Abstract

This research develops and evaluates queueing models that can be used to model characteristics of basic call centres, i.e. multi-server systems with time-dependent arrival rates, general service time distribution and state-dependent abandonments on arrival (balking). The discrete-time modelling approach which has previously been used for modelling the time-dependent behaviour of multi-server queues is extended to incorporate state-dependent balking. Pure birth state-dependent arrival processes are studied for different arrival rates and are extended for the case of a recurrent arrival rate. Two approximations are introduced to model time-dependent systems with state-dependent balking. These approximations are proved to bound the actual solution for M(t,n)/D/s systems. A simulation model for systems with state-dependent balking is developed. Empirical tests versus this model show that the two approximations provide bounds of controllable accuracy. The performance of systems with balking is studied. Results show insensitivity to the service time distribution. The pointwise stationary approximation (PSA) generally performs well for these systems. A simple formula to estimate the mean number in the system is derived for busy systems with balking. Insights potentially useful to call centre management are reported.

Contents

List of Tables

List of Figures

Chapter 1

Introduction to queue modelling of call centres

1.1 Introduction

A call centre is any group whose principal business activity is talking on the telephone to customers or prospects. Call centres are central to operations for a broad range of businesses, including travel reservations; product support; order taking; emergency services dispatch; and financial transactions [1].

Due to limited resources and unpredictable demand, not all calls can be answered immediately. However, call centres are equipped with technology which allows calls to wait when needed. For this reason, call centres can be seen as a subgroup of service systems where unmet demands are allowed to wait.

A queueing system is a stochastic system having a service facility at which a population (generally called 'customers') arrives for service, and whenever there are more customers in the system than the service facility can handle simultaneously, a queue, or waiting line develops. Queueing theory is the branch of applied probability theory that studies service systems prone to congestion. This area of study was established almost 100 years ago by the pioneering work of the Danish telephone engineer Agner Krarup Erlang.

In a queueing system the input is an arriving population that enters the system

in order to receive service. The output is the same population that leaves the system before or after receiving service. As a consequence a queueing model defines the interacting processes (arrival, service) and the nature of their interaction, which determines the characteristics of the generated processes (e.g. number of customers in the system, a sequence of customers' delays).

This research is concerned with the use of queueing models for call centres.

1.2 The call centre industry

There is no doubt that call centres are today a booming sector in Britain and all over the world. Having experienced a growth of 250% since 1995 the UK call centre industry is the largest in Europe [2]. At the end of 2003 there were 5,320 call centres in the UK employing 790.000 people [3], [2]. By 2007 it is predicted that call centres in Britain will be employing more than 1 million people [3].

Although some industry analysts predict that call centres will vanish when more people learn to use the internet, others say that these predictions existed some years ago, and were not validated. They argue, that people are not going to report a gas leak by e-mail, or seek advice on a private matter via the internet [2]. This seems to be the case. Although through the internet people do their transactions in a more independent way, they need help when it comes to more complicated needs, than a 'see, buy, pay' process. In fact, the number of call centres is increasing. Worldwide growth in the call services market, averaged about 20% per year, during the past five years, and is expected to continue at a similar rate [4].

The main reason of the recent growth of call centres is that both customers and organisations benefit from this 'remote' service. Indeed, advances in telecommunications and information technology, enable call centres nowadays to be more efficient. For example, computer-telephone integration (CTI) enables automatic identification of the customer's number and thus a search of any information about this customer in a company's database. In this way by the time a customer reaches an agent, the customer's record has appeared at the agent's terminal. This leads to a faster and

more efficient service.

Call centres have become the preferable way of contacting an organisation since they provide fast service (given that they are well managed). The access to service is easier, since there is no need to write or visit an office, or a retail outlet, and often call centres can be contacted outside normal working hours. On the other hand, organisations can also deliver services via call centres with reduced costs, since call centres do not have to be located in an expensive high street location, and because of the support and back up provided by information technology, less expensive staff can be used to handle most routine calls [5].

Call centres can be inbound, i.e. deal with incoming calls, or outbound, i.e. initiate calls to customers, or both. However, most of the call centre industry involves inbound call centres, and the outbound call centre industry is under pressure due to increase in relevant legislation and negative customer views of outbound call centres [3].

Recently some call centres have been referred to as contact centres when they are equipped to deal not only with telephone calls but also with other form of enquiries, for example emails. However, non-telephony interactions (email, web, letter, fax) account for less than 9% of contact centre's activities [3].

In this research we focus on inbound call centres which are the majority of call or contact centres. Whatever the details of the predictions, the call centre industry is well established and is not going to vanish. Indeed, its newness and rate of change mean that there are many management issues worthy of research, as discussed next.

1.3 Call centre management issues

Call centres have significant general management challenges, in human resources (recruitment, absenteeism, emotional support, burnout, call monitoring policies), MIS (multi-user multi-site databases, customer tracking, system integration), training, and quality. As these challenges are better managed and call centres grow larger and more costly, opportunities to use operations research techniques are of increasing interest to the industry. Call centre consultants, and telecommunications firms (e.g. AT&T),

engage in significant non-published research and applications, and there is a need for more public-domain work on the challenges faced by this important industry [4].

Call centre 'workforce' management can be defined as the procedure of matching service requests with resources. Ideally this means having the right number of skilled people and supporting resources in place at the right times to handle an accurately forecasted workload, at agreed service levels with acceptable quality standards. According to Cleveland and Mayben [6], practitioners summarise the above procedure in nine steps.

1. Choose a service level objective (this would be the service level for inbound call centres, e.g. 80% of the calls answered in 20 seconds).

2. Collect data, usually from the automatic call distributor (ACD), but also from other sources such as local networks or voice response units. For example, Mandelbaum, Sakov, and Zeltyn [7] provide a first, in depth, attempt to describe the type of data that are available in call centres. Even though this case study refers to a small bank's call centre, located in Israel, it gives us an idea of the notion of call centre data.

3. Forecast call load in each time block, usually the block's length is between 15 and 60 minutes. The call load includes three factors: average talk time, average after-call work and call volume.

4. Calculate base staff, i.e. for each time block calculate the number of agents needed to meet the service level objective.

5. Calculate trunks and related system resources

6. Calculate rostered staff factor, or shrink factor, or shrinkage. This takes into account breaks, training, and non-phone work.

7. Organise schedules, i.e. assign individual agents to specific shifts.

8. Calculate costs, i.e. since we now know the required number of agents, estimate the budget needed.

9. Repeat for higher or lower level of service depending on whether the costs are permissible or not.

Along similar lines, according to Koole [8], call centre *quantitative* management is about finding the optimal service level to personnel trade-off. So, it has to do with the staffing process, that can be decomposed into five distinct activities [4]:

1. Forecasting (call arrivals by time block and call duration)
2. Performance estimation (predict service level and utilisation for various telephone service representative (TSR) levels in each time block)
3. Staff requirements (select desired number of TSRs in each time block)
4. Shift scheduling (convert staff requirements into shifts, including breaks)
5. Rostering (assign individual people to shifts)

The fourth step in the first procedure and the second in the second one are very important since human resources contribute $70-80\%$ of the total call centre operating cost [5], [9]. Accurate calculation of the base staff will lead to a cost efficient balance between service requests and resources. Thus, it is crucial to use the right models to calculate the staff requirements, and as a result to control agent utilisation.

1.4 Modelling call centres as queueing systems

As mentioned in section 1.1 call centres can be seen as queueing systems. As a result, calls queueing provide a means to measure performance of call centres during operation, and to determine staff needed to serve calls within prespecified service levels during workforce planning.

Figure 1.1 provides a graphical description of a queueing system which could represent a simple call centre, or an elementary unit of a more complex call centre (for example a specific skills unit in a multi-skilled call centre).

Arrivals are calls generated by customers wishing to contact the call centre. Increasingly modern call centres have the policy, when all servers are busy, to inform

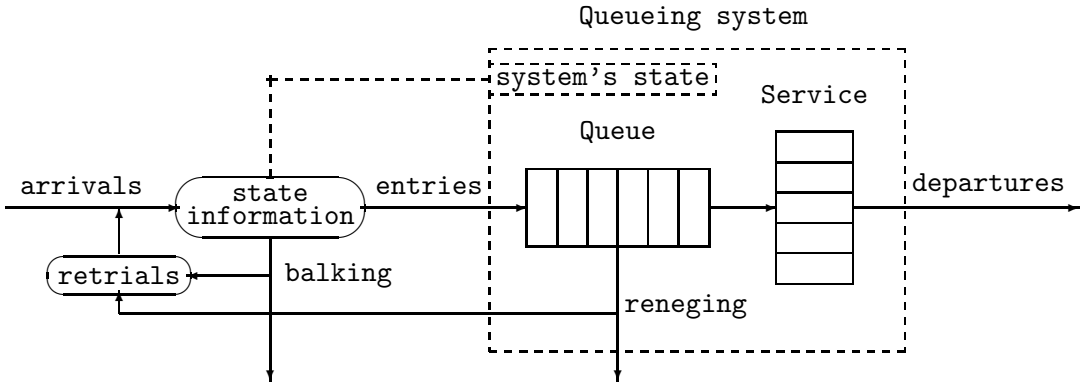


Figure 1.1: Entities and processes that comprise a simple call centre, or an elementary unit of a call centre.

customers on arrival about the current system's state (i.e. announce anticipated delays). The customers then decide either to wait for service, or leave before joining the system, a procedure called balking. While waiting for service they might still decide to abandon the system, a procedure known as reneging. Customers who balk or renege might call back at a later time; this type of call is known as retrial. When a server becomes free he is allocated a customer from the queue, usually with a first-come first-served policy.

It is aknowledged that once the service provider has decided to allow waiting, it is better to inform customers about anticipated delays [10], [11]. The most convincing argument on this is our own experience as customers [11]. Informing customers about anticipated delays tends to cause balking instead of reneging which usually occurs in invisible queues [11]. The decision whether to join the system or balk depends on the system's congestion, thus the announcement of anticipated delays introduces a state-dependent entry process.

The call centre described above assumes homogeneous customers and agents. Though this does not always apply in practice, see for example multi-skilled call centres, the complexity of the problem increases when these assumptions are violated. As we have pointed out before, in this case the above description could refer to a specific skilled group of agents, or to a unit of a complex call centre answering a specific type of calls.

Approximating time-varying behaviour (for example peaks and troughs in arrival rates) by piecewise-constant behaviour, and then applying stationary analysis over

intervals of *constancy* is what is often applied in practice for dealing with the time-variance [12]. This method is problematic since it assumes that each time block is independent of all others, with no backlog carried from one to another. Moreover, calculations used to obtain base staff within each time block, usually involve a classical queueing formula (*Erlang C*) (as reported for example in [6], [8]) which assumes a specific form of service time distribution, ignores abandonments and cannot give any estimation in case of overloads. The above method is popular among practitioners due to its simplicity. However, as this provides a crude approximation there is need for more sophisticated queueing theory models to be employed to shed more light on call centre queue management.

1.5 Aims of the research

In this context the overall aims of this research are:

- To develop one or more queueing models that incorporate important call centres characteristics, and overcome some of the limitations of existing models, such as: time-dependent arrival rates, general service time distribution, balking, and overload.
- To demonstrate the potential value of these models to provide understanding and insights into the behaviour of call centre queues of relevance to call centre management.

1.6 Structure of the rest of the thesis

We note here that because of the massive literature in general queueing systems, the style which is adopted in this thesis is to review relevant sections of the literature at appropriate points of the thesis.

From this chapter we conclude that call centres can be described as queueing systems with time and state-dependent arrival rates, and general service time distributions. For this reason in Chapter 2 we review those models from the relevant

queueing theory literature that have been applied, or could be used to model at least one of these characteristics. We conclude from this review that the discrete-time modelling approach is the most promising method, since it can deal successfully with time-dependent arrival rates, and general service time distributions.

Chapter 3 then describes the theoretical concepts of the discrete-time modelling approach, and reviews in more detail work on this subject. This is done first to show how the discrete-time approach models time-dependent arrival rates and non-exponential service time distributions, and second to show why this approach has the potential to incorporate state-dependent arrivals. Having done so Chapter 4 provides formulae for calculating state-dependent arrivals, and introduces two approximations in order to incorporate these state-dependent arrivals in the discrete-time modelling algorithm.

An investigation is undertaken next to see whether one approximation describes always a more congested system than the other approximation, and whether these two approximations bound the actual solution. Chapter 5 presents some theoretical results. Since deriving theoretical results proved to be difficult and limited, a simulation model was also developed in order to provide a broader investigation of the bounding behaviour of the two approximations. Chapter 6 describes and validates this simulation model. It then uses this simulation model as the exact solution in order to evaluate the two approximations. It also investigates the factors which affect the accuracy of the two approximations.

In Chapter 7 the two approximations are used in order to study the performance of systems with balking and to demonstrate important insights for call centre management and modelling. Finally, Chapter 8 summarises the conclusions of this research, and suggests issues for further research.

