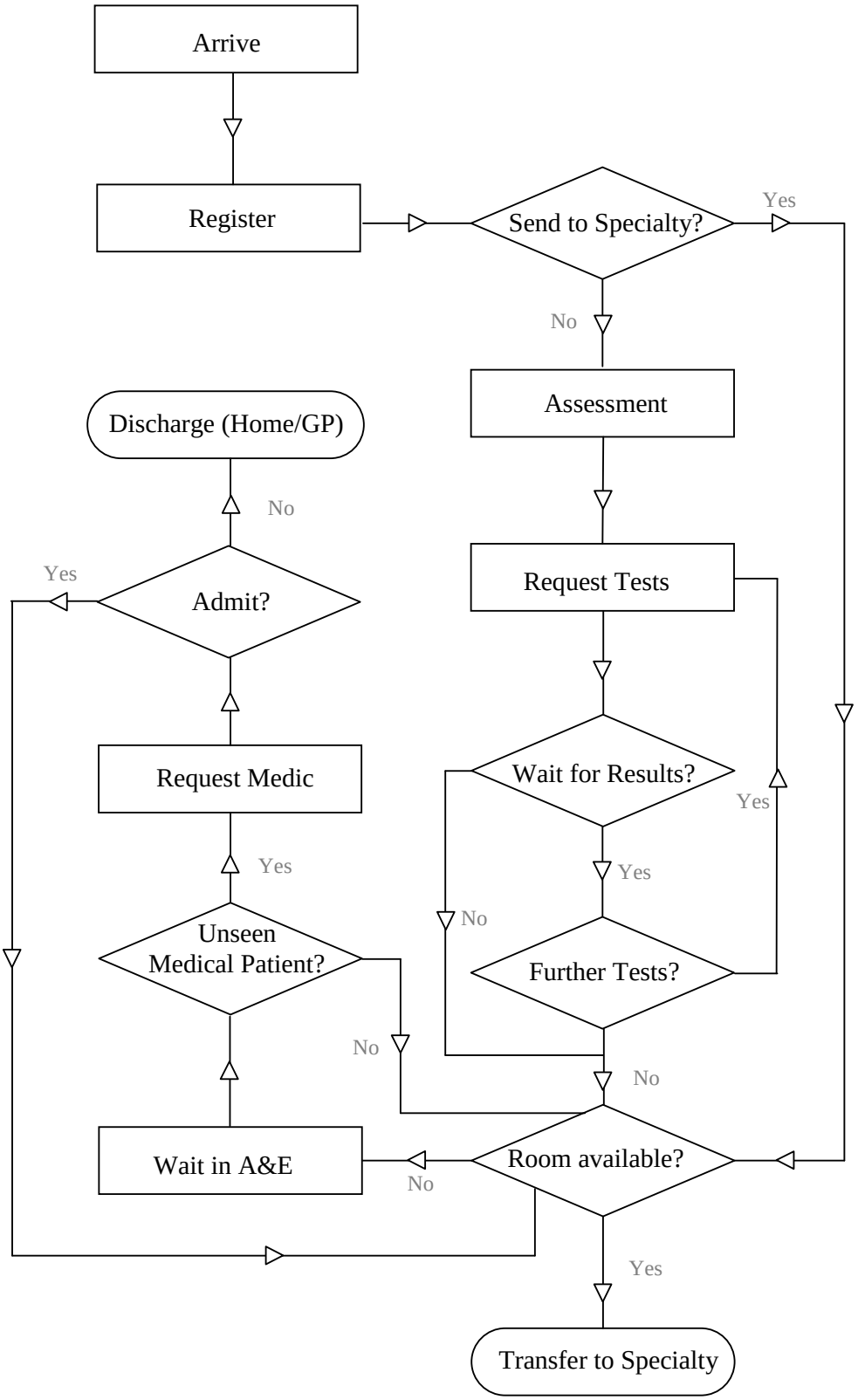
Appendix C

Patient Flow Diagrams

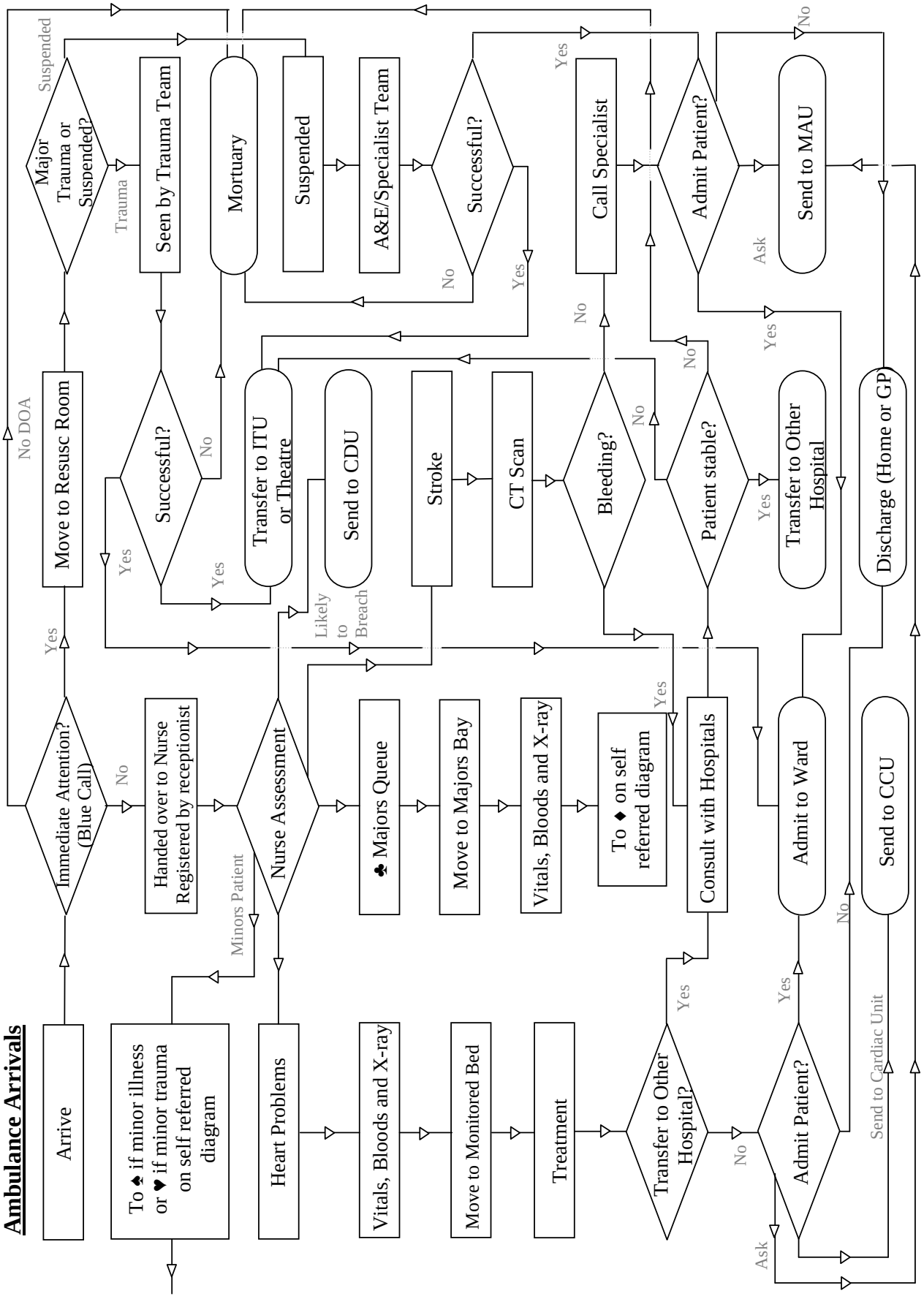
C.1 Self-Referred Arrival Flow Diagrams



C.2 GP-Referred Arrival Flow Diagrams



C.3 Ambulance Arrival Flow Diagrams



Appendix D

Mean and Standard Deviation of Patient Service Times

D.1 Simulation Results (by arrival mode)

	walk-in arrivals					
	majors priority		**minors priority**		**no priority**	
load	mean	std dev	mean	std dev	mean	std dev
0.25	1.5367	1.3876	1.5360	1.3866	1.5369	1.3862
0.3	1.5457	1.3935	1.5448	1.3922	1.5452	1.3924
0.4	1.5699	1.4086	1.5681	1.4076	1.5695	1.4082
0.5	1.6080	1.4355	1.6031	1.4322	1.6053	1.4312
0.6	1.6662	1.4771	1.6547	1.4689	1.6591	1.4671
0.7	1.7609	1.5489	1.7291	1.5220	1.7427	1.5226
0.75	1.8349	1.6082	1.7798	1.5590	1.8030	1.5632
0.8	1.9401	1.6981	1.8447	1.6073	1.8815	1.6165
0.85	2.1005	1.8453	1.9247	1.6667	1.9967	1.7024
0.9	2.3822	2.1323	2.0324	1.7474	2.1641	1.8345
0.95	2.9197	2.7274	2.1788	1.8628	2.4569	2.1020
1.0	4.4391	4.6034	2.3926	2.0271	3.0440	2.7149
1.05	–	–	2.7347	2.3059	–	–
1.1	–	–	3.4227	2.9136	–	–
1.15	–	–	5.6451	5.0911	–	–

Table D.1: Mean and standard deviation (std dev) of service times (in hours) for walk-in arrivals under varying workloads as calculated by simulation, including the results for workloads over 1.0 for minors priority only.

	ambulance arrivals					
	majors priority		**minors priority**		**no priority**	
load	mean	std dev	mean	std dev	mean	std dev
0.25	1.9712	1.4617	1.9731	1.4640	1.9721	1.4616
0.3	1.9789	1.4656	1.9801	1.4652	1.9797	1.4663
0.4	1.9986	1.4772	2.0042	1.4849	2.0007	1.4777
0.5	2.0278	1.4948	2.0382	1.5088	2.0331	1.5000
0.6	2.0688	1.5195	2.0894	1.5512	2.0779	1.5286
0.7	2.1274	1.5559	2.1600	1.6114	2.1422	1.5711
0.75	2.1704	1.5845	2.2074	1.6557	2.1848	1.6001
0.8	2.2235	1.6256	2.2658	1.7124	2.2387	1.6370
0.85	2.3003	1.6926	2.3370	1.7860	2.3093	1.6892
0.9	2.4206	1.8247	2.4271	1.8822	2.4013	1.7620
0.95	2.6338	2.1283	2.5466	2.0223	2.5433	1.8979
1.0	3.1934	3.2386	2.7096	2.2101	2.7952	2.2145
1.05	–	–	2.9528	2.5084	–	–
1.1	–	–	3.4064	3.080	–	–
1.15	–	–	4.5646	4.5912	–	–

Table D.2: Mean and standard deviation (std dev) of service times (in hours) for ambulance arrivals under varying workloads as calculated by simulation, including the results for workloads over 1.0 for minors priority only.