Chapter 2

Literature review of queueing theory relevant to call centres characteristics

2.1 Introduction

This chapter is a literature review of queueing research that has been applied or could be used to model call centres.

Since call centres are complex queueing systems we do not expect to find a method which will match all their characteristics. For this reason, we review separately methods that deal successfully with one of their challenging characteristics, in order to investigate possible extensions to include more. Thus, we focus in Section 2.2 on systems with time-dependent arrival rates, in Section 2.3 on systems with abandonments, and in Section 2.4 on systems with general service time distribution.

This review shows the limitations of most analytic approaches. It also indicates the potential success of the discrete-time approach. We review the relevant literature for this latter method in the next chapter, where it is presented and discussed in greater detail as a basis for the research developed in later chapters to model call centres. Finally, in Section 2.5 we justify why numerical methods are still valuable, though clearly simulation also provides a valuable option, especially for complex systems.

2.2 Time-dependent arrival rate

Most of the queueing theory textbooks are concerned with steady-state models, for example the basic textbooks Kleinrock [13], and Gross and Harris [14]. There are also tables and graphs for the steady state distribution of number in the system, and for the expected queue lengths. For example Hillier and Yu [15] cover a wide range of queueing models ($M/M/c$, $M/D/c$, $D/M/c$, $E_k/M/c$, $E_k/E_m/c$). In [16] the more general $G/G/c$ model is considered and, based on the coefficients of variation, tables are given for the steady-state expected number in the queue and the probability of not being able to join the system.

In a call centre scenario, at least the arrival rate varies with time. One could argue that the service rate also varies with time, since a factor of server's fatigue could be introduced. However, this variation is insignificant compared with the variation that is observed in the arrival rates. Different factors, such as advertisements, working hours, e.t.c., trigger more people to call at some times, thus leading to a time-dependent arrival rate.

Most of the attempts to deal with queueing systems with time-dependent arrival rates assume markovian arrivals. This means that the arrival process is Poisson. Poisson processes often occur in reality due to the Palm and Khinchin limit theorem [13]. In call centres, there are many independent and statistically-identical potential customers, who during a small time interval have a small probability of ringing the call centre, so that arrivals should in theory be at random and as such should follow a Poisson process. As explained in the previous paragraph the customers' call probabilities are time-dependent, and thus the arrival process is a time-inhomogeneous Poisson process. Statistical analysis of arrival data from a call centre has showed consistency with this assumption [9].

For the transient behaviour of $M/M/1/\infty$, $M/M/1/1$ and $M/M/\infty$ models closed form solutions exist, and can be found in [14]. However, these solutions are not very useable because they involve Bessel functions making them difficult to apply in practice. Solutions that avoid the use of Bessel functions in the above cases are

given by Sharma [17]. Still, when there is more than one server, the problem becomes too complex. As a result we need to use numerical methods, approximation methods, or simulation [18]. For example, for the $M(t)/M/s(t)$ case, the Chapman-Kolmogorov differential equations can be solved numerically by applying the Runge-Kutta method. Ingólfsson et al. [19] in a survey on approximation methods usually used for $M(t)/M/s(t)$ systems reported, as expected, that gains in accuracy are paid for by excessive computer time.

A method which is often used to model time-dependent behaviour is the diffusion approximation. The method was proposed by Newell [20] for a single server time-dependent queueing system. Since it is simple and flexible it has been very popular (see for example Kleinrock [21], Duda [22]). According to this method discrete queueing processes are represented by continuous diffusion ones. However, heavy-traffic assumptions are needed in order to apply the method. As a result this limits the applicability of the method, and it cannot be applied in a call centre scenario where quiet as well as busy time intervals can be observed.

The simplest approximation method that can be used for the $M(t)/M/s$ model is the simple stationary approximation (SSA). For the SSA to be applied the number of servers needs to be fixed. SSA uses the stationary model with arrival rate equal to the overall mean value of the time varying arrival rate. In this way it loses all the time-dependent nature of the system, however it has been observed [23] that it performs well when the arrival rate changes very rapidly relative to the service times.

Some effects of non-stationarity on multi-server Markovian queueing systems are given in [24]. Even though there is no measure for the degree of non-stationarity in the literature, the authors give a description of what this would be, and are concerned with sinusoidal arrivals rates. For this kind of arrivals, when the number of servers or the arrival volatility from the mean arrival rate increases, the stationary approximation becomes worse.

Another approximation is the pointwise stationary approximation (PSA) introduced by Green and Kolesar [25], which assumes that at each time point steady state is achieved, and uses the instantaneous arrival rate $\lambda(t)$, for the mean arrival rate at

time t. The authors state that if the arrival rate changes sufficiently slowly, relatively to the service times, the PSA gives good approximations.

These two approximations are used because of their simplicity and because they often provide upper and lower bounds of the performance measures. However, there are important cases where both PSA and SSA perform poorly. It is generally admitted that SSA is a crude approximation since it ignores the time-dependent arrival rate, while PSA performs poorly for example when fluctuations on the arrival rate are not sufficiently small (see for example [23]) and cannot be applied in systems in which even temporarily the offered load exceeds the service capacity. For these reasons other approximation methods are required.

One successful approach to deal with time-dependent arrival rate is to apply discrete-time modelling, and use numerical methods to solve the resulting difference equations [26], [27], [28]. We will discuss this method in more detail in Chapter 3.

2.3 Abandonments

It is acknowledged that research into understanding the lost demand due to abandonments would be of great value in managing call centres [4]. Abandonments play a major role in call centres. Customers of call centres demand quick and efficient service, otherwise they abandon the system. Percentages are given to support the above observations [29], derived from a study focused on calls to an airline 's reservation centre, for example:

- Faced with a busy signal, over 30% of callers would not call back.
- Faced with a delay of approximately 15 seconds before being connected with an attendant, 44% of callers abandoned the call and did not call back.
- Faced with a delay of 30 seconds or more before being connected with an attendant, 69% of callers abandoned and did not call back.

There are two ways in which abandonments occur in a queueing system, balking and reneging. Balking occurs when an arriving customer leaves the system as soon as he

realises that he will not be served immediately. In this way he abandons the system immediately after joining it. Reneging occurs when a customer abandons the system after waiting for some time.

Usually call centres experience abandon rates over 10%. Hence if abandonments are taken into account waiting time is no longer appropriate as a single performance measure, but should be used in combination with desired limits on the abandonment rate.

As mentioned in section 1.4 modern call centres are increasingly choosing to inform their customers about anticipated delays. As a result customers decide upon arrival whether to join the system or not. It is expected that having been informed about the time they need to wait until service, if they decide to wait they will wait until they receive service (given that they were given the right information about their expected waiting time). For this reason when we have state-dependent balking (i.e. abandonments which depend on the system's congestion) we will have a negligible percentage of customers reneging (leaving after waiting for some time). The approach adopted in this thesis agrees with what Whitt states in [11] : 'Assuming that customers know their preferences, it is natural to assume that customers would respond to this additional information when all servers are busy by replacing reneging after waiting with state-dependent balking; i.e., customers should be able to decide immediately upon arrival whether or not they are willing to join the queue and wait to receive service. Having joined the queue, customers should be much more likely to remain until they begin service. Reneging is even less likely if the customer can see that the remaining time to wait is steadily declining.' Thus it is important to model state-dependent balking since this is the dominant mechanism of abandonments when information about the expected delays is announced upon arrival.

Whitt [11] assumes stationary arrivals and uses birth-and-death models to study state-dependent balking and reneging. The results of this paper are limited to steady state and negative exponential service time distribution. Brand and Brand [30] studied a $M(n)/M(n)/s + G$ system. Again the service is assumed to follow a negative exponential distribution, though this time it can be state-dependent. Also they can

include balking since they allow for state-dependent entries, and they can have general impatience distribution. Still their results are limited to steady state.

Another approach to model abandonments has been using the idea of customer's 'patience' distribution. The time that someone is willing to spend waiting in an invisible queue, depends on his estimate of the time he has to wait until receiving service, on his patience, on the service benefits, and on the cost of the call. For example when someone uses a toll free line he might wait longer than when he pays for the call, etc.

The problem of taking these factors into account is highlighted in [31], which attempts to derive the customer's distribution, by assuming that customers' estimates of their waiting times coincide with the actual waiting times. In other words it assumes that each customer knows how long he will wait until he steps into service. The basis for this assumption is that a customer has knowledge of the system from previous visits to it. He then decides when he will abandon the system by balancing the service benefits against the cost of waiting. However, the assumption that someone knows how long he has to wait is something that we do not expect to apply in practice, unless this information is is announced in which case case state-dependent balking would occur and not reneging.

Another attempt to deal with abandonments can be found in [32]. This assumes negative exponential service and patience distributions, and steady state, so it is an $M/M/N+M$ model which they call an 'Erlang A' model. In addition this model cannot deal with state-dependent abandonments.

There are very few papers that deal with abandonments and time-dependent behaviour. For example Mandelbaum, Massey, Reiman, Reider and Stolyar [12], [33] use fluid and diffusion approximations to deal with time-dependent systems which face abandonments and retrials. The abandonments in their work concern only reneging and not state-dependent balking, and Mandelbaum and Koole in a subsequent paper [34] recognise that it is not ready for serious applications.

From the above we conclude that there is need for broader models that can be developed to model state-dependent balking which is an important characteristic of

modern call centres.

2.4 Service time distribution

Service time distribution is the distribution of time that the server spends in order to provide service to a customer. In call centres this is the time to answer a call and any rapping up time that the agent will spend after the call to update records in the company's database or to keep notes, thus it is the time duration between answering the call and becoming available again.

Much of the queueing theory literature, and in particular most of the models described in the previous two sections, are concerned with systems with negative exponential service time distributions. This is because analytic steady-state solutions exist for the $M/M/s$ systems with finite or infinite capacity. These solutions are obtained by solving the forward Chapman-Kolmogorov equations. Due to the stationary nature of these systems, the Chapman-Kolmogorov equations can be easily solved, after introducing time invariance.

Call centres are expected to have general service time distributions, and not necessarily a negative exponential one. For example statistical analysis of a particular bank telephone call centre showed that the service time distribution fitted a lognormal distribution [9]. Though in [32] the authors claim that the service time distribution in call centres seems to be lognormal, as this was also observed in another call centre, this clearly does not imply that it can be generalised to all call centres. We could go further and provide a counter example on this, by considering a simple bank call centre that receives two types of customers one requiring a short service, the other requiring a long service. As a result the service time distribution will be bimodal which obviously cannot described by a lognormal distribution since the latter is a unimodal one.

However, there are no analytic models for time-dependent queues when general service distribution is used. There are only some for the limited cases where infinite servers are used (see for example [35], [36]), which are not useful for the multi-server

case that we are interested in. Numerical methods could be employed in this case. These include the phase-type approximation and the discrete-time appoximation.

The phase-type approximation is based on the method of stages which was introduced by Erlang and was generalized by Neuts [37]. According to this method non-exponential distributions are approximated by distributions that are built up from mixtures and/or convolutions of exponential distributions. For example an Erlang k distribution can be represented as k exponential services in series, while a hyperexponential k distribution can be represented by k exponential services in parallel. As a result the phase-type method approximates the non-markovian system with a continuous time Markov chain, in which the state variable includes the phase in which each customer in service is, in addition to the number in the system. However, there is no convenient way of linking this formulation to the real time axis and thus it is difficult to incorporate non-stationary arrivals in these models.

Steady-state results with state-dependent arrivals can be found in Marie and Pellaumail [38] who study a single server system with feedback with the use of Coxian distributions, and Driscoll [39] who uses numerical methods to study an $E_m/E_k/s$ system with state-dependent arrivals.

Other research by Gupta and Rao [40] includes calculation of the steady-state probability distribution in $M(n)/G/1/K$ system. The success of their analysis is due to the fact that the system under consideration has only one server and that their analysis is limited to steady state.

The idea of approximating the non-exponential service time distribution with another distribution is also met in the discrete-time modelling. However according to this approach the continuous general distribution is approximated with a discrete one and the system is observed only at specific moments. Unlike the phase-type approximation discrete-time modelling has been successfully applied to model time-dependent arrival rates, as we will see in more detail in the next chapter.

2.5 Numerical, Approximate and Simulation Models

In addition to the analytical methods already mentioned, simulation also offers an important modelling approach for real queueing problems including call centres [6].

Whilst simulation can provide the only resort for studying some complex systems and a convenient approach in others, as is nicely stated by Marcel Neuts in [26], for models whose structure is mathematically well understood it is desirable that algorithms making use of existing theory be developed. Indeed as stated by Pidd [41] computer simulation is no panacea. Realistic simulations may require long computer programmes of some complexity and producing useful results can turn out to be a surprisingly time-consuming process.

Numerical algorithms provide feasible computability and not the mere formal correctness of transform solutions, that most of the time cannot be applied in practice. Approximation techniques are valuable and this is currently acknowledged from the scientific community. For example we quote from [42], [43] Schweitzer's view on approximations: 'We have reached the end of the road for exact models and future efforts should be devoted to developing better classes of approximation models ... it is better to have an approximate treatment of an accurate model than an accurate treatment of an inaccurate model'.

In this research we develop the discrete-time approach which is a numerical method. We believe that having both analytic (or numerical) and simulation methods promotes powerful modelling. This agrees with Koole and Mandelbaum [34] who also think that one should blend analytic and simulation methods: 'analytical models for insight and calibration, simulation also for fine tuning. In fact, our experience strongly suggests that, having analytical models in one's arsenal, even limited in scope, improves dramatically one's use of simulation'.

2.6 Summary

In conclusion both analytical (or numerical) approaches and simulation have their advantages and disadvantages, and in many practical and research situations the 'best' strategy may well be to use both approaches together. This research tries to fill a gap by developing queueing theory to model call centres incorporating time-dependent and state-dependent arrivals and general service time distribution. Simulation models will be used where appropriate to help validate the queueing models developed.

This thesis will adopt the discrete-time modelling approach which from the literature review of this chapter seems to be the most promising analytical method to model call centres. For this reason we describe and review this approach in the next chapter.

Chapter 3

Discrete-Time Modelling of queueing systems

3.1 Introduction

Call centres are service systems where unmet demands are allowed to wait, therefore they can be described as we have seen in section 1.4 as $M(t)/G/s(t)$ queueing systems. However, it is not possible to find an analytic solution for these systems. By applying discrete-time modelling the above system can be approximated with an appropriate discrete-time system which is tractable.

This chapter provides the background in discrete-time modelling. In particular, section 3.2 gives a brief description of how discrete-time modelling is achieved. Section 3.3 reviews early and pioneer work on discrete-time modelling. Section 3.4 is a literature review on work that deals with the difference equations numerically, while section 3.5 looks at work that deals with the difference equations analytically. The rest of this chapter describes the discrete-time algorithm which is our starting point in order to extend it to include balking. This numerical algorithm was developed here at Lancaster, through the work of previous thesis, and forms the basis for the work undertaken in the remainder of the thesis.

3.2 The discrete-time approach

Discrete-time systems are systems which are accessed (observed and updated) at specific times. For discrete-time modelling of queueing systems the time axis is segmented into a sequence of equal non-overlapping intervals of unit duration, called slots. The points on the time axis which are defined by all multiples of this slot are called epochs (a term introduced by J. Riordan [13]).

In the discrete-time approach we would like to update the state probabilities at one epoch based only on the information we have from the previous epoch, since this would make the calculations easier. This is possible if the system's description at epochs forms a first order Markov chain. In discrete-time modelling this is achieved by defining the basic unit of time (slot) to be equal to the basic unit of service. Thus, the service duration is an integer multiple of slot duration, and the system's state description is then extended to include extra variables which record the residual service times of all ongoing services. This introduces at epochs an embedded Markov chain. Extending the system's description to introduce a Markov chain is reffered to the literature as the supplementary variable technique (see for example [44]).

In this way we deal with any discrete service time distribution, and time dependent arrival rates. In other words we obtain the system's state at time $t+1$, based only on the system's state description at t, and by taking into account all the events (arrivals and departures) that might occur during $(t, t+1]$. This first order Markov chain enables us to derive a set of recurrence equations which, depending on the complexity of the problem, can be solved analytically or numerically.

3.3 Early discrete-time modelling research

The idea of applying discrete-time modelling to queueing systems can be traced to Galliher and Wheeler [45], who studied the $M(t)/D/c$ system. Their service time distribution is deterministic, which can also be seen as a single point discrete distribution. They divide the time axis at intervals equal to the constant service time. For

each time interval they calculate the probability of having n customers in the system, by using the corresponding probabilities for the previous interval and the possible arrivals. They calculate in this way the probabilities of having n customers at each time t, where n can take any integer value.

The idea for using discrete-time approximations to deal with continuous time queues, was introduced in a breakthrough paper by Dafermos and Neuts [46]. The authors suggest that the service times are measured as multiples of an elementary length of time, which could be the unit of time, and hence could be used to define the epochs. They say that this is a reasonable assumption, since it applies in practice in the way we conceive the time, which is in some units, that we name as hours or minutes or seconds, and generally argue extensively for the advantages of analyzing many queues in terms of a discrete-time parameter. The system under consideration in their paper is a single server queue, with stationary arrival rate, and general service time distribution. Using discrete-time modelling they write the recurrence relations, which they say show clearly the dynamics of the system, unlike the other theoretical treatments of this system, where the recurrence relations are hidden under multiple Laplace transforms, and generating functions. These recurrence relations are then solved theoretically.

In a pioneer paper, Neuts [26] suggests that numerical methods could be used to solve the recurrence relations resulting from discrete-time modelling of queueing systems. However the systems under consideration in this paper and in subsequent papers (see Klimko and Neuts [47] and Neuts and Klimko [48]) are limited to single server stationary systems with restricted small numbers of arrivals per time unit.

Minh [49] uses discrete-time modelling to study a single server queue with a time dependent compound poisson arrival process. He uses three variables to define the state of the system: the number of customers in the system, the residual service time, and the number of customers who have arrived in the system. The author uses mainly generating functions, and though he states that the form of the results is suitable for computer applications, there is no evidence to suggest that these expressions can be used to produce numerical results.

3.4 Numerical discrete-time modelling

Dafermos and Neuts not only provided the breakthrough on discrete-time modelling of queueing systems, but also declared that numerical computations is the way to get results useful to practitioners. In [46] they say: 'It is rarely indicated how well suited the basic recurrence relations governing the classical queueing models are for numerical computation. By emphasizing this aspect in the present paper we hope to make a number of readers, with practical interest at heart, more aware of this.'

Their suggestions inspired a series of researchers in Lancaster University, where considerable research in this direction, motivated by Dr. Dave Worthington, has taken place. This includes three theses (Omosigho [50], Brahimi [51], and Wall [52]), and subsequent publications as well as this research.

Omosigho and Worthington [27] motivated by the work of Neuts [26] study the time dependent behaviour of single server queues, with time inhomogeneous arrival rate, and discrete-time service time distribution. This work uses two variables to represent the system's state: the number of customers in the system, and the residual service time. Thus, unlike Minh [49], who probably in an effort to study the departure process, introduced unnecessary complexity to the problem, the resulting equations are much simpler. These equations were then solved numerically to provide the probability distribution of the number in the system. Omosigho and Worthington [53] extend this method to provide an approximation for single-server systems with continuous service time distributions.

Brahimi [51], and Brahimi and Worthington [28] extend this work to multi-server queues. For the first time we have results for queueing systems with more than one server. They provide an exact algorithm for multi-server queueing systems with discrete service time distributions. Using a programming language (Brahimi used Pascal) a computer programme was written to implement this algorithm and calculate the time dependent probability distribution of the number in the system, at each epoch. They also devised a method for approximating continuous service time distributions with discrete ones based on matching moments. This method leads to more

efficient computations and higher accuracy than the one proposed by Omosigho and Worthington [53]. In the rest of this thesis we are going to refer to this discrete-time modelling algorithm as the DTM.

Wall [52] developed DTM to apply for infinite capacity queues and for time dependent number of servers. He also improved the software implementation of DTM by introducing dynamic memory allocation, and by discarding the null elements of the matrix used to store the system's state probabilities, in order to reduce the computational memory demands. Wall [52] and Wall and Worthington [54] also study the time-dependent behaviour of virtual waiting time, i.e. the time that an imaginary customer would have to wait before he receives service if he arrives at the moment under consideration.

Summarizing, the research that has been undertaken in Lancaster, has devised a numerical method, which was called DTM, in order to provide discrete-time modelling of queueing systems with either discrete or continuous service time distributions. DTM is an exact approach for $M(t)/G_D/s(t)$ systems, i.e. systems with discrete service time distribution, providing the distribution of the number in the system at each epoch, and can be used to approximate $M(t)/G/s(t)$ systems, i.e. systems with continuous service time distributions.

In conclusion the multi-server non-stationary $M(t)/G/s$ system cannot be studied using analytical models. However, the DTM algorithm provides a high accuracy approximation for this system, and for the case where the service time distribution is discrete it provides an exact method of modelling this system. Since call centres have more than one server, in this research we will use the DTM method and try to extend it in order to include more call centre characteristics. For this reason in the next sections we describe attempts to solve these models analytically before providing a more detailed description of the numerical DTM algorithm.

3.5 Analytical solutions to discrete-time models

There are two major groups of systems where discrete-time modelling applies. The first group is systems that are continuous in time, but are approximated with discrete-time systems. The second group is systems that are discrete in time due to their nature. These systems occur for example in the field of computers and communications where the natural elementary unit of time can take discrete values only, since it is the machine cycle time of a processor, the bit or byte duration of signals on a channel or transmission line, or the pulse duration of any fixed-size data unit. A recent example is the BISDN (Broadband Integrated Services Digital Network) which is transported by means of discrete units of 53-octet ATM (Asynchronous Transfer Mode) cells.

The increasing interest in the systems mentioned above has resulted in an increasing number of publications on this subject including textbooks [55], [56], [57], [43]. These works are focused on providing analytic solutions to the set of recurrence relations, which can be written at points of time where the embedded Markov chain has been introduced, usually by using generating functions, and Laplace transforms. These methods do not deal with time dependent arrival rates, and transient behaviour, but are limited to steady state. Also most of the times they use discrete arrival process. For example Woodward [43] studies single server queues with geometrical and batch geometrical inter-arrival times, and geometrical service time distributions. Hunter [55] studies $Geo/G/1$ and $G/Geo/1$ systems, while Takagi [56] studies $Geo/G/1$ and $Geo^{[X]}/G_D/1$ with and without server vacations, and the finite population $Geo/G/1//N$ system. All these cases are limited to one server systems, so that analytic calculations are made feasible. Also the results are presented as transform equations, without indicating how they could be applied in practice. Gao, Wittevrongel and Bruneel [58] study a $Geo/Geo/s$ system in discrete-time, however they apply $z-$transforms and thus they limit to steady-state results.

There is also a small number of papers on transient probabilities with balking in discrete time. Again the success in acquiring any exact solutions is due to the fact

that they use single server systems and assume special cases. The ideas appearing in these works could not be extended for the multiple servers systems as they are mainly based on mathematical functions that could only be applied in single server systems. For example Parthasarathy and Selvaraju [59] study a single server system with a specific form of balking and obtain the exact solution due to the use of the confluent hypergeometric function. This function is used while applying transforms to the system, and the solution is only feasible because of the use of this function. If the form of balking changes or there is more than one server, they would not be able to use this function and obtain a solution.

3.6 Approximating continuous service time distributions

DTM deals with $M(t)/G/s$ systems by approximating the continuous service time distribution with a discrete one. Thus, a part of the DTM algorithm concerns this approximation.

In the context of DTM two different methods have been applied for discretising the continuous time service time distribution. Omosigho and Worthington [53] proposed a shape matching of a finite continuous distribution. For this method increasing the number of intervals that are used to represent the continuous distribution leads to better accuracy.

The second method for approximating continuous time distributions is by moment matching [60]. Brahimi and Worthington [28] used this method and have showed that matching the first two moments results in acceptable accuracy for most practical purposes. This second approach requires fewer points to represent the service time distribution (or else number of stages of service) compared with the shape matching. Usually 2, 3, 4, 5, or 6 points are needed depending on the relationship between the variance and the mean of service time, instead of about 20 which are needed for the shape matching approach.

The number of points is important because the Markov chain introduced by DTM, has a number of states that during busy times increases with a factor of $\frac{(s+m-1)!}{s!(m-1)!)}$, where s is the number of servers, and m is the number of points of the discrete service time distribution [51], [54]. It is obvious that the number of states is an increasing function of both m and s. As a result, m is a determinant factor for the computational requirements (computer memory and runtime) of the DTM implementation, and minimising it is crucial, especially when modelling systems with large number of servers. For this reason, moment matching is recommended for discretising the continuous service time distribution. Wall and Worthington [54] showed that the minimum number of points needed for matching the first two moments depends on the size of the discrete interval relative to the mean and also on the squared coefficient of variation. If higher degrees of accuracy are required, higher than the second moment matching should be used for the approximation, since Brahimi [51] has showed that the residual errors of the two moment approximation can mainly be attributed to the unmatched third moment. In this research, two moment matching was used when dealing with continuous time distributions.

3.7 The Markov chain

In the previous section we saw that in the context of DTM $M(t)/G/s$ systems are approximated by $M(t)/G_D/s$ systems. In this section we are concerned with an $M(t)/G_D/s$ system, and we describe how we introduce an embedded Markov chain in order to get a set of recurrence relations which describe this system.