D.2 AGFA Technique Results (by priority scheme)

	no priority			
	walk-in arrivals		**ambulance arrivals**	
load	mean	std dev	mean	std dev
0.25	1.5148	1.5517	1.9238	1.7725
0.3	1.5238	1.5592	1.9321	1.7797
0.4	1.5494	1.5805	1.9550	1.7999
0.5	1.5878	1.6130	1.9881	1.8300
0.6	1.6446	1.6615	2.0350	1.8740
0.7	1.7305	1.7346	2.1016	1.9386
0.75	1.7910	1.7855	2.1456	1.9822
0.8	1.8705	1.8516	2.2002	2.0366
0.85	1.9818	1.9438	2.2720	2.1077
0.9	2.1615	2.0983	2.3867	2.2152
0.95	2.5461	2.4498	2.6717	2.4424
1.0	3.5283	3.2031	3.5589	2.9775

Table D.3: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under no priority for varying workloads as calculated using the AGFA technique.

	majors priority			
	walk-in arrivals		**ambulance arrivals**	
load	mean	std dev	mean	std dev
0.25	1.5151	1.5517	1.9231	1.7716
0.3	1.5243	1.5592	1.9309	1.7780
0.4	1.5506	1.5806	1.9519	1.7954
0.5	1.5905	1.6136	1.9815	1.8202
0.6	1.6504	1.6636	2.0223	1.8547
0.7	1.7434	1.7419	2.0796	1.9039
0.75	1.8116	1.7992	2.1178	1.9373
0.8	1.9060	1.8787	2.1670	1.9810
0.85	2.0518	2.0048	2.2405	2.0459
0.9	2.3428	2.2793	2.4125	2.1822
0.95	3.1581	3.0696	3.0824	2.5904
1.0	4.2218	4.2426	4.0598	3.4293

Table D.4: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under majors priority for varying workloads as calculated using the AGFA technique.

	minors priority			
	walk-in arrivals		**ambulance arrivals**	
load	mean	std dev	mean	std dev
0.25	1.5145	1.5517	1.9244	1.7734
0.3	1.5234	1.5591	1.9333	1.7814
0.4	1.5481	1.5802	1.9582	1.8046
0.5	1.5850	1.6125	1.9951	1.8412
0.6	1.6388	1.6612	2.0486	1.8988
0.7	1.7181	1.7366	2.1266	1.9932
0.75	1.7718	1.7910	2.1787	2.0648
0.8	1.8393	1.8643	2.2432	2.1654
0.85	1.9259	1.9701	2.3243	2.3185
0.9	2.0462	2.1426	2.4293	2.5850
0.95	2.2004	2.4935	2.5723	3.1652
1.0	2.4418	3.4517	2.7880	4.7862

Table D.5: Mean and standard deviation (std dev) of service times (in hours) for walk-in and ambulance arrivals under minors priority for varying workloads as calculated using the AGFA technique.