In order to model the time dependent behaviour of a discrete-time system, we need to know the probability distribution of the system's state at each epoch. By setting the time interval between two epochs (i.e. the slot) equal to the basic unit of service, a first order Markov chain can be introduced, and as a result updating the system can be based only on the the previous epoch. However, in order to achieve this, an appropriate definition of the system's state is required. This is because if we describe the system at each epoch just by using a scalar quantity n, which represents

the number of entities in the system, the system's states do not form a Markov chain.

According to DTM, the system's state description is extended to include the remaining stages of service of each customer that receives service. This is done by introducing a vector description, as shown in Figure 3.1, where the service time distribution has m discrete stages, n is the number in the system, and x_i is the number of customers in service that need i units of service until they depart, where residual service times are rounded up to the nearest integer values.

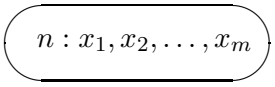


Figure 3.1: Vector state description

In this way we introduce the sequence $[n_t : \mathbf{x}_t]$, where n_t is a random variable describing the number in the system at time t, and $\mathbf{x}_t = [x_1, \ldots, x_m]$ is a vector describing the unfinished service stages of the customers receiving service, at epoch t. The elements of $\mathbf{x}_t$ are random variables, with x_i representing the number of customers who still need i stages of service in order to complete service. The vector $[n_t, \mathbf{x}_t]$ takes finite or countable infinite number of values, thus this description leads to a Markov chain. This Markov chain is described by the following equations:

$$\begin{aligned} n_{t+1} &= n_t - x_1 + r_t \\ \mathbf{x}_{t+1} &= \mathbf{x}_t - [x_1 - x_2, x_2 - x_3, \ldots, x_m] + \mathbf{u} \end{aligned}$$

where r_t is a random variable describing the number of arrivals during $(t, t+1]$, and $\mathbf{u}$ is a vector of random variables that describe the service demands of the customers who start service during $(t, t+1]$.

For this time inhomogeneous, discrete-time Markov chain, the Chapman-Kolmogorov equations can be written down (for example see [26] for the single server model) and give the state probabilities at time t+1, in terms of the state probabilities at time t. In general these recursive relations take the following form:

$$p_{\mathbf{j},\mathbf{k}}^{(0,t+1)} = \sum_{\mathbf{l}} p_{\mathbf{j},\mathbf{l}}^{(0,t)} p_{\mathbf{l},\mathbf{k}}^{(t,t+1)} \tag{3.1}$$

where $p_{\mathbf{j},\mathbf{k}}^{(t_1,t_2)}$ is defined to be the probability that at time t_2 the system is at state $\mathbf{k}$ given that at time t_1 it was at state $\mathbf{j}$.

The first term in the summation on the right hand side is known, if the system's state probability distribution at time t is known. The second term depends on the number of arrivals and departures that occur during one slot, as well as on the service demands of the customers who will start receiving service at epoch $t+1$. However, when we know the vector state of the system at time t, we know how many customers will depart during $(t, t+1]$, since we know how many were in the their last service phase. In this way this last probability is the probability that a certain number of arrivals will occur during a slot, and a certain combination of new service times will be requested.

In conclusion, the problem of finding the probability distribution of the number in the system reduces from integration of the Kolmogorov differential equations, which we would have if we were dealing with continuous time, to solution of state difference equations, where numerical methods are straightforward to apply. The algorithm devised by Brahimi [51] and improved by Wall [52] is introduced next.

3.8 Assumptions and notation

In this section we give the assumptions and notation used while applying DTM for a multi-server queue with time dependent arrival rate and discrete service time distribution.

The following general assumptions are made according to the DTM algorithm:

- The probability distribution of the number of arrivals in any interval can be calculated, and is independent of arrivals in other intervals. This assumption allows for a wide class of arrival processes such as homogeneous and inhomogeneous Poisson processes, and scheduled arrivals.

- The service times of successive customers are independent and identically distributed random variables measured in terms of the elementary unit of time.
- The arrival and service processes are independent.
- When an arriving customer finds all servers busy, he joins a first in first out (FIFO) queue.
- When an arriving customer finds more than one server idle, he is allocated to a free server randomly.

The following notation is used for the description of the DTM algorithm:

n_t	:	Number of customers in the system at time t
s	:	Number of servers in the system
m	:	Maximum number of stages for the service process
$S(i)$	:	Probability that a customer's service demand is i units of time, $i \in \{1, \ldots, m\}$
x_i	:	Number of customers in service whose residual service time, when rounded up to the nearest integer, is i units
r	:	Number of arrivals during a slot
$V_r(t)$	:	Probability of r arrivals during the interval $(t, t+1]$

3.9 Description of DTM algorithm

In this section we give a brief description of the DTM algorithm as introduced by Brahimi [51].

Initially the general continuous service time distribution is discretised by representing it by m equally spaced values: $u, 2u, \ldots, mu$, and by assigning a proper probability to each of these values. In this way the service procedure is split into m stages. The time unit is set to be equal to the time interval of the basic service stage (i.e. $u = 1$). If we know the number in the system, and the remaining number of

service stages that each customer in service requires, we have a full description of the current system's state and due to the markovian arrivals all possible system's states which can be reached at the next epoch can be calculated.

This is done by first removing the service completions and updating the remaining residual service times. The probabilities that we have $0, 1, 2, \ldots, r$ arrivals in the unit interval are then calculated (i.e. $V_0(t), V_1(t), V_2(t), \ldots$). The algorithm then expands like a tree, having as a root the current state, and as branches the states that arise if we take into account the different possible arrivals. However, the new states are not specified completely. We also have to incorporate the new service times of customers who start service by taking into account all the possible combinations of service time demands that could occur. In this way we calculate iteratively the system's state probability distribution at successive time points.

We will now demonstrate how the DTM algorithm applies for each time step. Let us assume that we are at time t and we are trying to calculate the state probabilities at time $t + 1$. We consider one by one all the states at time t and we update them as described above to calculate the probabilities of the resultant states that might occur. We will show how the algorithm works for one of them. Suppose we consider the state $[n_t : x_1, x_2, \ldots, x_m]$.

- First we need to remove from the system customers that complete their service before time $t+1$. Since there were x_1 customers at time t with one remaining unit of service, at time $t + 1$ these customers will leave the system. Also customers that were in service will now reduce their remaining service time by one unit. For this reason the resultant state is $[n_t - x_1 : x_2, x_3, \ldots, x_m, 0]$.

- Then the number r of possible arrivals during $(t, t+1)$ needs to be added in the system so the resultant state is $[n_{t+1} = n_t - x_1 + r : x_2, \ldots, x_m, 0]$, where r can take any non-negative integer value.

- The number of free places in the service that can accept new customers is then calculated as $newc = \min\{c, n_{t+1}\} - \sum_{i=2}^{m} x_i$. Each of the $newc$ customers who start service will need i units of service, where i is a random variable,

$i \in \{1, \ldots, m\}$. In this way we create vectors of new service times $(z_1, \ldots, z_m)$, where z_i is the number of customers starting service who need i stages of service. For this reason $newc = \sum_{i=1}^{m} z_i$. Each of these vectors $\mathbf{z} = (z_1, \ldots, z_m)$ has an associated probability, which can be calculated by taking into account all possible combinations of service requests. It is:

$$P(\mathbf{z}) = prob(z_1, \ldots, z_m) = \frac{newc!}{z_1! \ldots z_m!} S(1)^{z_1} \ldots S(m)^{z_m}$$

where $S(i)$ is the probability that the service time will last i units of time.

- The final states are now of the form $[n_{t+1} : x_2 + z_1, \ldots, x_m + z_{m-1}, z_m]$ and the probability of reaching each of these states is the probability of being initially at state $[n_t : x_1, \ldots, x_m]$, multiplied by the probability of r arrivals, multiplied by the probability of having $[z_1, \ldots, z_m]$ new service demands. This is:

$$P_{t+1}[n_t - x_1 + r : x_2 + z_1, \ldots, x_m + z_{m-1}, z_m] = P_t[n_t : x_1, \ldots, x_m] \times V_r(t) \times P(\mathbf{z}) \quad (3.2)$$

According to this forward algorithm, starting from a specific state at time t, Equation (3.2) gives its contribution to the probability of a resultant state at time $t + 1$. We should repeat this calculation for all possible resultant states. Then, by sweeping all possible states at time t we can find all possible states at time $t + 1$. Each time a resulting state has an associated probability (i.e. was also resulting state from a previous initial state), the latest probability contribution is accumulated to the previous one. In this way at the end of this procedure we have the system's state probability distribution at time $t+1$. Having found the system's state distribution at time $t+1$ we use it as a starting point in order to find the system's state distribution at the next epoch, i.e. time $t + 2$.

3.10 Summary

In this chapter we have described the discrete-time modelling of queueing systems. According to this method the description of the systems under consideration is expressed fully by a system of difference equations. We have reviewed the literature of the two major categories used to solve these equations, i.e. the numerical and the analytic methods. We have also described the DTM algorithm which we are going to use in this research. We are next going to investigate whether the DTM theory that up to now has been used successfully to model $M(t)/G/s(t)$ systems, can be extended to model systems with balking. We will see how this is achieved in the next chapter.

Chapter 4

Extending the DTM theory to include state-dependent balking

4.1 Introduction

The aim of this chapter is to develop the DTM theory described in chapter 3 in order to include balking. The rate at which balking occurs depends on the state of the system, i.e. on the number of customers in the system. In practical terms this corresponds to informing incoming calls about their expected waiting time, or their position in the queue, so depending on how long this is, they will either enter the system, or hang up immediately which is called balking. In this way, from a formulation point of view, balking is about introducing state-dependent entries in DTM.

The entry process is assumed to be a state-dependent Poisson process, in which the arrival rate changes when an arrival manages to join the system and when a departure occurs. This chapter considers the entry process in two stages. In Sections 4.2-4.5 theory is developed for Poisson processes where arrival rates only change due to arrivals, i.e. the effect of departures is ignored. Then in Section 4.6 two approximate methods for introducing departures are described.

When events occur as a Poisson process at constant rate, it is well known that the number of events during time T follows a Poisson distribution. However when

the arrival rate changes when arrivals occur, the number of arrivals during time T will depend on the starting state x, and will not follow a Poisson distribution. In section 4.2 these state-dependent entries are described while in section 4.3 a theorem is presented which allows their probabilities to be calculated. In section 4.4 another way of calculating these state-dependent entries is presented. This is motivated by the relevant literature and it leads to a different formula, however both formulae give the same results as expected. Neither of these formulae can deal with a recurrent arrival rate, thus section 4.5 extends the previous theorems to deal with this case.

Having calculated the state-dependent entry probabilities we want to incorporate them in the DTM algorithm. This is done in section 4.6 by introducing two approximations.

4.2 State-dependent Poisson processes

A Poisson process is a stochastic process in which events occur at random at some constant rate λ, i.e. $prob(event\ in\ [t,\ t+\delta t]) = \lambda\delta t + o(\delta t)$, where $o(\delta t)$ denotes any function that goes to zero with δt faster than δt itself. For such a process it is well known that the number of events during time T follows a Poisson distribution with mean λT. A state-dependent Poisson process is a stochastic process in which events occur at random at a rate λ_x, where x is the current state, i.e. $prob(event\ in\ [t,\ t+\delta t]/state\ x) = \lambda_x\delta t + o(\delta t)$.

Our interest is a state-dependent Poisson process, in which the state x changes after every event. For such processes the number of events during time T will depend on the starting state x, and will not follow a Poisson distribution. This sort of process is important in a wide range of applications in addition to the call centre problem studied here. For example, state-dependence is important to include when dynamic routing strategies are being considered [61]. Another queueing system, where state dependent arrivals occur is the breakdowns from a limited population, i.e. the machine interference problem (see for example [14]). Also state-dependent Poisson arrivals occur while modelling a computing facility dedicated to batch-job processing, where

job submissions are discouraged when the facility is heavily used [59]. Nevertheless, the general framework of this situation is the state-dependent occurrences of random events and is not limited to queueing systems. Examples include market penetration in limited population, successful orders on depleting stock, epidemic models that describe the spread of a disease, and multi-cast calls in wavelength-routing networks [62].

4.3 Calculation of state-dependent entry probabilities

In this section we derive a formula for the probability distribution of number of arrivals from a state-dependent Poisson process in time T. We therefore consider a system in which the only events are state-dependent arrivals, and the state of the system is the number of customers in the system.

Let us assume that at time t the system is in state x. The probability of finding the system in state y after time T, is the probability that exactly $y - x$ arrivals will occur, during T. Each time an arrival occurs, the system's state changes, and this affects the probability of a new arrival. The following theorem is derived using the fact that for the system to move from state x to state y, exactly k arrivals should occur, where $k = y - x$, and the system's successive states are $x_1(= x), x_2, \ldots, x_{k+1}(= y)$.