Appendix E

Mathematica Implementation of the AGFA Technique

```
(* comments are placed inside brackets *)

(*********************************************************)
(*                                                       *)
(*          Closed 2-class MVA Priority Model            *)
(*                                                       *)
(*********************************************************)

MVA2MMmP[S1_, S2_, v_, vf2_, m_, lam_] :=
 Block[{M, R, totmu, B, B2, H, Hf2, Hcheck, Q, Q2, W, W2, pi, q, X}, lam1 = lam;
  {K1, K2} = Dimensions[S1][[{2, 3}]] - 1;
  M = Length[S1];
  R = 2; IR = {{1, 0}, {0, 1}};
  totmu =
   Table[If[S1[[i, a, b, 1]] == 0, If[S1[[i, a, b, 2]] == 0, Infinity, 1 / S1[[i, a, b, 2]]],
     1 / S1[[i, a, b, 1]] + If[S1[[i, a, b, 2]] == 0, 0, 1 / S1[[i, a, b, 2]]]],
    {i, 1, M}, {a, 1, K1 + 1}, {b, 1, K2 + 1}];
  B = Table[{S1[[i, Min[2, K1 + 1], 1, 1]], S1[[i, 1, Min[2, K2 + 1], 2]]}, {i, 1, M}];
  B2 = Table[{S2[[i, Min[2, K1 + 1], 1, 1]], S2[[i, 1, Min[2, K2 + 1], 2]]}, {i, 1, M}];
  H = Hf2 = Hcheck = Table[0, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {i, 1, M}];
  W = W2 = Q =
     Q2 = L = Lf2 = LQ = LQf2 = Table[0, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {i, 1, M}, {r, 1, R}];
  R1 = R2 = Table[0, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {r, 1, R}];
  pi = Table[0, {i, 1, M}, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {j1, 1, K1 + 1}, {j2, 1, K2 + 1}];
  Do[pi[[i, 1, 1, 1, 1]] = 1, {i, 1, M}];
  T = Table[0, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {r, 1, R}];
  S0 = S02 = Table[0, {i, 1, M}, {r, 1, R}];
  q = Table[0, {i, 1, M}, {r, 1, R}];
  X = Table[0, {r, 1, R}, {j, 1, 3}];

  Do[Do[If[k1 + k2 > m[[i]] + 2,
     X = {If[k1 > 1, {0, 0, 0} + Sum[pi[[i, k1 - 1, k2, j1, j2]] / {totmu[[i, j1, j2]],
            0.5 totmu[[i, j1, j2]]^2, 1}, {j1, 1, k1 - 1}, {j2, Max[1, m[[i]] - j1 + 2], k2}],
         {0, 0, 0}], If[k2 > 1 , {0, 0, 0} + Sum[pi[[i, k1, k2 - 1, j1, j2]] /
            {totmu[[i, j1, j2]], 0.5 totmu[[i, j1, j2]]^2, 1}, {j1, 1, k1}, {j2,
            Max[1, m[[i]] - j1 + 2], k2 - 1}], {0, 0, 0}]}, X = Table[0, {r, 1, R}, {j, 1, 3}]];
    q[[i]] = Transpose[X][[3]];
    S0[[i]] = Transpose[X][[1]] /
      {If[q[[i, 1]] > 0, q[[i, 1]], 1], If[q[[i, 2]] > 0, q[[i, 2]], 1]}; S02[[i]] =
     Transpose[X][[2]] / {If[q[[i, 1]] > 0, q[[i, 1]], 1], If[q[[i, 2]] > 0, q[[i, 2]], 1]};

    If[k1 > 1, Q[[k1, k2, i, 1]] = (q[[i, 1]] + LQ[[k1 - 1, k2, i, 1]]) S0[[i, 1]]];
    If[k2 > 1, Q[[k1, k2, i, 2]] = (q[[i, 2]] + H[[k1, k2 - 1, i]])
       S0[[i, 2]] / (1 - lam1 v[[i, 1]] S0[[i, 2]])]; W[[k1, k2, i]] = Q[[k1, k2, i]] + B[[i]];
    If[k1 == 1, W[[k1, k2, i, 1]] = 0];
    If[k2 == 1, W[[k1, k2, i, 2]] = 0];

    If[k1 > 1, Q2[[k1, k2, i, 1]] = (LQf2[[k1 - 1, k2, i, 1]] + 2 LQ[[k1 - 1, k2, i, 1]])
        S0[[i, 1]]^2 + (q[[i, 1]] + LQ[[k1 - 1, k2, i, 1]]) S02[[i, 1]]];
```

```
    If[k2 > 1, kk1 = k1; kk2 = k2 - 1;  rho = lam1 v[[i, 1]] S0[[i, 2]]; Q2[[k1, k2, i, 2]] =
      (q[[i, 2]] S02[[i, 2]] + Hf2[[kk1, kk2, i]] S0[[i, 2]]^2 + (H[[kk1, kk2, i]] +
            lam1 v[[i, 1]] Q[[k1, k2, i, 2]]) (S02[[i, 2]] + 2 S0[[i, 2]]^2)) / (1 - rho^2) +
       2 rho S0[[i, 2]]^2 (Hf2[[kk1, kk2, i]] + H[[kk1, kk2, i]]) / ((1 - rho) (1 - rho^2))];

    W2[[k1, k2, i]] = Q2[[k1, k2, i]] + B2[[i]] + 2 Q[[k1, k2, i]] B[[i]];

    , {i, 1, M}];

   T[[k1, k2]] =
    {k1 - 1, k2 - 1} / (If[k1 == 1, If[k2 == 1, {1, 1}, {1, 0}], If[k2 == 1, {0, 1}, {0, 0}]]
       + Sum[v[[i]] W[[k1, k2, i]], {i, 1, M}]);

   Do[LQ[[k1, k2, i]] = T[[k1, k2]] v[[i]] Q[[k1, k2, i]]; L[[k1, k2, i]] =
     T[[k1, k2]] v[[i]] W[[k1, k2, i]]; H[[k1, k2, i]] = Apply[Plus, LQ[[k1, k2, i]]];
    Hcheck[[k1, k2, i]] = Dot[T[[k1, k2]] v[[i]], Q[[k1, k2, i]]];
    LQf2[[k1, k2, i]] = (T[[k1, k2]] v[[i]])^2 Q2[[k1, k2, i]];
    Lf2[[k1, k2, i]] = (T[[k1, k2]] v[[i]])^2 W2[[k1, k2, i]]; Hf2[[k1, k2, i]] =
     (Plus @@ (T[[k1, k2]] v[[i]]))^2 Dot[Q2[[k1, k2, i]], v[[i]]] / Plus @@ v[[i]];
    , {i, 1, M}];

   R1[[k1, k2]] = Sum[v[[i]] W[[k1, k2, i]], {i, 1, M}];
   R2[[k1, k2]] = R1[[k1, k2]]^2 +
     Sum[(vf2[[i]] - v[[i]]^2) W[[k1, k2, i]]^2 + v[[i]] W2[[k1, k2, i]], {i, 1, M}];

   Do[pi[[i, k1, k2, j1, j2]] = v[[i, 1]] T[[k1, k2, 1]] pi[[i, k1 - 1, k2, j1 - 1, j2]]
      S1[[i, j1, j2, 1]], {j1, 2, k1}, {j2, 1, k2}, {i, 1, M}];

   Do[pi[[i, k1, k2, 1, j2]] = v[[i, 2]] T[[k1, k2, 2]]
      pi[[i, k1, k2 - 1, 1, j2 - 1]] S1[[i, 1, j2, 2]], {j2, 2, k2}, {i, 1, M}];

   Do[pi[[i, k1, k2, 1, 1]] = 1 + pi[[i, k1, k2, 1, 1]] -
      Sum[pi[[i, k1, k2, j1, j2]], {j1, 1, k1}, {j2, 1, k2}], {i, 1, M}];
   , {k1, 1, K1 + 1}, {k2, 1, K2 + 1}];
 ]


(*********************************************************)
(*                                                       *)
(*          Closed 2-class MVA No-priority Model         *)
(*                                                       *)
(*********************************************************)

MVA2MMmNP[S1_, S2_, v_, vf2_, m_] :=
 Block[{totmu, B, B2, H, Hf2, Hcheck, Q, Q2, pi, q, X},
  {K1, K2} = Dimensions[S1][[{2, 3}]] - 1;
  M = Length[S1];
  R = 2; IR = {{1, 0}, {0, 1}};
  totmu =
   Table[If[S1[[i, a, b, 1]] = 0, If[S1[[i, a, b, 2]] = 0, Infinity, 1 / S1[[i, a, b, 2]]],
     1 / S1[[i, a, b, 1]] + If[S1[[i, a, b, 2]] = 0, 0, 1 / S1[[i, a, b, 2]]]],
```

```
  {i, 1, Length[mu]}, {a, 1, K1 + 1}, {b, 1, K2 + 1}];
B = Table[{S1[[i, Min[2, K1 + 1], 1, 1]], S1[[i, 1, Min[2, K2 + 1], 2]]}, {i, 1, M}];
B2 = Table[{S2[[i, Min[2, K1 + 1], 1, 1]], S2[[i, 1, Min[2, K2 + 1], 2]]}, {i, 1, M}];
H = Hf2 = Hcheck = Table[0, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {i, 1, M}];
W = W2 = Q =
   Q2 = L = Lf2 = LQ = LQf2 = Table[0, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {i, 1, M}, {r, 1, R}];
R1 = R2 = Table[0, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {r, 1, R}];
pi = Table[0, {i, 1, M}, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {j1, 1, K1 + 1}, {j2, 1, K2 + 1}];
Do[pi[[i, 1, 1, 1, 1]] = 1, {i, 1, M}];
T = Table[0, {k1, 1, K1 + 1}, {k2, 1, K2 + 1}, {r, 1, R}];
S0 = S02 = Table[0, {i, 1, M}, {r, 1, R}];
q = Table[0, {i, 1, M}, {r, 1, R}];
X = Table[0, {r, 1, R}, {j, 1, 3}];

Do[Do[If[k1 + k2 > m[[i]] + 2,
   X = {If[k1 > 1, {0, 0, 0} + Sum[pi[[i, k1 - 1, k2, j1, j2]] / {totmu[[i, j1, j2]],
         0.5 totmu[[i, j1, j2]]^2, 1}, {j1, 1, k1 - 1}, {j2, Max[1, m[[i]] - j1 + 2], k2}],
      {0, 0, 0}], If[k2 > 1 , {0, 0, 0} + Sum[pi[[i, k1, k2 - 1, j1, j2]] /
        {totmu[[i, j1, j2]], 0.5 totmu[[i, j1, j2]]^2, 1}, {j1, 1, k1}, {j2,
        Max[1, m[[i]] - j1 + 2], k2 - 1}], {0, 0, 0}]}, X = Table[0, {r, 1, R}, {j, 1, 3}]];
  q[[i]] = Transpose[X][[3]];
  S0[[i]] = Transpose[X][[1]] /
    {If[q[[i, 1]] > 0, q[[i, 1]], 1], If[q[[i, 2]] > 0, q[[i, 2]], 1]}; S02[[i]] =
   Transpose[X][[2]] / {If[q[[i, 1]] > 0, q[[i, 1]], 1], If[q[[i, 2]] > 0, q[[i, 2]], 1]};

  If[k1 > 1, Q[[k1, k2, i, 1]] = (q[[i, 1]] + H[[k1 - 1, k2, i]]) S0[[i, 1]]];
  If[k2 > 1, Q[[k1, k2, i, 2]] = (q[[i, 2]] + H[[k1, k2 - 1, i]]) S0[[i, 2]]];
  W[[k1, k2, i]] = Q[[k1, k2, i]] + B[[i]];
  If[k1 == 1, W[[k1, k2, i, 1]] = 0];
  If[k2 == 1, W[[k1, k2, i, 2]] = 0];

  If[k1 > 1, Q2[[k1, k2, i, 1]] = (Hf2[[k1 - 1, k2, i]] + 2 H[[k1 - 1, k2, i]]) S0[[i, 1]]^2 +