Let z_1 be the time after t at which the first arrival occurs, and z_i the inter-arrival time between arrival $(i-1)$ and arrival i, for $i > 1$. The joint density function of these variables $f(z_1, z_2, \ldots, z_k)$ equals the product of the density function of each variable, since they are independent. Because arrivals occur as a Poisson process with rate λ_{x_i}, each random variable z_i for $i > 1$ is exponentially distributed, and its probability density function is $\lambda_{x_i} e^{-\lambda_{x_i} z_i}$. The random variable z_1 is not an inter-arrival time, however, because of the memoryless property of the exponential distribution, its density function will have the same form with others.We now prove the following theorem:

Theorem 4.1

$$P(x_{k+1}|x_1,\ during\ T) = e^{-\lambda_{x_{k+1}}T}\lambda_{x_1}\lambda_{x_2}\cdots\lambda_{x_k}Rec_{x_1}(\lambda_{x_{k+1}},\lambda_{x_k}), \qquad (4.1)$$

where in general $Rec_{x_1}(\lambda_{x_p},\lambda_{x_q})$ *for* $p>q$ *is a recursive function defined by:*

$$Rec_{x_1}(\lambda_{x_p},\lambda_{x_q})=\begin{cases} \frac{1}{\lambda_{x_p}-\lambda_{x_1}}(e^{(\lambda_{x_p}-\lambda_{x_1})T}-1), & for\ q=1 \\ \frac{1}{\lambda_{x_p}-\lambda_{x_q}}\left[e^{(\lambda_{x_p}-\lambda_{x_q})T}Rec_{x_1}(\lambda_{x_q},\lambda_{x_{q-1}})-Rec_{x_1}(\lambda_{x_p},\lambda_{x_{q-1}})\right], & for\ q>1 \end{cases} \qquad (4.2)$$

Proof 4.1

$$\begin{aligned} P(x_{k+1}|x_1) =& P(exactly\ k\ events\ during\ time\ T) \\ =& P\left(\{0<z_1<T\}\cap\{0<z_2<T-z_1\}\cap\ldots\cap\{0<z_k<T-z_1-\ldots-z_{k-1}\}\cap\right. \\ & \left.\cap\{T-z_1-\ldots-z_k<z_{k+1}<\infty\}\right) \\ =& \int_0^T\int_0^{T-z_1}\cdots\int_0^{T-z_1-\ldots-z_{k-1}}\int_{T-z_1-\ldots-z_k}^{\infty} f(z_1,z_2,\ldots,z_k,z_{k+1})dz_{k+1}dz_k\cdots dz_2dz_1 \\ =& \int_0^T\int_0^{T-z_1}\cdots\int_0^{T-z_1-\ldots-z_{k-1}}\int_{T-z_1-\ldots-z_k}^{\infty} f_1(z_1)f_2(z_2)\ldots f_k(z_k)f_{k+1}(z_{k+1})dz_{k+1}dz_k\cdots dz_2dz_1 \\ =& \int_0^T\int_0^{T-z_1}\cdots\int_0^{T-z_1-\ldots-z_{k-1}}\int_{T-z_1-\ldots-z_k}^{\infty} \lambda_{x_1}e^{-\lambda_{x_1}z_1}\lambda_{x_2}e^{-\lambda_{x_2}z_2}\cdots\lambda_{x_k}e^{-\lambda_{x_k}z_k}\lambda_{x_{k+1}}e^{-\lambda_{x_{k+1}}z_{k+1}} \\ & dz_{k+1}dz_k\cdots dz_2dz_1 \end{aligned}$$

$$\begin{aligned} & \{calculating\ the\ innermost\ integral\} \\ =& \int_0^T\int_0^{T-z_1}\cdots\int_0^{T-z_1-\ldots-z_{k-1}} \lambda_{x_1}e^{-\lambda_{x_1}z_1}\lambda_{x_2}e^{-\lambda_{x_2}z_2}\cdots\lambda_{x_k}e^{-\lambda_{x_k}z_k}e^{-\lambda_{x_{k+1}}(T-z_1-\ldots-z_k)}dz_k\cdots dz_2dz_1 \\ =& e^{-\lambda_{x_{k+1}}T}\int_0^T\int_0^{T-z_1}\cdots\int_0^{T-z_1-\ldots-z_{k-1}} \lambda_{x_1}e^{(\lambda_{x_{k+1}}-\lambda_{x_1})z_1}\lambda_{x_2}e^{(\lambda_{x_{k+1}}-\lambda_{x_2})z_2}\cdots\lambda_{x_k}e^{(\lambda_{x_{k+1}}-\lambda_{x_k})z_k}dz_k\cdots dz_2dz_1 \\ =& e^{-\lambda_{x_{k+1}}T}I_k(\lambda_{x_{k+1}},\lambda_{x_1}) \end{aligned}$$

where we define:

$$I_k(\lambda,\lambda_{x_1}) = \int_0^T\int_0^{T-z_1}\cdots\int_0^{T-z_1-\ldots-z_{k-1}} \lambda_{x_1}e^{(\lambda-\lambda_{x_1})z_1}\lambda_{x_2}e^{(\lambda-\lambda_{x_2})z_2}\cdots\lambda_{x_k}e^{(\lambda-\lambda_{x_k})z_k}dz_k\cdots dz_2dz_1$$

To prove the theorem we will now prove by induction that

$$I_k(\lambda, \lambda_{x_1}) = \lambda_{x_1}\lambda_{x_2}\cdots\lambda_{x_k} Rec_{x_1}(\lambda, \lambda_{x_k}), \qquad \text{for} \quad k \geq 1, \tag{4.3}$$

For $k = 1$ we have:

$$\begin{aligned} I_1(\lambda, \lambda_{x_1}) = \int_0^T \lambda_{x_1} e^{(\lambda-\lambda_{x_1})z_1} dz_1 &= \lambda_{x_1} \tfrac{1}{\lambda-\lambda_{x_1}} (e^{(\lambda-\lambda_{x_1})T} - 1) \\ \{\text{from Equation (4.2)}\} &= \lambda_{x_1} Rec_{x_1}(\lambda, \lambda_{x_1}) \end{aligned}$$

which is what we require according to Equation (4.3) for $k = 1$.

Let us assume that Equation (4.3) is valid for k.We will prove that it is valid for $k+1$, that is:

$$I_{k+1}(\lambda, \lambda_{x_1}) = \lambda_{x_1}\lambda_{x_2}\cdots\lambda_{x_{k+1}} Rec_{x_1}(\lambda, \lambda_{x_{k+1}}) \tag{4.4}$$

We start from the left side of Equation (4.4). From its definition:

$$\begin{aligned}
&I_{k+1}(\lambda, \lambda_{x_1}) = \\
&= \int_0^T \int_0^{T-z_1} \cdots \int_0^{T-z_1-\ldots-z_k} \lambda_{x_1} e^{(\lambda-\lambda_{x_1})z_1} \lambda_{x_2} e^{(\lambda-\lambda_{x_2})z_2} \cdots \lambda_{x_{k+1}} e^{(\lambda-\lambda_{x_{k+1}})z_{k+1}} dz_{k+1} \cdots dz_2 dz_1 \\
&\{\text{calculating the innermost integral}\} \\
&= \int_0^T \cdots \int_0^{T-\ldots-z_{k-1}} \lambda_{x_1} e^{(\lambda-\lambda_{x_1})z_1} \cdots \lambda_{x_k} e^{(\lambda-\lambda_{x_k})z_k} \frac{\lambda_{x_{k+1}}}{\lambda-\lambda_{x_{k+1}}} (e^{(\lambda-\lambda_{x_{k+1}})(T-\ldots-z_k)} - 1) dz_k \cdots dz_1 \\
&= \frac{\lambda_{x_{k+1}}}{\lambda-\lambda_{x_{k+1}}} \left[e^{(\lambda-\lambda_{x_{k+1}})T} \int_0^T \cdots \int_0^{T-z_1-\ldots-z_{k-1}} \lambda_{x_1} e^{(\lambda x_{k+1}-\lambda_{x_1})z_1} \cdots \lambda_{x_k} e^{(\lambda_{x_{k+1}}-\lambda_{x_k})z_k} dz_k \cdots dz_1 \right. \\
&\left. - \int_0^T \cdots \int_0^{T-z_1-\ldots-z_{k-1}} \lambda_{x_1} e^{(\lambda-\lambda_{x_1})z_1} \cdots \lambda_{x_k} e^{(\lambda-\lambda_{x_k})z_k} dz_k \cdots dz_1 \right] \\
&\left\{\text{using the definition of } I_k(\lambda, \lambda_{x_1}) \text{ for } \lambda = \lambda_{x_{k+1}} \text{ and } \lambda = \lambda \text{ respectively}\right\} \\
&= \frac{\lambda_{x_{k+1}}}{\lambda-\lambda_{x_{k+1}}} \left[e^{(\lambda-\lambda_{x_{k+1}})T} I_k(\lambda_{x_{k+1}}, \lambda_{x_1}) - I_k(\lambda, \lambda_{x_1}) \right] \\
&\{\text{applying Equation (4.3) for } \lambda = \lambda_{x_{k+1}} \text{ and } \lambda = \lambda \text{ respectively}\} \\
&= \frac{\lambda_{x_{k+1}}}{\lambda-\lambda_{x_{k+1}}} \left[e^{(\lambda-\lambda_{x_{k+1}})T} \lambda_{x_1}\lambda_{x_2}\cdots\lambda_{x_k} Rec_{x_1}(\lambda_{x_{k+1}}, \lambda_{x_k}) - \lambda_{x_1}\lambda_{x_2}\cdots\lambda_{x_k} Rec_{x_1}(\lambda, \lambda_{x_k}) \right] \\
&= \frac{\lambda_{x_1}\cdots\lambda_{x_k}\lambda_{x_{k+1}}}{\lambda-\lambda_{x_{k+1}}} \left[e^{(\lambda-\lambda_{x_{k+1}})T} Rec_{x_1}(\lambda_{x_{k+1}}, \lambda_{x_k}) - Rec_{x_1}(\lambda, \lambda_{x_k}) \right] \\
&\{\text{from the definition of } Rec_{x_1}(\lambda_{x_p}, \lambda_{x_q}) \; for \; \lambda_{x_p} = \lambda, q = k+1\} \\
&= \lambda_{x_1} \cdots \lambda_{x_{k+1}} Rec_{x_1}(\lambda, \lambda_{x_{k+1}})
\end{aligned}$$

which is the right hand side of Equation (4.4), as required. Q.E.D.

■

Theorem 4.1 thus gives the exact probability for going from state x_1 to x_{k+1} during time t. The values of the Rec_x function can be calculated quite easily, in terms of a computer programme, using the recursive function.

In order to have an idea of the shape of the state dependent distributions, we give in Figure 4.1 an example of these distributions. In this example we look at a finite population system ($N = 16$) with arrival rate $\lambda(n) = 0.2(16 - n)$. Four probability distributions are presented depending on the initial state of the system. These distributions represent the probabilities of going from an initial state which in our example takes the values 0, 4, 8, or 12 to higher (resulting) states. We can see that the distributions have different shapes depending on the initial state in the system.

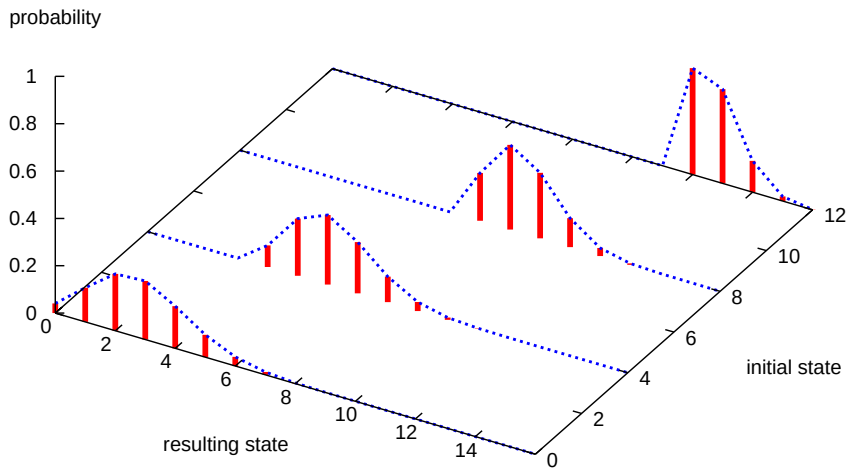


Figure 4.1: The resulting distribution when the initial state is 0, 4, 8, 12 and $\lambda(n) = 0.2(16 - n)$, T=1.

4.4 Extending Pure Birth Processes

An alternative approach to this same problem is suggested (but not proved) by Feller [63], as an extension to pure birth processes. We develop the alternative proof next.

Theorem 4.2 *Suppose that a system passes through a sequence of states $E_0 \rightarrow E_1 \rightarrow \cdots$, staying at E_k for a sojourn time $\mathbf{X_k}$. If $\mathbf{X}_j$ has a probability density function $\lambda_j e^{-\lambda_j x}$ for $j = 0, \ldots, n$ where $\lambda_j \neq \lambda_k$ unless $j = k$, then $\mathbf{S}_n = \mathbf{X}_0 + \ldots + \mathbf{X}_n$ (that is the epoch of the transition $E_n \rightarrow E_{n+1}$) has a probability density function given by*

$$P_n(t) = \lambda_0 \cdots \lambda_n [\psi_{0,n} e^{-\lambda_0 t} + \ldots + \psi_{n,n} e^{-\lambda_n t}] \tag{4.5}$$

where:

$$\psi_{k,n} = [(\lambda_0 - \lambda_k) \cdots (\lambda_{k-1} - \lambda_k)(\lambda_{k+1} - \lambda_k) \cdots (\lambda_n - \lambda_k)]^{-1} \tag{4.6}$$

We prove this by induction.

Proof 4.2 For n=1 the pdf of $\mathbf{X}_0 + \mathbf{X}_1$, is the convolution of the individual pdfs

$$\begin{aligned}
P_1(t) &= \lambda_0 e^{-\lambda_0 x} \star \lambda_1 e^{-\lambda_1 x} \\
&\quad \{ \text{using Equation (A.2) from Appendix A} \} \\
&= \frac{\lambda_0 \lambda_1}{\lambda_1 - \lambda_0} (e^{-\lambda_0 x} - e^{-\lambda_1 x}) \\
&= \lambda_0 \lambda_1 \left(\frac{1}{\lambda_1 - \lambda_0} e^{-\lambda_0 x} + \frac{1}{\lambda_0 - \lambda_1} e^{-\lambda_1 x} \right) \\
&= \lambda_0 \lambda_1 \left(\psi_{0,1} e^{-\lambda_0 x} + \psi_{1,1} e^{-\lambda_1 x} \right)
\end{aligned}$$

We now assume that Equation (4.5) is valid for n, that is the summation of n random variables $\mathbf{X}_j$, with pdf $\lambda_j e^{-\lambda_j x}$, has a pdf that equals the product of the coefficients λ_j times the summation of the products of the exponentials with the corresponding coefficients ψ. Hence if we are interested in the summation of

$\mathbf{X}_0 + \ldots + \mathbf{X}_{n-1}$, the pdf will be given by:

$$P_{n-1}(t) = \lambda_0 \cdots \lambda_{n-1}[\psi_{0,n-1}e^{-\lambda_0 t} + \ldots + \psi_{n-1,n-1}e^{-\lambda_{n-1}t}] \tag{4.7}$$

while if we are interested in the summation of $\mathbf{X}_1 + \ldots + \mathbf{X}_n$, the pdf will be given by:

$$P'_{n-1}(t) = \lambda_1 \cdots \lambda_n[\psi'_{1,n}e^{-\lambda_1 t} + \ldots + \psi'_{n,n}e^{-\lambda_n t}] \tag{4.8}$$

where $\psi'_{k,n}$ is given by Equation (4.6) without the term $(\lambda_0 - \lambda_k)$ since $\mathbf{X}_0$ was not included, i.e. $\psi'_{k,n} = (\lambda_0 - \lambda_k)\psi_{k,n}$.

We will now prove that the pdf for $n+1$ variables has the same form, that is the pdf of $\mathbf{X}_0 + \ldots + \mathbf{X}_n$ is given by:

$$P_n(t) = \lambda_0 \cdots \lambda_n[\psi_{0,n}e^{-\lambda_0 t} + \ldots + \psi_{n,n}e^{-\lambda_n t}]$$

By definition:

$$\begin{aligned}
P_n(t) = \;& pdf(\mathbf{X}_0 + \ldots + \mathbf{X}_{n-1}) \star pdf(\mathbf{X}_n) = \{using\ (4.7)\} \\
&= \left[\lambda_0 \cdots \lambda_{n-1}(\psi_{0,n-1}e^{-\lambda_0 t} + \ldots + \psi_{n-1,n-1}e^{-\lambda_{n-1}t})\right] \star \left[\lambda_n e^{-\lambda_n t}\right] \\
&= \left(\lambda_0 \cdots \lambda_{n-1}\psi_{0,n-1}e^{-\lambda_0 t}\right) \star \left(\lambda_n e^{-\lambda_n t}\right) + \ldots + \left(\lambda_0 \cdots \lambda_{n-1}\psi_{n-1,n-1}e^{-\lambda_{n-1}t}\right) \star \left(\lambda_n e^{-\lambda_n t}\right) \\
&\{\text{using Equation (A.2) from Appendix A }\} \\
&= \lambda_0 \cdots \lambda_{n-1}\lambda_n \left[\frac{\psi_{0,n-1}}{\lambda_n - \lambda_0}(e^{-\lambda_0 t} - e^{-\lambda_n t}) + \ldots + \frac{\psi_{n-1,n-1}}{\lambda_n - \lambda_{n-1}}(e^{-\lambda_{n-1}t} - e^{-\lambda_n t})\right] \\
&\left\{\text{From the definition of } \psi_{k,n}\ (4.6) \Rightarrow \psi_{k,n} = \frac{\psi_{k,n-1}}{\lambda_n - \lambda_k}\right\} \\
&= \lambda_0 \cdots \lambda_n \left[\psi_{0,n}(e^{-\lambda_0 t} - e^{-\lambda_n t}) + \ldots + \psi_{n-1,n}(e^{-\lambda_{n-1}t} - e^{-\lambda_n t})\right] \\
&= \lambda_0 \cdots \lambda_n \left[\psi_{0,n}e^{-\lambda_0 t} + \ldots + \psi_{n-1,n}e^{-\lambda_{n-1}t} - \left(\sum_{k=0}^{n-1}\psi_{k,n}\right)e^{-\lambda_n t}\right] \qquad (4.9)
\end{aligned}$$

However, addition is associative, so instead of calculating the summation $(\mathbf{X}_0 + \ldots + \mathbf{X}_{n-1}) + \mathbf{X}_n$ we can calculate the summation $\mathbf{X}_0 + (\mathbf{X}_1 + \ldots + \mathbf{X}_n)$. While the grouping of the random variables is different, so the convolution includes different terms,

the result should be the same, since nothing has actually changed. Thus we also have:

$$
\begin{aligned}
P_n(t) = \;& pdf(\mathbf{X}_0) \star pdf(\mathbf{X}_1 + \ldots + \mathbf{X}_n) = \{using\ (4.8)\} \\
&= \lambda_0 e^{-\lambda_0 t} \star \left(\lambda_1 \cdots \lambda_n [\psi'_{1,n} e^{-\lambda_1 t} + \ldots + \psi'_{n,n} e^{-\lambda_n t}] \right) \\
&= \left(\lambda_0 e^{-\lambda_0 t} \right) \star \left(\lambda_1 \cdots \lambda_n \psi'_{1,n} e^{-\lambda_1 t} \right) + \ldots + (\lambda_0 e^{-\lambda_0 t}) \star \left(\psi'_{n,n} e^{-\lambda_n t} \right) \\
&\{\text{using Equation (A.2) from Appendix A }\} \\
&= \lambda_0 \cdots \lambda_n \left[\frac{\psi'_{1,n}}{\lambda_0 - \lambda_1} (e^{-\lambda_1 t} - e^{-\lambda_0 t}) + \ldots + \frac{\psi'_{n,n}}{\lambda_0 - \lambda_n} (e^{-\lambda_n t} - e^{-\lambda_0 t}) \right] \\
&\{\text{As before } \psi_{k,n} = \frac{\psi'_{k,n}}{\lambda_0 - \lambda_k}\} \\
&= \lambda_0 \cdots \lambda_n \left[\psi_{0,n} (e^{-\lambda_0 t} - e^{-\lambda_n t}) + \ldots + \psi_{n-1,n} (e^{-\lambda_{n-1} t} - e^{-\lambda_n t}) \right] \\
&= \lambda_0 \cdots \lambda_n \left[\psi_{1,n} e^{-\lambda_1 t} + \ldots + \psi_{n,n} e^{-\lambda_n t} - \left(\sum_{k=1}^{n} \psi_{k,n} \right) e^{-\lambda_0 t} \right] \qquad (4.10)
\end{aligned}
$$

Comparing (4.9) and (4.10), since these two equations have to be the same for any value of t, we have that $\psi_{0,n} = -\left(\sum_{k=1}^{n} \psi_{k,n}\right)$ and $\psi_{n,n} = -\left(\sum_{k=0}^{n-1} \psi_{k,n}\right)$, both of which imply $\sum_{k=0}^{n} \psi_{k,n} = 0$. Replacing this in either of the above formulae we have (e.g. from (4.9)):

$$
P_n(t) \;=\; \lambda_0 \cdots \lambda_n \left[\psi_{0,n} e^{-\lambda_0 t} + \ldots + \psi_{n-1,n} e^{-\lambda_{n-1} t} + \psi_{n,n} e^{-\lambda_n t} \right]
$$

which is what we wanted to prove. ■

Using the above theorem we can now calculate the probability of having exactly n arrivals during T. It is:

$Prob$[exactly n arrivals by T] =

$$
\begin{aligned}
&= Prob[\text{at least n arrivals by T}] - Prob[\text{at least n+1 arrivals by T}] \\
&= Prob[\mathbf{X}_0 + \ldots + \mathbf{X}_{n-1} \leq T] - Prob[\mathbf{X}_0 + \ldots + \mathbf{X}_n \leq T] \\
&= \int_0^T P_{n-1}(t)dt - \int_0^T P_n(t)dt \\
&\{\text{using Theorem 4.2}\} \\
&= \lambda_0 \cdots \lambda_{n-1} \left[\psi_{0,n-1} \int_0^T e^{-\lambda_0 t} dt + \ldots + \psi_{n-1,n-1} \int_0^T e^{-\lambda_{n-1} t} dt \right] - \\
&- \lambda_0 \cdots \lambda_n \left[\psi_{0,n} \int_0^T e^{-\lambda_0 t} dt + \ldots + \psi_{n,n} \int_0^T e^{-\lambda_n t} dt \right] \\
&= \lambda_0 \cdots \lambda_{n-1} \left[(\psi_{0,n-1} - \lambda_n \psi_{0,n}) \int_0^T e^{-\lambda_0 t} dt + \ldots + \right. \\
&+ \left. (\psi_{n-1,n-1} - \lambda_n \psi_{n-1,n}) \int_0^T e^{-\lambda_{n-1} t} dt - \lambda_n \psi_{n,n} \int_0^T e^{-\lambda_n t} dt \right] \\
&\left\{ \text{From definition (4.6) } \psi_{k,n} = \frac{\psi_{k,n-1}}{\lambda_n - \lambda_k} \Rightarrow \psi_{k,n-1} - \lambda_n \psi_{k,n} = -\lambda_k \psi_{k,n} \right\} \\
&= \lambda_0 \cdots \lambda_{n-1} \left[-\lambda_0 \psi_{0,n} \int_0^T e^{-\lambda_0 t} dt - \ldots - \lambda_{n-1} \psi_{n-1,n} \int_0^T e^{-\lambda_{n-1} t} dt - \lambda_n \psi_{n,n} \int_0^T e^{-\lambda_n t} dt \right] \\
&= \lambda_0 \cdots \lambda_{n-1} \left[\psi_{0,n}(e^{-\lambda_0 T} - 1) + \ldots + \psi_{n,n}(e^{-\lambda_n T} - 1) \right] \\
&= \lambda_0 \cdots \lambda_{n-1} \left[\psi_{0,n} e^{-\lambda_0 T} + \ldots + \psi_{n,n} e^{-\lambda_n T} - \sum_{i=0}^{n} \psi_{i,n} \right] \\
&\{\text{using that } \textstyle\sum_{i=0}^{n} \psi_{i,n} = 0 \text{ as showed before}\} \\
&= \lambda_0 \cdots \lambda_{n-1} \left[\psi_{0,n} e^{-\lambda_0 T} + \ldots + \psi_{n,n} e^{-\lambda_n T} \right]
\end{aligned}
$$

As a result:

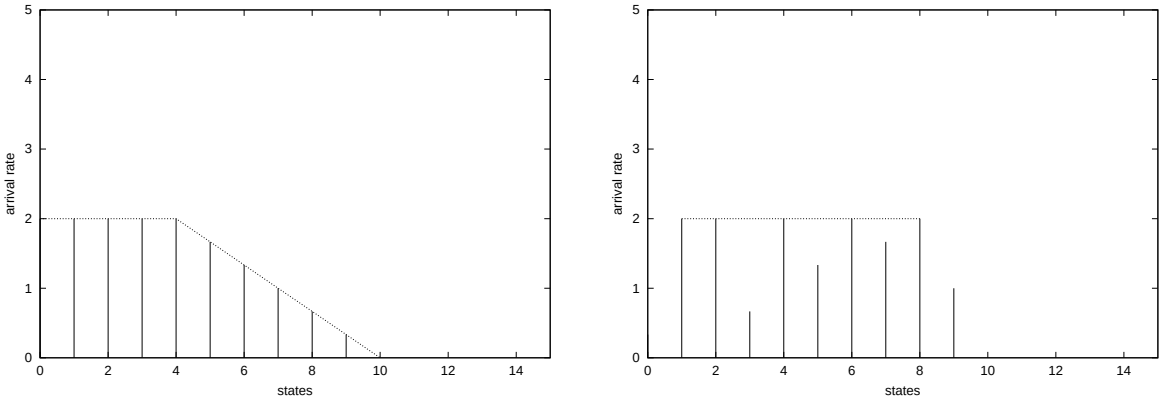
$$Prob[\text{exactly n arrivals by T}] = \lambda_0 \cdots \lambda_{n-1} \left[\psi_{0,n} e^{-\lambda_0 T} + \ldots + \psi_{n,n} e^{-\lambda_n T} \right] \qquad (4.11)$$

This is an alternative formula to Equation (4.1). We have tested the two formulae numerically and they have given the same results. In this way we have managed to check Equation (4.1) which we have used in our numerical programmes.