     (q[[i, 1]] + H[[k1 - 1, k2, i]]) S02[[i, 1]]];
  If[k2 > 1, Q2[[k1, k2, i, 2]] = (Hf2[[k1, k2 - 1, i]] + 2 H[[k1, k2 - 1, i]]) S0[[i, 2]]^2 +
     (q[[i, 2]] + H[[k1, k2 - 1, i]]) S02[[i, 2]]];

  W2[[k1, k2, i]] = Q2[[k1, k2, i]] + B2[[i]] + 2 Q[[k1, k2, i]] B[[i]];

  , {i, 1, M}];

 T[[k1, k2]] =
  {k1 - 1, k2 - 1} / (If[k1 == 1, If[k2 == 1, {1, 1}, {1, 0}], If[k2 == 1, {0, 1}, {0, 0}]]
     + Sum[v[[i]] W[[k1, k2, i]], {i, 1, M}]);

 Do[LQ[[k1, k2, i]] = T[[k1, k2]] v[[i]] Q[[k1, k2, i]]; L[[k1, k2, i]] =
   T[[k1, k2]] v[[i]] W[[k1, k2, i]]; H[[k1, k2, i]] = Apply[Plus, LQ[[k1, k2, i]]];
  Hcheck[[k1, k2, i]] = Dot[T[[k1, k2]] v[[i]], Q[[k1, k2, i]]];
  LQf2[[k1, k2, i]] = (T[[k1, k2]] v[[i]])^2 Q2[[k1, k2, i]];
  Lf2[[k1, k2, i]] = (T[[k1, k2]] v[[i]])^2 W2[[k1, k2, i]]; Hf2[[k1, k2, i]] =
   (Plus @@ (T[[k1, k2]] v[[i]]))^2 Dot[Q2[[k1, k2, i]], v[[i]]] / Plus @@ v[[i]];
  , {i, 1, M}];
```

```
  R1[[k1, k2]] = Sum[v[[i]] W[[k1, k2, i]], {i, 1, M}];
  R2[[k1, k2]] = R1[[k1, k2]]^2 +
    Sum[(vf2[[i]] - v[[i]]^2) W[[k1, k2, i]]^2 + v[[i]] W2[[k1, k2, i]], {i, 1, M}];

  Do[pi[[i, k1, k2, j1, j2]] = v[[i, 1]] T[[k1, k2, 1]] pi[[i, k1 - 1, k2, j1 - 1, j2]]
     S1[[i, j1, j2, 1]], {j1, 2, k1}, {j2, 1, k2}, {i, 1, M}];

  Do[pi[[i, k1, k2, 1, j2]] = v[[i, 2]] T[[k1, k2, 2]]
     pi[[i, k1, k2 - 1, 1, j2 - 1]] S1[[i, 1, j2, 2]], {j2, 2, k2}, {i, 1, M}];

  Do[pi[[i, k1, k2, 1, 1]] = 1 + pi[[i, k1, k2, 1, 1]] -
     Sum[pi[[i, k1, k2, j1, j2]], {j1, 1, k1}, {j2, 1, k2}], {i, 1, M}];
  , {k1, 1, K1 + 1}, {k2, 1, K2 + 1}];
]


(*********************************************************)
(*                                                       *)
(*          Variable rate set-up for Multi-server        *)
(*                                                       *)
(*********************************************************)

Mu2[mu_, i_, 0, 0, m_] := {Infinity, Infinity}; Mu2[mu_, i_, j1_, j2_, m_] :=
 If[j1 == 0, {Infinity, Min[m, j2] mu[[i, 2]]}, If[j2 == 0, {Min[m, j1] mu[[i, 1]], Infinity},
   If[j1 + j2 < m + 1, ({j1, j2} mu[[i]]), ({j1, j2} mu[[i]]) m / (j1 + j2)]]]



(*********************************************************)
(*                                                       *)
(*          Open 2-class MVA non-priority Model          *)
(*                                                       *)
(*********************************************************)

(* Parallel independent multi-servers for each of 2 classes,
 single node.  Max[kr,Kr] at class r population kr (1≤r≤R, R=2).  Single
  server first and second moments r1,r2, cf. s1, s2 in closed model *)

SingleNode2[r1_, r2_, lambda_] := Block[{},
  {K1, K2} = Dimensions[r1][[{1, 2}]] - 1;
  R = Last[Dimensions[r1]];
  B = {r1[[Min[2, K1 + 1], 1, 1]], r1[[1, Min[2, K2 + 1], 2]]};
  B2 = {r2[[Min[2, K1 + 1], 1, 1]], r2[[1, Min[2, K2 + 1], 2]]};
  W = W2 = Q = Q2 = L = LQ = LQf2 = Table[0, {r, 1, R}];

  Mu[j1_, j2_] :=
   {If[j1 < K1 + 2, Max[1, j1 - 1], Max[1, K1]], If[j2 < K2 + 2, Max[1, j2 - 1], Max[1, K2]]} /
    ({If[j1 == 1, Infinity, 0], If[j2 == 1, Infinity, 0]} +
      r1[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1]]]);
```

```
statemap[i_, j_, n_] :=
 If[(i == n) || (j == n), 1 + (n - 1)^2, (i - 1) (n - 1) + j]; invmap[k_, n_] :=
 If[k == 1 + (n - 1)^2, {n, 1}, {Quotient[k - 1, n - 1], Mod[k - 1, n - 1]} + {1, 1}];

eps = 1.0; n = Max[K1, K2] - 5; pi = Table[0, {i, 1, n - 1}, {j, 1, n - 1}];
While[eps > 10^-3, pi1 = pi; n += 10;
  pi = Table[0, {i, 1, n - 1}, {j, 1, n - 1}];

  m = Table[0, {i, 1, n}, {j, 1, n}, {k, 1, n}, {l, 1, n}];
  Do[If[i > 1, m[[i - 1, j, i, j]] = lambda[[1]]];
   If[j > 1, m[[i, j - 1, i, j]] = lambda[[2]]];
   m[[i + 1, If[i == n - 1, 1, j], i, j]] += Mu[i + 1, j][[1]];
   m[[If[j == n - 1, n, i], If[j == n - 1, 1, j + 1], i, j]] += Mu[i, j + 1][[2]];
   m[[i, j, i, j]] = -(Dot[{If[i > 1, 1, 0], If[j > 1, 1, 0]}, Mu[i, j]] + Plus @@ lambda),
   {i, 1, n - 1}, {j, 1, n - 1}];

  Do[m[[n - 1, j, n, 1]] += lambda[[1]];
   m[[j, n - 1, n, 1]] += lambda[[2]],
   {j, 1, n - 1}];
  m[[n, 1, n, 1]] = -Sum[Mu[i, n][[2]], {i, 1, n - 1}] - Sum[Mu[n, j][[1]], {j, 1, n - 1}];

  m2 = Table[m[[#1, #2, #3, #4]] & @@ Flatten[(invmap @@ #) & /@ {{k, n}, {l, n}}],
    {k, 1, 1 + (n - 1)^2}, {l, 1, 1 + (n - 1)^2}];

  m3 = Append[Drop[#, -2], 1] & /@ Drop[m2, -1];
  pilin = N[LinearSolve[Transpose[m3], Append[Table[0, {i, 1, Length[m3] - 1}], 1]]];
  Do[pi[[i, j]] = pilin[[statemap[i, j, 1 + Length[pi]]]],
   {i, 1, Length[pi]}, {j, 1, Length[pi]}];

  eps = Max @@ Abs[Flatten[(Take[#, Length[pi1] - 5] &) /@ Take[pi, Length[pi1] - 5] -
       (Take[#, Length[pi1] - 5] &) /@ Take[pi1, Length[pi1] - 5]]];
  Print["n=", n, ", eps=", eps];]

 q = 1 - {Sum[pi[[j1, j2]], {j1, 1, Min[Length[pi], K1]}, {j2, 1, Length[pi]}],
    Sum[pi[[j1, j2]], {j1, 1, Length[pi]}, {j2, 1, Min[Length[pi], K2]}]};

S0 =
 q / {Sum[pi[[j1, j2]] Mu[j1, j2][[1]], {j1, K1 + 1, Length[pi]}, {j2, 1, Length[pi]}],
   Sum[pi[[j1, j2]] Mu[j1, j2][[2]], {j1, 1, Length[pi]}, {j2, K2 + 1, Length[pi]}]};

S01 = {Sum[pi[[j1, j2]] r1[[K1 + 1, If[j2 < K2 + 2, j2, K2 + 1], 1]] / Max[1, K1],
    {j1, K1 + 1, Length[pi]}, {j2, 1, Length[pi]}],
   Sum[pi[[j1, j2]] r1[[If[j1 < K1 + 2, j1, K1 + 1], K2 + 1, 2]] / Max[1, K2],
    {j1, 1, Length[pi]}, {j2, K2 + 1, Length[pi]}]} /
  {If[q[[1]] > 0, q[[1]], 1], If[q[[2]] > 0, q[[2]], 1]};

S02 = {Sum[pi[[j1, j2]] r2[[K1 + 1, If[j2 < K2 + 2, j2, K2 + 1], 1]] / Max[1, K1]^2,
    {j1, K1 + 1, Length[pi]}, {j2, 1, Length[pi]}],
   Sum[pi[[j1, j2]] r2[[If[j1 < K1 + 2, j1, K1 + 1], K2 + 1, 2]] / Max[1, K2]^2,
    {j1, 1, Length[pi]}, {j2, K2 + 1, Length[pi]}]} /
  {If[q[[1]] > 0, q[[1]], 1], If[q[[2]] > 0, q[[2]], 1]};
```

```
  B = Sum[pi[[j1, j2]] {r1[[If[j1 < K1 + 1, j1 + 1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1], 1]],
      r1[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 1, j2 + 1, K2 + 1], 2]]},
    {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
  B2 = Sum[pi[[j1, j2]] {r2[[If[j1 < K1 + 1, j1 + 1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1], 1]],
      r2[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 1, j2 + 1, K2 + 1], 2]]},
    {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];

  pimarg = Transpose[{Plus @@ Transpose[pi], Plus @@ pi}];
  L = Sum[pi[[j1, j2]] {j1 - 1, j2 - 1}, {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
  Lf2 = Sum[pi[[j1, j2]] {(j1 - 2) (j1 - 1), (j2 - 2) (j2 - 1)},
    {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
  LQ = {Sum[pi[[j1, j2]] (j1 - K1 - 1), {j1, K1 + 2, Length[pi]}, {j2, 1, Length[pi]}],
    Sum[pi[[j1, j2]] (j2 - K2 - 1), {j1, 1, Length[pi]}, {j2, K2 + 2, Length[pi]}]};
  LQf2 = {Sum[pi[[j1, j2]] (j1 - K1 - 1) (j1 - K1 - 2), {j1, K1 + 2, Length[pi]},
     {j2, 1, Length[pi]}], Sum[pi[[j1, j2]] (j2 - K2 - 1) (j2 - K2 - 1),
     {j1, 1, Length[pi]}, {j2, K2 + 2, Length[pi]}]};

  nzlambda = Table[If[lambda[[i]] == 0, -1, lambda[[i]]], {i, 1, Length[lambda]}];
  W = L / nzlambda;
  Q = LQ / nzlambda;
  W20 = Lf2 / (nzlambda^2);
  Q2 = LQf2 / (nzlambda^2);
  W2 = Q2 + B2 + 2 Q B;
  (*
   Q=q S0/(1-lambda S0);
   W=Q+B;
   LQ=lambda Q;
   L=lambda W;

   Q2=(2 LQ S0^2+(q+LQ) S02)/(1-(lambda S0)^2);
   W2=Q2+B2+2 Q B;

   LQf2=(lambda)^2 Q2;
   Lf2=(lambda)^2 W2;
   *)
 ]

(*********************************************************)
(*                                                       *)
(*          Open 1-class MVA non-priority Model          *)
(*                                                       *)
(*********************************************************)

(* Parallel independent multi-servers for each of 2 classes,
 using product-form approximation.  WORKS FOR SINGLE CLASS CASE IF YOU SET K2 TO 0 *)

SingleNode[r1_, r2_, lambda_] := Block[{},
  {K1, K2} = Dimensions[r1][[{1, 2}]] - 1;
  R = Last[Dimensions[r1]];
  B = {r1[[Min[2, K1 + 1], 1, 1]], r1[[1, Min[2, K2 + 1], 2]]};
```

```
B2 = {r2[[Min[2, K1 + 1], 1, 1]], r2[[1, Min[2, K2 + 1], 2]]};
W = W2 = Q = Q2 = L = LQ = LQf2 = Table[0, {r, 1, R}];

mst = Table[r1[[ix1 = If[j1 < K1 + 2, j1, K1 + 1], ix2 = If[j2 < K2 + 2, j2, K2 + 1]]] /
   {If[j1 < K1 + 2, Max[1, j1 - 1], Max[1, K1]], If[j2 < K2 + 2, Max[1, j2 - 1], Max[1, K2]]},
  {j1, 1, 100}, {j2, 1, 100}];

eps = 10.0; n = 0; pi2 = {{1}};
While[eps > 10^-8, n++; pi = pi2;
 pi1 = Append[pi, Last[pi] lambda[[1]] (First /@ Take[mst[[n + 1]], n])];
 piT = Transpose[pi1]; pi2 = Transpose[
   Append[piT, Last[piT] lambda[[2]] (Last /@ Take[Transpose[mst][[n + 1]], n + 1])]];
 eps = Plus @@ Last[pi1] + Plus @@ Last[Transpose[pi2]]];

pi = pi2 / (Plus @@ Plus @@ pi);

q = 1 - {Sum[pi[[j1, j2]], {j1, 1, Min[Length[pi], K1]}, {j2, 1, Length[pi]}],
   Sum[pi[[j1, j2]], {j1, 1, Length[pi]}, {j2, 1, Min[Length[pi], K2]}]};

S0 = {Sum[pi[[j1, j2]] r1[[K1 + 1, If[j2 < K2 + 2, j2, K2 + 1], 1]] / Max[1, K1],
    {j1, K1 + 1, Length[pi]}, {j2, 1, Length[pi]}],
   Sum[pi[[j1, j2]] r1[[If[j1 < K1 + 2, j1, K1 + 1], K2 + 1, 2]] / Max[1, K2],
    {j1, 1, Length[pi]}, {j2, K2 + 1, Length[pi]}]} / q;
S02 = {Sum[pi[[j1, j2]] r2[[K1 + 1, If[j2 < K2 + 2, j2, K2 + 1], 1]] / Max[1, K1]^2,
    {j1, K1 + 1, Length[pi]}, {j2, 1, Length[pi]}],
   Sum[pi[[j1, j2]] r2[[If[j1 < K1 + 2, j1, K1 + 1], K2 + 1, 2]] / Max[1, K2]^2,
    {j1, 1, Length[pi]}, {j2, K2 + 1, Length[pi]}]} / q;

B = Sum[pi[[j1, j2]] {r1[[If[j1 < K1 + 1, j1 + 1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1], 1]],
    r1[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 1, j2 + 1, K2 + 1], 2]]},
  {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
B2 = Sum[pi[[j1, j2]] {r2[[If[j1 < K1 + 1, j1 + 1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1], 1]],
    r2[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 1, j2 + 1, K2 + 1], 2]]},
  {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];

pimarg = Transpose[{Plus @@ Transpose[pi], Plus @@ pi}];
L = Sum[pi[[j1, j2]] {j1 - 1, j2 - 1}, {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
Lf2 = Sum[pi[[j1, j2]] {(j1 - 2) (j1 - 1), (j2 - 2) (j2 - 1)},
  {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
LQ = {Sum[pi[[j1, j2]] (j1 - K1 - 1), {j1, K1 + 2, Length[pi]}, {j2, 1, Length[pi]}],
  Sum[pi[[j1, j2]] (j2 - K2 - 1), {j1, 1, Length[pi]}, {j2, K2 + 2, Length[pi]}]};
LQf2 = {Sum[pi[[j1, j2]] (j1 - K1 - 1) (j1 - K1 - 2), {j1, K1 + 2, Length[pi]},
   {j2, 1, Length[pi]}], Sum[pi[[j1, j2]] (j2 - K2 - 1) (j2 - K2 - 1),
   {j1, 1, Length[pi]}, {j2, K2 + 2, Length[pi]}]};

nzlambda = Table[If[lambda[[i]] == 0, -1, lambda[[i]]], {i, 1, Length[lambda]}];
W = L / nzlambda;
Q = LQ / nzlambda;
W20 = Lf2 / (nzlambda^2);
Q2 = LQf2 / (nzlambda^2);
W2 = Q2 + B2 + 2 Q B;
```

```
 (*Priority = 0 is no priority,
Priority = 1 is majors priority and Priority = 2 is minors priority*)

 Main[PRIORITY_, WalkInRate_, AmbRate_, BluRate_] := Block[{},

 (*********************************************************)
 (*                                                       *)
 (*         Traffic equations for open Whole Model        *)
 (*                                                       *)
 (*********************************************************)

 NCC = False;
 MAJ = Max[1, PRIORITY];
 MIN = 3 - MAJ; BLU = 3; SR = 4;
 {K1, K2} = KAEU = {25, 9}[[{MAJ, MIN}]];
 M = 5; R = 4;
 TR = Table[0, {a1, 1, M}, {c1, 1, R}, {a2, 1, M}, {c2, 1, R}];
 rowsums = v1 = Lambda = Table[0, {a1, 1, M}, {c1, 1, R}];
 Lambda[[1, SR]] = WalkInRate;
 Lambda[[2, MAJ]] = AmbRate;
 Lambda[[3, BLU]] = BluRate;
 e1 = Lambda;

 TR[[1, SR, 4, SR]] = 0.7;
TR[[1, SR, 5, MIN]] = 0.206;
 TR[[1, SR, 5, MAJ]] = 0.094;
 TR[[2, MAJ, 5, MAJ]] = 0.65;
TR[[2, MAJ, 1, SR]] = 0.35;
 TR[[4, SR, 5, MIN]] = 0.7682;
TR[[4, SR, 5, MAJ]] = 0.0854;
TR[[5, MAJ, 3, BLU]] = 0.0041 / 0.8335;
 TR[[5, MIN, 5, MAJ]] = p12 = 0.1336 / (0.1336 + 0.4236 + 0.09);

 If[NCC,
  TR[[1, SR, 4, SR]] = 0.7;
TR[[1, SR, 5, MIN]] = 0.3;
  TR[[1, SR, 5, MAJ]] = 0.0;
  TR[[2, MAJ, 5, MAJ]] = 1;
TR[[2, MAJ, 1, SR]] = 0.0;
  TR[[4, SR, 5, MIN]] = 0.8536;
TR[[4, SR, 5, MAJ]] = 0.0;
TR[[5, MAJ, 3, BLU]] = 0;
  TR[[5, MIN, 5, MAJ]] = 0];

 Do[rowsums[[j, r]] = Sum[TR[[j, r, i, s]], {i, 1, M}, {s, 1, R}], {j, 1, M}, {r, 1, R}];
eps = 1; While[eps > 10^-8, e = e1;
    Do[e1[[i, s]] = Lambda[[i, s]] + Sum[e[[j, r]] TR[[j, r, i, s]], {j, 1, M}, {r, 1, R}],
     {i, 1, M}, {s, 1, R}]; eps = Plus @@ Flatten[Abs[e - e1]]]
```

```
    (*********************************************************)
    (*                                                       *)
    (*            Patient Paths in open Whole Model          *)
    (*                                                       *)
    (*********************************************************)
    M = Length[Lambda];
  rowsums = v1 = Table[0, {a1, 1, M + 1}, {c1, 1, R}];
  PP = Table[0, {a1, 1, Length[v1]}, {c1, 1, R}, {a2, 1, Length[v1]}, {c2, 1, R}];
  v1[[1, SR]] = v1[[3, BLU]] = v1[[2, MAJ]] = 1;
  extv = v1;

  PP[[1, SR, 4, SR]] = 0.7;
PP[[1, SR, 5, SR]] = 0.094;
  PP[[1, SR, 6, SR]] = 0.206;
  PP[[2, MAJ, 5, MAJ]] = 0.65;
PP[[2, MAJ, 1, MAJ]] = 0.35;
  PP[[1, MAJ, 4, MAJ]] = 0.7;
PP[[1, MAJ, 5, MAJ]] = 0.094;
  PP[[1, MAJ, 6, MAJ]] = 0.206;
  PP[[4, SR, 5, SR]] = 0.0854;
PP[[4, MAJ, 5, MAJ]] = 0.0854;
PP[[4, SR, 6, SR]] = 0.7682;
PP[[4, MAJ, 6, MAJ]] = 0.7682;
PP[[5, MAJ, 3, MAJ]] = 0.0041 / 0.8335;
  PP[[6, SR, 5, SR]] = p12;
  PP[[6, MAJ, 5, MAJ]] = p12;

  If[NCC,
   PP[[1, SR, 4, SR]] = 0.7;
PP[[1, SR, 5, SR]] = 0;
   PP[[1, SR, 6, SR]] = 0.3;
   PP[[2, MAJ, 5, MAJ]] = 1.0;
PP[[2, MAJ, 1, MAJ]] = 0;
   PP[[1, MAJ, 4, MAJ]] = 0;
PP[[1, MAJ, 5, MAJ]] = 0;
   PP[[1, MAJ, 6, MAJ]] = 0;
   PP[[4, SR, 5, SR]] = 0;
PP[[4, MAJ, 5, MAJ]] = 0;
PP[[4, SR, 6, SR]] = 0.8536;
PP[[4, MAJ, 6, MAJ]] = 0;
PP[[5, MAJ, 3, MAJ]] = 0;
   PP[[6, MAJ, 5, MAJ]] = 0;
   PP[[6, SR, 5, SR]] = 0];

  Do[rowsums[[j, r]] = Sum[PP[[j, r, i, s]], {i, 1, Length[v1]}, {s, 1, R}],
   {j, 1, Length[v1]}, {r, 1, R}];

eps = 1; While[eps > 10^-8, v = v1; Do[v1[[i, s]] =
      extv[[i, s]] + Sum[v[[j, r]] PP[[j, r, i, s]], {j, 1, Length[v1]}, {r, 1, R}],
     {i, 1, Length[v1]}, {s, 1, R}]; eps = Plus @@ Flatten[Abs[v - v1]]]
```

```
 eps = 1; vf21 = v1; n = 0;
While[eps > 10^-8, vf2 = vf21; n++; Do[vf21[[i, s]] =
     v[[i, s]]^2 - extv[[i, s]]^2 + Sum[(vf2[[j, r]] - v[[j, r]]^2) (PP[[j, r, i, s]])^2,
       {j, 1, Length[v1]}, {r, 1, R}], {i, 1, Length[v1]}, {s, 1, R}];
   eps = Plus @@ Flatten[Abs[vf2 - vf21]]]

  {lREC, lRN, lRES, lAR, lAEU} = e;
lREC = Append[Select[e[[1]], # > 0 &], 0];