However, neither formula can be applied when two or more arrival rates are the same, since differences between arrival rates concerning different states, appear in the denominator. While modelling state-dependent balking this occurs often because arrival rates remain the same until a queue forms. We elaborate this issue in the next section.

4.5 State-dependent probabilities with a recurrent arrival rate

Equations (4.1) and (4.11) cannot be applied when two different states have the same arrival rate. However, this case might occur in practice, as indicated in Figure 4.2.



(a) (b)

Figure 4.2: (a) Arrival rate that is initially constant and then decreasing. (b) State dependent arrival rate, where two or more different states are allowed to have the same arrival rate.

We are first concerned with the case where the arrival rate is constant and then state-dependent as for example in Figure 4.2(a). This can occur in systems with balking, for example, when arrivals occur at a constant rate (i.e. no balking applies until a queue forms). Each time an arrival occurs it increases the system's state, thus, when this state becomes equal to or exceeds the number of servers the arrival rates will become state-dependent. Hence when calculating the probability of going from state $s - k \rightarrow s + m$, k arrivals occur at a constant rate and m at state-dependent

rates.

We want to calculate the probability that we will have exactly k arrivals during time T. This can be seen as the probability of k_1 (where k_1 is the number of arrivals needed to cause the arrival rate to first change) arrivals to occur during T_0, and $k_2 = k - k_1$ arrivals to occur during $T - T_0$. Hence the arrival rate for the first k_1 arrivals is constant and for the remaining k_2 arrivals is state-dependent. The probability density function which describes the time required to observe k_1 arrivals, when the arrival rate is constant, is given by (see for example [13]):

$$f(t) = \frac{\lambda(\lambda t)^{k_1-1}}{(k_1-1)!}e^{-\lambda t}$$

which belongs to the family of Erlang distributions. The probability density function which describes the time required to observe k_2 arrivals when the arrival rate is state-dependent with $\lambda_0, \lambda_1, \ldots, \lambda_{k_2-1}$, is given by Equation (4.5) and takes the form:

$$g(t) = \lambda_0 \cdots \lambda_{k_2-1} \sum_{i=0}^{k_2-1} \psi_{i,k_2-1} e^{-\lambda_i(t)}$$

For this reason:

$$
\begin{aligned}
P_k(t) &= Prob(exactly\ k\ arrivals\ by\ T) \\
&= \int_0^T pdf(k\ arrivals\ at\ t) pdf(0\ arrivals\ during\ T-t) dt \\
&= \int_0^T [\int_0^t pdf(k_1\ arrivals\ at\ t_0\ with\ constant\ \lambda) \times \\
&\times \quad pdf(k_2 = k - k_1\ arrivals\ state\ dependent\ at\ t - t_0) dt_0] e^{-\lambda_k (T-t)} dt \\
&= \int_0^T \int_0^t \left(\frac{\lambda (\lambda t_0)^{k_1 - 1}}{(k_1 - 1)!} e^{-\lambda t_0} \right) \left(\lambda_0 \cdots \lambda_{k_2 - 1} \sum_{i=0}^{k_2 - 1} \psi_{i,k_2-1} e^{-\lambda_i (t - t_0)} \right) e^{-\lambda_k (T-t)} dt_0 dt \\
&= \frac{\lambda^{k_1}}{(k_1 - 1)!} \lambda_0 \cdots \lambda_{k_2 - 1} \sum_{i=0}^{k_2 - 1} \psi_{i,k_2-1} e^{-\lambda_k T} \int_0^T e^{(\lambda_k - \lambda_i) t} \left(\int_0^t t_0^{k_1 - 1} e^{(\lambda_i - \lambda) t_0} dt_0 \right) dt
\end{aligned}
$$

{calculating the innermost integral using Equation (A.3) from Appendix A}

{with $k = k_1 - 1$, $T = t$, and $\alpha = \lambda_i - \lambda$ }

$$
\begin{aligned}
&= \frac{\lambda^{k_1} \lambda_0 \cdots \lambda_{k_2 - 1}}{(k_1 - 1)!} \sum_{i=0}^{k_2 - 1} \psi_{i,k_2-1} e^{-\lambda_k T} \int_0^T e^{(\lambda_k - \lambda_i) t} \left[\left(\sum_{j=0}^{k_1 - 1} \frac{(-1)}{(\lambda - \lambda_i)^{j+1}} \frac{(k_1 - 1)!}{(k_1 - 1 - j)!} t^{k_1 - 1 - j} \right) \times \right. \\
&\times \quad \left. e^{(\lambda_i - \lambda) t} + \frac{(k_1 - 1)!}{(\lambda - \lambda_i)^{k_1}} \right] dt
\end{aligned}
$$

{eliminating $(k_1 - 1)!$ which appears both as a numerator and as a denominator}

$$
\begin{aligned}
&= \lambda^{k_1} \lambda_0 \cdots \lambda_{k_2 - 1} \sum_{i=0}^{k_2 - 1} \psi_{i,k_2-1} e^{-\lambda_k T} \int_0^T e^{(\lambda_k - \lambda_i) t} \left[\left(\sum_{j=0}^{k_1 - 1} \frac{-1}{(\lambda - \lambda_i)^{j+1}} \frac{1}{(k_1 - 1 - j)!} t^{k_1 - 1 - j} \right) \times \right. \\
&\times \quad \left. e^{(\lambda_i - \lambda) t} + \frac{1}{(\lambda - \lambda_i)^{k_1}} \right] dt
\end{aligned}
$$

{interchanging summation and integration order}

$$
\begin{aligned}
&= \lambda^{k_1} \lambda_0 \cdots \lambda_{k_2 - 1} \sum_{i=0}^{k_2 - 1} \psi_{i,k_2-1} e^{-\lambda_k T} \left[\left(\sum_{j=0}^{k_1 - 1} \frac{-1}{(\lambda - \lambda_i)^{j+1}} \frac{1}{(k_1 - 1 - j)!} \int_0^T t^{k_1 - 1 - j} e^{(\lambda_k - \lambda) t} dt \right) + \right. \\
&+ \quad \left. \frac{1}{(\lambda - \lambda_i)^{k_1}} \int_0^T e^{(\lambda_k - \lambda_i) t} dt \right]
\end{aligned}
$$

$$\left\{ \begin{array}{l} \text{evaluating the integrals. For the first one we use Equation (A.3) from Appendix A,} \\ \text{with } k = k_1 - 1 - j,\ T = T, \text{ and } \alpha = \lambda_k - \lambda. \text{ The second one is straightforward.} \end{array} \right\}$$

$$\begin{aligned}
&= \lambda^{k_1}\lambda_0 \cdots \lambda_{k_2} \sum_{i=0}^{k_2} \psi_{i,k_2} e^{-\lambda_k T} \left[\left(\sum_{j=0}^{k_1-1} \frac{-1}{(\lambda-\lambda_i)^{j+1}} \frac{1}{(k_1-1-j)!} \times \right.\right. \\
&\times \left.\left(\sum_{m=0}^{k_1-1-j} \frac{-(k_1-1-j)!}{(\lambda-\lambda_k)^{m+1}(k_1-1-j-m)!} T^{k_1-1-j-m} e^{(\lambda_k-\lambda)T} + \frac{(k_1-1-j)!}{(\lambda-\lambda_k)^{k_1-j}} \right) \right) + \\
&+ \left. \frac{1}{(\lambda_k-\lambda_i)(\lambda-\lambda_i)^{k_1}} (e^{(\lambda_k-\lambda_i)T} - 1) \right] \\
&\{\text{eliminating } (k_1-1-j)! \text{ which appears both as a numerator and as a denominator}\} \\
&= \lambda^{k_1}\lambda_0 \cdots \lambda_{k_2} \sum_{i=0}^{k_2} \psi_{i,k_2} e^{-\lambda_k T} \left[\left(\sum_{j=0}^{k_1-1} \frac{1}{(\lambda-\lambda_i)^{j+1}} \times \right.\right. \\
&\times \left.\left(\sum_{m=0}^{k_1-1-j} \frac{1}{(\lambda-\lambda_k)^{m+1}(k_1-1-j-m)!} T^{k_1-1-j-m} e^{(\lambda_k-\lambda)T} - \frac{1}{(\lambda-\lambda_k)^{k_1-j}} \right) \right) + \\
&+ \left. \frac{1}{(\lambda_k-\lambda_i)(\lambda-\lambda_i)^{k_1}} (e^{(\lambda_k-\lambda_i)T} - 1) \right] \qquad (4.12)
\end{aligned}$$

Thus, Equation (4.12) can be used to calculate state-dependent arrival probabilities when a mixture of a constant and state-dependent arrival rates occur, as in Figure 4.2(a).

Let us now look at the case where some arrivals have a recurrect rate as it is illustrated in the example on Figure 4.2(b). This is less likely in the context of call centre behaviour, however it is included for completeness. This can be reduced to the previous case by appropriate rearrangement of the arrival rates. The probability of exactly k arrivals to occur, with rate of occurrence for the i arrival equal to λ_i, remains the same if we change the order in which the arrivals occur. For example, instead of calculating the probability of exactly 3 arrivals to occur during t, with corresponding arrival rates $\lambda_0, \lambda_1, \lambda_2$, we can equivalently calculate the probability of exactly 3 arrivals to occur during t, with corresponding arrival rates $\lambda_1, \lambda_2, \lambda_0$ or any other ordering. The only rule we need to apply is that we cannot change the order of the arrival rate concerning the arrival which will not occur. This last arrival rate in the previous example where the 3 arrivals occur at $\lambda_0, \lambda_1, \lambda_2$ would be λ_3 and refers to the arrival that we want to happen beyond the time interval we are interested in. By

rearrangement of the arrival rates Figure 4.2(b) becomes equivalent to Figure 4.2(a), and thus the relevant probabilities can also be calculated from Equation (4.12).

4.6 Approximations

The full DTM approach, as described in Chapter 3, involves state definition of the form

$$n : x_1, x_2, \ldots, x_m$$

and forward recurrence equations:

$$P_{t+1}[n_{t+1} : x_1^{'}, \ldots, x_m^{'}] = P_t[n_t : x_1, \ldots, x_m] \times V_r(t) \times P(\mathbf{z})$$

When balking is not present, departures and arrivals are independent events. Departures in a slot are simply a consequence of the residual service times at the start of the slot, and arrival probabilities just depend on the current arrival rate.

However when balking is present the arrival probabilities will not only depend on the number in the system at the start of the slot (as assumed in theorem **??**), but also depend on the timing of departures during the slot. Because an exact formulation of this situation would be very complex, two approximations are proposed instead. These approximations are described in the remainder of this chapter, and are evaluated in chapter 5 and 6.

The two different approximations are illustrated in Figure 4.3 and are referred to as the 'early departure' and 'late departure' approximations. In the 'early departure'

(a) 'early departure' system (b) 'late departure' system

Figure 4.3: The two approximations assume different order for the events which take place between two epochs

approximation, for the purpose of calculating arrival rates only, we assume that all departures occur at the start of an interval. This approximation intuitively seems likely to lead to an upper approximation in terms of the queue length. This is because by first removing the departures, arriving customers see the system to be emptier than it actually is, so extra arrivals are allowed to enter, thus leading to a system that is more congested than an exact model would imply. On the other hand the 'late departure' approximation, for calculating the arrival rate, assumes that all departures occur at the end of the interval. Vice versa, this system now seems likely to lead to a lower approximation, in terms of the queue length, since arrivals see a system that is more congested than the actual system.

It is worth noting here that the exact time at which departures occur does not affect any other aspect of system's performance, since we look at the system only at equally spaced intervals (epochs), and not between them.

We next give a simple example so that these concepts become clearer. Suppose we have a system with one server, and that when someone is in the system receiving service, no one else can join this system, i.e. very strong balking. In Figure 4.4 we show a possible realisation of the actual system, and the two approximations. In the exact model the first arrival occurs at time $t_1 = 0.2$, finds the system empty and so enters, has a service time of one unit, and so leaves at $t_2 = 1.2$. The second arrival occurs at $t_3 = 1.8$, finds the system empty and so enters, has a service time of one unit, and so leaves at $t_4 = 2.8$. Beside each illustration of these systems we show the sample path of the number in the system.