lRN = Append[Select[e[[2]], # > 0 &], 0];
lRES = Append[Select[e[[3]], # > 0 &], 0];
lAR = Append[Select[e[[4]], # > 0 &], 0];
lAEU = Select[e[[5]], # > 0 &];



(*********************************************************)
(*                                                       *)
(*   Closed Accident and Emergency Unit sub-network (AEU)  *)
(*                                                       *)
(*********************************************************)

MAEU = {KAEU[[MIN]], 4, 2, 2, 2, 2, KAEU[[MAJ]], 3, 4, 8};
muMIN = {60, 3.5, 3, 1.2, ttt, ttt, ttt, ttt, ttt, 2};
muMAJ = {ttt, ttt, 2.4, 1, 0.5, 1, 60, 4, 3, 2};
muAEU = Transpose[{muMAJ, muMIN}[[{MAJ, MIN}]]];
M = Length[MAEU]; R = 2;
RP = Table[0, {a1, 1, M + 1}, {c1, 1, R}, {a2, 1, M + 1}, {c2, 1, R}];
rowsums = vis = Table[0, {a1, 1, M + 1}, {c1, 1, R}];
vis[[M + 1, MIN]] = vis[[M + 1, MAJ]] = 1;
vis1 = visf2 = vis;

RP[[1, MIN, 2, MIN]] = 1;
RP[[2, MIN, M + 1, MIN]] = 0.5572;
RP[[2, MIN, 3, MIN]] = 0.3528 0.45;
RP[[2, MIN, 10, MIN]] = 0.3528 0.55;
RP[[2, MIN, 4, MIN]] = 0.09;
RP[[3, MIN, 2, MIN]] = 1;
RP[[3, MAJ, 9, MAJ]] = 1;
RP[[4, MIN, M + 1, MIN]] = 1;
RP[[4, MAJ, M + 1, MAJ]] = 1;
RP[[5, MAJ, M + 1, MAJ]] = 1;
RP[[6, MAJ, M + 1, MAJ]] = 1;
RP[[7, MAJ, 8, MAJ]] = 0.7745;
RP[[7, MAJ, 9, MAJ]] = 0.2255;
RP[[8, MAJ, 3, MAJ]] = 0.07;
RP[[8, MAJ, 10, MAJ]] = 0.93;
RP[[9, MAJ, 8, MAJ]] = 0.1665;
RP[[9, MAJ, 4, MAJ]] = 0.127;
RP[[9, MAJ, 5, MAJ]] = 0.06;
RP[[9, MAJ, 6, MAJ]] = 0.122;
RP[[9, MAJ, M + 1, MAJ]] = 0.5245;
```

```
  RP[[10, MIN, 2, MIN]] = 0.8364;
  RP[[10, MIN, 3, MIN]] = 0.1636;
  RP[[10, MAJ, 3, MAJ]] = 0.3871;
  RP[[10, MAJ, 9, MAJ]] = 0.6129;
  RP[[M + 1, MAJ, 7, MAJ]] = 1;
  RP[[M + 1, MIN, 1, MIN]] = 1;

  Do[rowsums[[j, r]] = Sum[RP[[j, r, i, s]], {i, 1, M}, {s, 1, R}], {j, 1, M}, {r, 1, R}];

eps = 1; While[eps > 10^-8, vis = vis1;
   Do[vis1[[i, s]] = Sum[vis[[j, r]] RP[[j, r, i, s]], {j, 1, M + 1}, {r, 1, R}],
    {i, 1, M}, {s, 1, R}]; eps = Plus @@ Flatten[Abs[vis - vis1]]];

  eps = 1; visf2 = visf21 = vis1; n = 0;
  While[eps > 10^-8, visf2 = visf21; n++;
   Do[visf21[[i, s]] = vis1[[i, s]]^2 + Sum[(visf2[[j, r]] - vis1[[j, r]]^2)
        (RP[[j, r, i, s]])^2, {j, 1, M}, {r, 1, R}], {i, 1, M}, {s, 1, R}];
   eps = Plus @@ Flatten[Abs[visf2 - visf21]]];

  VAEU = vis1;
  VAEUf2 = visf2; ttt = Infinity ;
  mu = muAEU;
  s1AEU =
   Table[1 / Mu2[mu, i, a, b, MAEU[[i]]], {i, 1, Length[mu]}, {a, 0, K1}, {b, 0, K2}];
  s2AEU = 2 s1AEU^2;
  If[PRIORITY > 0, MVA2MMmP[s1AEU, s2AEU, VAEU, VAEUf2, MAEU, lAEU[[1]]],
   MVA2MMmNP[s1AEU, s2AEU, VAEU, VAEUf2, MAEU]];
  r1AEU = R1; r2AEU = R2; LAEU = L; Lf2AEU = LQf2;
  TAEU = T;


  (*********************************************************)
  (*                                                       *)
  (*        Closed Assessment Room sub-network (AR)        *)
  (*                                                       *)
  (*********************************************************)

  VAR = {{1, 0}, {1, 0}};
  VARf2 = {{0, 0}, {0, 0}};
  KAR = {K1, K2} = {5, 0};
  MAR = {5, 3};
  muAR = {60.0, 4.0};
  mu = Transpose[{muAR, {ttt, ttt}}];
  s1AR = Table[1 / Mu2[mu, i, a, b, MAR[[i]]], {i, 1, Length[mu]}, {a, 0, K1}, {b, 0, K2}];
  s2AR = 2 s1AR^2;
  If[PRIORITY > 0, MVA2MMmP[s1AR, s2AR, VAR, VARf2, MAR, lAR[[1]]],
   MVA2MMmNP[s1AR, s2AR, VAR, VARf2, MAR]];
r1AR = R1; r2AR = R2; LAR = LQ; Lf2AR = LQf2;
```

```
  (*********************************************************)
  (*                                                       *)
  (*          Closed Resusc sub-network (RES)              *)
  (*                                                       *)
  (*********************************************************)

  VRES = {{1, 0}, {1, 0}};
  VRESf2 = {{0, 0}, {0, 0}};
  KRES = {K1, K2} = {4, 0};
  MRES = {4, 2};
  muRES = {60.0, 0.5};
  mu = Transpose[{muRES, {ttt, ttt}}];
  s1RES =
   Table[1 / Mu2[mu, i, a, b, MRES[[i]]], {i, 1, Length[mu]}, {a, 0, K1}, {b, 0, K2}];
  s2RES = 2 s1RES^2;
  If[PRIORITY > 0, MVA2MMmP[s1RES, s2RES, VRES, VRESf2, MRES, lRES[[1]]],
   MVA2MMmNP[s1RES, s2RES, VRES, VRESf2, MRES]];
r1RES = R1; r2RES = R2; LRES = LQ; Lf2RES = LQf2;


  (*********************************************************)
  (*                                                       *)
  (*              Simple nodes -- Reception (REC)          *)
  (*                                                       *)
  (*********************************************************)

  MREC = KREC = {K1, K2} = {2, 0};
  muREC = {{6.0, ttt}};
  r1REC = Table[{a, b} / Mu2[muREC, 1, a, b, KREC[[1]]], {a, 0, K1}, {b, 0, K2}];
  r2REC = 2 r1REC^2;


  (*********************************************************)
  (*                                                       *)
  (*              Receiving Nurse (RN)                     *)
  (*                                                       *)
  (*********************************************************)

  MRN = KRN = {K1, K2} = {2, 0};
  muRN = {{4.0, ttt}};
  r1RN = Table[{a, b} / Mu2[muRN, 1, a, b, KRN[[1]]], {a, 0, K1}, {b, 0, K2}];
  r2RN = 2 r1RN^2;

  OW = OW2 = {}; VREC = VRN = VAR; wght = 1;

  SingleNode[r1REC, r2REC, lREC];
LsREC = {LAGGREC = L}; Lf2sREC = {Lf2REC = Lf2};
OW = Append[OW, W]; OW2 = Append[OW2, W2];

SingleNode[r1RN, r2RN, lRN];
LsRN = {LAGGRN = L}; Lf2sRN = {Lf2RN = Lf2};
```

```
OW = Append[OW, W]; OW2 = Append[OW2, W2];

  SingleNode[r1RES, r2RES, lRES];
LAGGRES = L;
  LsRES = Sum[pi[[j1, j2]] LRES[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1]]],
    {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
Lf2sRES = Sum[pi[[j1, j2]] Lf2RES[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1]]],
    {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
OW = Append[OW, W]; OW2 = Append[OW2, W2];

SingleNode[r1AR, r2AR, lAR];
LAGGAR = L;
  LsAR = Sum[pi[[j1, j2]] LAR[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1]]],
    {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
Lf2sAR = Sum[pi[[j1, j2]] Lf2AR[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1]]],
    {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
OW = Append[OW, W]; OW2 = Append[OW2, W2];

If[PRIORITY >= 0, SingleNode2[r1AEU, r2AEU, lAEU], SingleNode[r1AEU, r2AEU, lAEU]];
LAGGAEU = L;
  LsAEU = Sum[pi[[j1, j2]] LAEU[[If[j1 < K1 + 2, j1, K1 + 1], If[j2 < K2 + 2, j2, K2 + 1]]],
    {j1, 1, Length[pi]}, {j2, 1, Length[pi]}];
OW = Append[OW, W]; OW2 = Append[OW2, W2];

  POW = Join[First /@ Drop[OW, -1], Last[OW][[{MAJ, MIN}]]];
POW2 = Join[First /@ Drop[OW2, -1], Last[OW2][[{MAJ, MIN}]]];
OR1 = Table[Sum[v[[i, c]] POW[[i]], {i, 1, Length[v]}], {c, 1, 4}];
OR2 = OR1^2 + Table[Sum[(vf2[[i, c]] - v[[i, c]]^2) POW[[i]]^2 + v[[i, c]] POW2[[i]],
      {i, 1, Length[v1]}], {c, 1, 4}];
  ORSD = Sqrt[OR2 - OR1^2];
  Print[OR1[[{SR, BLU, MAJ}]], "  ",
   Sqrt[OR2 - OR1^2][[{SR, BLU, MAJ}]]];]



(*run for workloads between 0.25 to 1.0 under no priority*)
Table[Main[0, t 8.1, t 3, 0.1667], {t, 0.25, 1, 0.05}]

(*run for workloads between 0.25 to 1.0 under majors priority*)
Table[Main[1, t 8.1, t 3, 0.1667], {t, 0.25, 1.0, 0.05}];

(*run for workloads between 0.25 to 1.0 under minors priority*)

Table[Main[2, t 8.1, t 3, 0.1667], {t, 0.25, 1.0, 0.05}];
```

Bibliography

[1] J. Abate, G.L. Choudhury, and W. Whitt. On the Laguerre method for numerically inverting Laplace transforms. *INFORMS Journal on Computing*, 8(4):413–427, 1996.

[2] J. Abate, G.L. Choudhury, and W. Whitt. An introduction to numerical transform inversion and its application to probability models. In W. Grassman, editor, *Computational Probability*, pages 257–323, Kluwer, Boston, 2000.

[3] J. Abate and W. Whitt. The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10(1):5–88, 1992.

[4] J. Abate and W. Whitt. Numerical inversion of Laplace transforms of probability distributions. *ORSA Journal on Computing*, 7(1):36–43, 1995.

[5] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.

[6] I.F. Akyildiz. Mean Value Analysis for blocking queueing networks. *IEEE Transactions on Software Engineering*, 14(4):418–428, 1988. April.

[7] S.W.M. Au-Yeung. Finding probability distributions from moments. Master's thesis, Imperial College London, September 2003.

[8] S.W.M. Au-Yeung, N.J. Dingle, and W.J. Knottenbelt. Efficient Approximation of Response Time Densities and Quantiles in Stochastic Models. In *Proceedings of the 4th International Workshop on Software and Performance (WOSP 2004)*, pages 151–155, Redwood Shores, California, USA, January 2004.

[9] S.W.M. Au-Yeung, U. Harder, E.J. McCoy, and W.J. Knottenbelt. Predicting patient arrivals to an Accident and Emergency department. Submitted for review to the Emergency Medicine Journal, 2007.

[10] S.W.M. Au-Yeung, P.G. Harrison, and W.J. Knottenbelt. A Queueing Network Model of Patient Flow in an Accident and Emergency Department. In *Proceedings of the 20th Annual European and Simulation Modelling Conference*, pages 60–67, Toulouse, France, October 2006.

[11] S.W.M. Au-Yeung, P.G. Harrison, and W.J. Knottenbelt. Approximate queueing network analysis of patient treatment times. In *Proceedings of the Second International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2007)*, Nantes, France, October 2007.

[12] J. Banks, J.S. Carson, and B.L. Nelson. *Discrete-event System Simulation*. Prentice Hall International, 1994.

[13] M.S. Bartlett. Periodogram analysis and continuous spectra. *Biometrika*, 37(1/2):1–16, 1950.

[14] F. Bause and P.S. Kritzinger. *Stochastic Petri net theory*. Verlag Vieweg, Wiesbaden, Germany, 1995.

[15] BBC News Website. A&E closures "put lives at risk", August 2007. `http://news.bbc.co.uk/1/hi/health/6955438.stm`.

[16] J.T. Blake and M.W. Carter. An analysis of Emergency Room wait time issues via computer simulation. *Information Systems and Operational Research (INFOR)*, 34(4):263–273, November 1996.

[17] G. Bolch, S. Greiner, H. de Meer, and K.S. Trivedi. *Queueing Networks and Markov Chains*. Wiley, August 1998.

[18] L. von Bortkewitsch. *Das Gesetz der kleinen Zahlen (The Law of Small Numbers)*. B. G. Teubner, Leipzig, 1898.

[19] G.E.P. Box, G.M. Jenkins, and G.C. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice-Hall, 3rd edition, 1994.

[20] J.T. Bradley, N.J. Dingle, W.J. Knottenbelt, and P.G. Harrison. Performance queries on semi-Markov stochastic Petri nets with an extended Continuous Stochastic Logic. In *Proc. Petri Nets and Performance Models (PNPM'03)*, pages 62–71, 2003.

[21] P.J. Brockwell and R.A. Davis. *Introduction to Time Series and Forecasting*. Springer Texts In Statistics. Springer, 2nd edition, 2002.

[22] M.A. Centeno, R. Giachetti, R. Linn, and A.M. Ismail. A simulation-ILP based tool for scheduling ER staff. In *Proceedings of the 2003 Winter Simulation Conference*, pages 1930–1938, 2003.

[23] N. Channouf, P. L'Ecuyer, A. Ingolfsson, and A.N. Avramidis. The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. *Health Care Management Science*, 10(1):25–45(21), Feb 2007.

[24] C. Chatfield. Calculating interval forecasts (with discussion). *Journal of Business and Economic Statistics*, 11:121–144, 1993.

[25] C. Chatfield. *The Analysis of Time Series: Theory and Practice*. Chapman and Hall, 5th edition, 1996.

[26] K.K. Christoffel. Effect of Season and Weather on Paediatric Emergency Department Use. *American Journal of Emergency Medicine*, 3(4):327–330, 1985.

[27] T.J. Coats and S. Michalis. Mathematical modelling of patient flow through an Accident and Emergency department. *Emergency Medicine Journal*, 18:190–192, 2001.

[28] A. Cobham. Priority assignment in waiting line problems. *Operations Research*, 2(1):70–76, 1954.

[29] A. Codrington-Virtue. Simulating Accident and Emergency Services with a generic process. Nosokinetics newsletter issue 26, December 2005.

[30] L.G. Connelly and A.E. Bair. Discrete event simulation of Emergency Department activity: A platform for system-level operations research. *Academic Emergency Medicine*, 11(11):1177–1185, 2004.

[31] M. Cooke, J. Fisher, and J. Dale et al. Reducing attendances and waits in Emergency Departments: A systematic review of present innovations. Technical report, 2005. Report to the National Co-ordinating Centre for NHS Service Delivery and Organisation R & D (NCCSDO).

[32] M.W. Cooke, S. Wilson, and S. Pearson. The effect of a separate stream for minor injuries on Accident and Emergency department waiting times. *Emergency Medicine Journal*, 19:28–30, 2002.

[33] M.J. Côté and W.E. Stein. An Erlang-based stochastic model for patient flow. *Omega: The International Journal of Management Science*, 28:347–359, 2000.

[34] A. Darzi. A framework for action. Technical report, NHS London, 2007. `http://www.healthcareforlondon.nhs.uk/framework_for_action.asp`.

[35] R. Davies and H.T.O. Davies. Modelling patient flows and resource provision in health systems. *Omega: The International Journal of Management Science*, 22:123–131, 1994.

[36] Department of Health. Quarterly Monitoring of Key Standards and Targets: Accident and Emergency, England (QMAE), 2006.

[37] A.K. Diehl, M.D. Morris, and S.A. Mannis. Use of Calendar and Weather Data to Predict Walk-In Attendance. *Southern Medical Journal*, 74(6):709–712, 1981.

[38] N.J. Dingle. *Parallel Computation of Response Time Densities and Quantiles in Large Markov and Semi-Markov Models*. PhD thesis, Imperial College London, October 2004.

[39] N.J. Dingle, P.G. Harrison, and W.J. Knottenbelt. Response time densities in Generalised Stochastic Petri Net models. In *Proceedings of the 3rd International Workshop on Software and Performance (WOSP 2002)*, pages 46–54, 2002.

[40] S.R. Earnshaw and S.L. Dennett. Integer/linear mathematical programming models: A tool for allocating healthcare resources. *PharmacoEconomics*, 21(12):839–851, 2003.

[41] T. Eldabi, T. Young, and C. Picton. Simulating A&E systems: More of the same or lessons learned? In *Proceedings of the 2006 OR Society Simulation Workshop (SW 2006)*, 2006.

[42] J. Farrington-Douglas and R. Brooks. The future hospital: The progressive case for change. Technical report, Institute for Public Policy Research (ippr), January 2007. `http://www.ippr.org`.

[43] A.J. Field. *JINQS: An Extensible Library for Simulating Multiclass Queueing Networks*. Department of Computing, Imperial College London, 2006. `http://www.doc.ic.ac.uk/~ajf/Software/Simulation.jar`.

[44] G. Franks and C.M. Woodside. Multiclass multi-servers with deferred operations in layered queueing networks, with software system applications. In *Proc. 12th IEEE/ACM Intl. Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS 2004)*, Volendam, The Netherlands, October 2004.

[45] M. Freimer, G. Mudholkar, G. Kollia, and C. Lin. A study of the generalized Tukey Lambda family. *Communications in Statistics: Theory and Methods*, 17(10):3547–3567, 1988.

[46] The King's Fund. Has the government met the public's priorities for the NHS: A King's fund briefing for the BBC "your NHS" day 2004. Technical report, The King's Fund, 2004.

[47] The King's Fund. Our future health secured? A review of NHS funding and performance. Technical report, The King's Fund, 2007.

[48] W.J. Gordon and G.F. Newell. Closed queueing systems with exponential servers. *Operations Research*, 15(2):254–265, 1967.

[49] P.G. Gould, A.B. Koehler, and F. Vahid-Araghi et al. Forecasting time-series with multiple seasonal patterns. Unpublished, 2007.

[50] N. Gulpinar, P.G. Harrison, B. Rustem, and L-F. Pau. Optimization of a tandem M/GI/1 router network with batch arrivals. In *Proc. Workshop on Perfor-*

mance Modelling, Evaluation and Optimization of Parallel and Distributed Systems (PMEO-PDS 2005), 2005.

[51] P.G. Harrison. On teaching M/G/1 theory with extension to priority queues. *IEE Proceedings - Computers and Digital Techniques*, 147(1):23–26, January 2000.

[52] P.G. Harrison and W.J. Knottenbelt. Passage time distributions in large Markov chains. In *Proc. ACM SIGMETRICS 2002*, Marina Del Rey, California, June 2002.

[53] P.G. Harrison and N.M. Patel. *Performance Modelling of Communication Networks and Computer Architectures*. International Computer Science Series. Addison Wesley, 1993.

[54] A.C. Harvey. *Time Series Models*. Philip Allan, 1981.

[55] A.C. Harvey. *The Econometric Analysis of Time Series*. Philip Allan, 1982.

[56] A.C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, 1989.

[57] Healthcare Commission. Acute hospital portfolio review. Accident and Emergency. Technical report, August 2005.

[58] Healthcare Commission. Key targets for the star ratings 2005/06. Technical report, 2005.

[59] S.A. Jones, M.P. Joy, and J. Pearson. Forecasting Demand of Emergency Care. *Health Care Management Science*, 5:297–305, 2002.

[60] D.R. Holleman Jr, R.L. Bowling, and C. Gathy. Predicting Daily Visits to a Walk-in Clinic and Emergency Department Using Calendar and Weather Data. *Journal of General Internal Medicine*, 11:237–239, 1996.

[61] Z. Karian and E. Dudewicz. *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. CRC Press, Boca Raton, 2000.

[62] L. Kleinrock. *Queueing Systems*, volume 1. John Wiley and Sons, 1975.

[63] W.J. Knottenbelt. Generalised Markovian analysis of timed transition systems. Master's thesis, University of Cape Town, Cape Town, South Africa, July 1996.

[64] W.J. Knottenbelt. *Parallel Performance Analysis of Large Markov Models*. PhD thesis, Imperial College London, London, United Kingdom, February 2000.

[65] M.E. Kuhl, S.G. Sumant, and J.R. Wilson. An automated multiresolution procedure for modeling complex arrival processes. *INFORMS Journal on Computing*, 18(1):3–18, 2006.

[66] M.E. Kuhl, J.R. Wilson, and M.A. Johnson. Estimating and simulating Poisson processes having trends or multiple periodicities. *IIE Transactions*, 29:201–211, 1997.

[67] D. Kwiatkowski, P.C.B. Philips, P. Schmidt, and Y. Shin. Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics*, 54:159–178, 1992.

[68] D. Lane, C. Monefeldt, and J. Rosenhead. Looking in the wrong place for healthcare improvements: A system dynamics study of an Accident and Emergency department. *Journal of the Operational Research Society*, 51(5):518–531, 2000.

[69] A.M. Law and W.D. Kelton. *Simulation Modeling and Analysis*. McGraw Hill, 2000.

[70] A.S. Lebrecht and W.J. Knottenbelt. Response time approximations in fork-join queues. In *Proceedings of the 23rd Annual UK Performance Engineering Workshop (UKPEW 2007)*, July 2007.

[71] J.D.C. Little. A simple proof of L=λW. *Operations Research*, 9:383–387, 1961.

[72] W.A. Massey, G.A. Parker, and W. Whitt. Estimating the parameters of a nonhomogeneous Poisson process with linear rate. *Telecommunication Systems*, 5:361–388, 1996.

[73] L. Mayhew and E. Carney-Jones. Evaluating a new approach for improving care in an Accident and Emergency department: The NU-Care project. Technical report, Cass Business School, City University, 2003.

[74] L. Mayhew and D. Smith. Using queuing theory to analyse completion times in Accident and Emergency times in the light of the government 4-hour target. Technical report, Cass Business School, City University, 2006. Actuarial Research Paper No. 177.

[75] F. McGuire. Using simulation to reduce length of stay in Emergency Departments. In *Proceedings of the 1994 Winter Simulation Conference*, pages 861–867, 1994.

[76] The Met Office. Forecasting the nation's health – an evaluation by the health forecasting unit. Technical report, July 2001.

[77] The Met Office. Forecasting the nation's health – stage II review: an evaluation by the health forecasting unit. Technical report, July 2002.

[78] The Met Office. The 2004/2005 COPD forecasting winter pilot. Technical report, 2005.

[79] The Met Office. Health forecasting for COPD. Technical report, 2006.

[80] Ò. Miró, M. Sánchez, G. Espinosa, B. Coll-Vinent, E. Bragulat, and J. Millá. Analysis of patient flow in the Emergency Department and the effect of an extensive reorganisation. *Emergency Medical Journal*, 20:143–148, 2003.

[81] I. Mitrani. *Simulation techniques for discrete-event systems.* Cambridge Computer Science Texts 14, 1982.

[82] I. Mitrani. *Probabilistic Modelling.* Cambridge University Press, 1998.

[83] M.W. Mulholland, P. Abrahamse, and V. Bahl. Linear programming to optimize performance in a department of surgery. *Journal of the American College of Surgeons*, 200(6):861–868, June 2005.

[84] O. Nenadić and W. Zucchini. Statistical analysis with R – a quick start, September 2004. `http://www.statoek.wiso.uni-goettingen.de/mitarbeiter/ogi/pub/r_workshop.pdf`.

[85] W.K. Newey and K.D. West. A simple positive semi-definite heteroskedascity and autocorrelation consistent covariance matrix. *Econometrica*, 55:703–708, 1987.

[86] K.R. Pattipati, M.M. Kostreva, and J.L. Teele. Approximate Mean Value Analysis algorithms for queueing networks: existence, uniqueness and convergence results. *Journal of the ACM*, 37(3):643–673, July 1990.

[87] H.G. Perros. *Queueing Networks with Blocking*. Oxford University Press, 1994.

[88] D.C. Petrui and C.M. Woodside. Approximate Mean Value Analysis based on Markov chain aggregation by composition. *Linear Algebra and its Applications*, 386:335–358, July 2004.

[89] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1993.

[90] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. `http://www.R-project.org`.

[91] M. Reiser. Mean Value Analysis: A personal account. In *Performance Evaluation: Origins and Directions; Lecture Notes in Computer Science 1769/2000*, pages 491–504, 2000.

[92] M. Reiser and S. Lavenberg. Mean-Value Analysis of closed multiclass queuing networks. *Journal of the ACM*, 27(2):313–322, April 1980.

[93] M.D. Rossetti, G.F. Trzcinski, and S.A. Syverud. Emergency Department simulation and determination of optimal attending physician staffing schedules. In *Proceedings of the 1999 Winter Simulation Conference*, pages 1532–1540, 1999.

[94] A.A. Stinnett and A.D. Paltiel. Mathematical programming for the efficient allocation of health care resources. *Journal of Health Economics*, 15:641–653, 1996.

[95] D. Tandberg and C. Qualls. Time series forecasts of Emergency Department patient volume, length of stay, and acuity. *Annals of Emergency Medicine*, 23(2):299–306, February 1994.

[96] A. Trifunović. *Parallel Algorithms for Hypergraph Partitioning*. PhD thesis, Imperial College London, February 2006.

[97] A. Valmari. *Lecture Notes on Petri Nets I: Basic Models*, volume 1491 of *Lecture Notes in Computer Science*, chapter The State Explosion Problem, pages 429–528. Springer–Verlag, 1998.

[98] H. Wang and K.C. Sevcik. Experiments with improved approximate Mean Value Analysis algorithms. *Performance Evaluation*, 39(1–4):189–206, 2000.

[99] W.T. Weeks. Numerical inversion of Laplace transforms using Laguerre functions. *Journal of the ACM*, 13:419–426, 1966.

[100] E.N. Weiss, M.A. Cohen, and J.C. Hershey. An iterative estimation and validation procedure for specification of semi-Markov models with application to hospital patient flow. *Operations Research*, 30(6):1082–1104, 1982.

[101] E.M. Winands, I.J. Adan, and G.J. van Houtum. Mean Value Analysis for polling systems. *Queueing Systems: Theory and Applications*, 54(1):35–44, September 2006.

[102] Wolfram Research Inc. *Mathematica Edition: Version 5.1.* Champaign, Illinois, 2004.