We can see from this example that in the 'early departure' approximation the system is as busy as the actual system at the epochs, however in the 'late departure' system it is less busy than the actual one at the second epoch. This is because the arrival at $t = 1.8$ will not be allowed to enter in the 'late departure' system, since this arrival sees one customer in the system due to the late departures assumption.

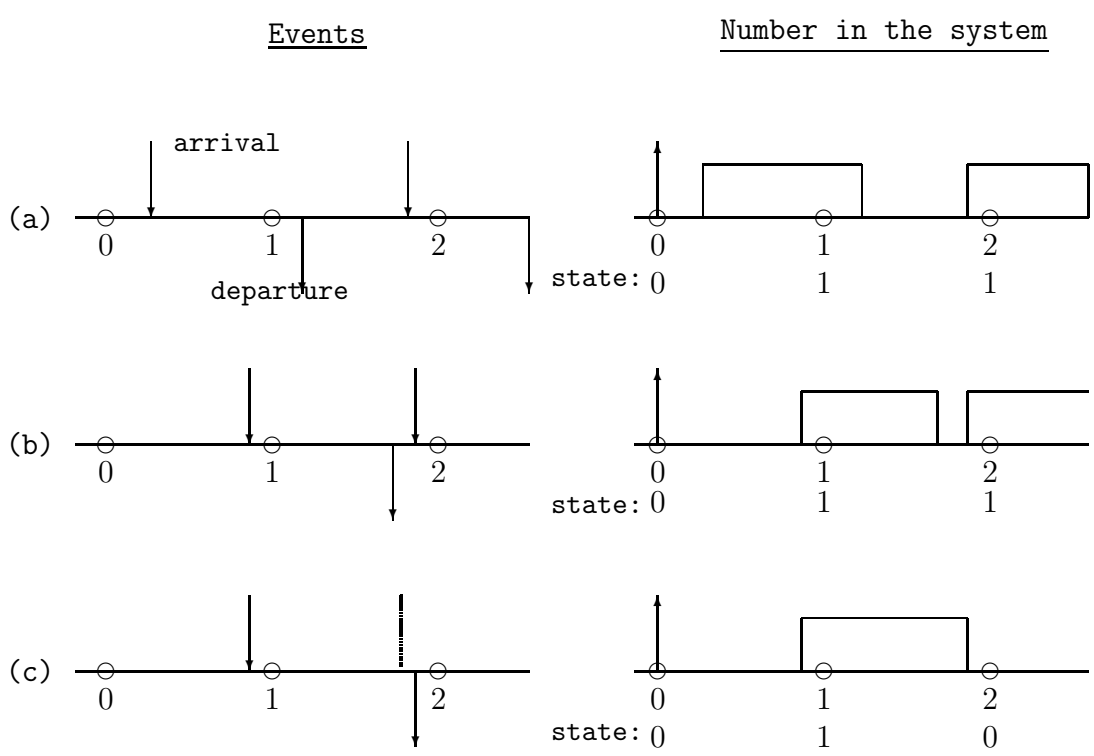


Figure 4.4: **(a)** 'exact' system, **(b)** 'early departure' system **(c)** 'late departure' system

4.7 Conclusions

This chapter describes the theoretical development of DTM to include state-dependent balking.

The first step in this process involved calculation of state-dependent Poisson probabilities. In Sections 4.2 and 4.3 a new formula (Theorem 4.1) is derived to calculate these probabilities. Having come across another suggested formula for solving the same problem, Section 4.4 provides a proof, for completeness. We have tested numerically the two formulae and, as expected, they provide the same results.

However, neither formula can be applied when a recurrent arrival rate occurs. Because our algorithms need to be able to deal with recurrent arrival rates we propose an extension of the current formula in Section 4.5. As a result this first part of the chapter provides an extended analysis of pure birth processes with non-homogeneous Poisson arrival rates beyond what is currently available in the literature.

The next step was to introduce these probabilities in DTM, where we are inter-

ested in arrivals during a slot. Due to their state-dependent nature, these arrivals are affected by the number and the time of departures during the slot. This complicates the problem analysed in Sections 4.2-4.5 extremely. A way to deal with this problem is to consider departures as one event and make assumptions about when they occur. For this reason in Section 4.6 we have introduced two approximations. An 'early departure' approximation when departures are assumed to occur just after the beginning of the slot, and a 'late departure' approximation when departures are assumed to take place just prior to the end of the slot. These are similar concepts to those of early and late arrivals that are sometimes used to model discrete-time systems, see for example [56]. However, in our case we use late and early departures and only use them to calculate the state-dependent arrival probabilities.

The approximations were introduced so that intuitively they lead to an upper and a lower approximation. Indeed in our preliminary tests of the algorithms the two approximations seem to behave as bounds of the actual solution. For this reason in the next chapter we undertake a theoretical investigation of the bounding behaviour of these two approximations.

Chapter 5

Theoretical investigation of the bounding behaviour of the two approximations

5.1 Introduction

In the previous chapter we have provided equations for calculating state-dependent entry probabilities and we have suggested two approximations in order to introduce these probabilities in the DTM algorithm. While providing an approximation for the actual solution is often useful, providing bounds is more helpful, as then the conditions that bring these bounds close together can be investigated. In this way the actual solution is estimated within a desired accuracy. Motivated by this, and by the fact that early empirical results and intuition suggest that the two approximations may behave as bounds for the actual solution, in this chapter we present the results from a theoretical investigation of this bounding behaviour.

In Section 5.2 we briefly describe how we designed a comparison between two different discrete-time scenarios. Due to the complexity of $M(t,n)/G_D/s$ systems, for the rest of this chapter we focus on $M(t,n)/D_{=1}/s$ systems. In Section 5.3 we present some basic inequalities which concern state-dependent arrival probabilities and which are going to be used later in the proofs. In Section 5.4 we show that the 'upper'

approximation estimates higher levels of congestion than the 'lower' approximation. This is the first time we implement a theoretical comparison between the two different approximations, and this extends what we have outlined in Section 5.2. All the proofs which we present in later sections follow a similar structure. In Section 5.5 we give the formulation for the exact solution. Finally, we prove in Section 5.6 that the 'upper' approximation always overestimates the actual congestion, and in Section 5.7 that the 'lower' approximation always underestimates the actual congestion.

5.2 Designing the proofs

In Section 5.4 we examine whether the upper approximation always provides a more congested system than the lower one. To date this task has proved to be impossible for the general case $M(t,n)/G_D/s$ for two main reasons. First the proof had to be designed from scratch. This is because there are no similar cases in the bibliography of discrete-time systems which are time variant and include state-dependent processes. In our provisional results the performance measure was the mean queue length at each epoch, although the probability distribution of the number in the system is also available.

Because the proof for the $M(t,n)/G_D/s$ system seemed very complicated, we have tried for simpler systems such as $M(t,n)/D_{=1}/1$ and $Geo/D_{=1}/s$. Working with these simpler systems clarified some issues. We wanted to show that applying the upper approximation to a more congested system, we end up in a more congested system than applying the lower approximation to a less congested system. So by using induction if at time t we are in the right ordering of congestion, we need to show that at time $t+1$, after applying the different approximation scenario the ordering still holds. It was obvious that since even for these systems the proof was difficult, defining this ordering based on the mean queue length was problematic. The mean queue length did not contain enough information to let us proceed for the next step. For this reason we decided to show that this ordering holds for the cumulative probability distributions of the number in the system. Suppose we have

two probability distributions $\{P_t(n)\}$, and $\{P_t^{'}(n)\}$. We assume that for each $n \geq 0$ at epoch r the following ordering for the cumulative probabilities holds:

$$\sum_{k=0}^{n} P_r(k) \geq \sum_{k=0}^{n} P_r^{'}(k)$$

We would like to show that these kind of relationships hold for the next epoch, that is:

$$\sum_{k=0}^{n} P_{r+1}(k) \geq \sum_{k=0}^{n} P_{r+1}^{'}(k)$$

When we have two cumulative probability distributions for the number in the system, and one of them is larger than, or equal to the other, for every possible value of the number in the system, then it is easy to show that an inverse ordering relates the mean queue lengths. Indeed, let us assume that we have two probability distributions $\{P(n)\}$, and $\{P^{'}(n)\}$, so that:

$$\sum_{k=0}^{n} P(k) \geq \sum_{k=0}^{n} P^{'}(k), \quad n \geq 0 \tag{5.1}$$

The mean value for $\{P(n)\}$ is:

$$\begin{aligned}
\sum_{k=0}^{n} kP(k) &= P(1) + 2P(2) + 3P(3) + \ldots \\
&= P(1) + [P(2) + P(2)] + [P(3) + P(3) + P(3)] + \ldots \\
&= [P(1) + P(2) + P(3) + \ldots] + [P(2) + P(3) + \ldots] + [P(3) + \ldots] + \ldots \\
&= [1 - P(0)] + [1 - P(0) - P(1)] + [1 - P(0) - P(1) - P(2)] + \ldots \\
&= [1 - P(0)] + \left[1 - \sum_{k=0}^{1} P(k)\right] + \left[1 - \sum_{k=0}^{2} P(k)\right] + \ldots \\
&\{ \text{ using Equation (5.1)} \} \\
&\leq [1 - P^{'}(0)] + [1 - \sum_{k=0}^{1} P^{'}(k)] + [1 - \sum_{k=0}^{2} P^{'}(k)] + \ldots \\
&= [P^{'}(1) + P^{'}(2) + P^{'}(3) + \ldots] + [P^{'}(2) + P^{'}(3) + \ldots] + [P^{'}(3) + \ldots] + \ldots \\
&= P^{'}(1) + 2P^{'}(2) + 3P^{'}(3) + \ldots = \sum_{k=0}^{n} kP^{'}(k)
\end{aligned}$$

For this reason, if we manage to prove that an ordered relationship holds for the

cumulative probability distributions, an inverse ordering will hold for the mean queue lengths.

We have managed to show that the two approximations behave as bounds of the actual solution for the system $M(t,n)/D_{=1}/s$, and we give this proof in the next section. The service time distribution of this system is deterministic so the unit slot equals the service time. This system can also be seen as an approximation of the $M(t,n)/G_D/s$ system, if we replace the service time distribution by its average.

5.3 Inequalities concerning the arrival probabilities

In this section we derive some useful inequalities for the arrival probabilities which we will use later in our proofs. Let us denote by $\mathbf{X}_k$ the sojourn time during which the system stays in state E_k, where k is the number in the system. It is also assumed in this section, unless stated differently, that the residual service time associated with each state is ∞. Let $a_k(i)$ represent the probability that starting from state E_k exactly i arrivals will occur during a pre-specified time interval. Again for $a_k(i)$, when referring to the starting state we assume that the associated residual service time is ∞, that is, no departures can occur, i.e. the corresponding arrival rates are $(\lambda_k, \lambda_{k+1}, \ldots \lambda_{k+i})$.

Lemma 5.1 *The probability of at least $n-r+1$ arrivals starting from state E_r is larger than the probability of at least $n-(r-1)+1 = n-r+2$ arrivals when starting from state E_{r-1}. That is:*

$$P(\mathbf{X}_r + \ldots + \mathbf{X}_n \leq T) > P(\mathbf{X}_{r-1} + \mathbf{X}_r + \ldots + \mathbf{X}_n \leq T)$$

Proof The above inequality is valid because the same sojourn times appear in each side of the inequality, but the probability in the right hand side contains an extra

sojourn time ($\mathbf{X}_{r-1}$). This can also be written as:

$$\sum_{i=n-r+1}^{\infty} a_r(i) > \sum_{i=n-r+2}^{\infty} a_{r-1}(i) \Leftrightarrow$$
$$a_r(n-r+1) + \sum_{i=n-r+2}^{\infty} [a_r(i) - a_{r-1}(i)] > 0 \tag{5.2}$$

where $n - r + 1 \geq 0$ or else $n \geq r - 1$. In other words the probability of going from a state r to a state i or higher is larger than the probability of going from the state $r - 1$ to state i or higher.

When we are at state $r-1$ the first arrival will occur at time $t_0 > 0$ and will move the system to state r. The probability of at least $i \geq 1$ arrivals during $t-t_0$ is smaller than the probability of at least i arrivals during t, where t is the slot duration, when the initial state is the same r. ■

Lemma 5.2 *The probability of at least n arrivals starting from state E_{k-m} is larger than the probability of at least n arrivals when starting from state E_k:*

$$P(\mathbf{X}_{k-m} + \ldots + \mathbf{X}_{k-m+n-1} \leq T) > P(\mathbf{X}_k + \ldots + \mathbf{X}_{k+n-1} \leq T)$$

Proof Each probability in the above inequality involves n sojourn times. However, since $\lambda_0 > \ldots > \lambda_{k-m} > \ldots > \lambda_k$, starting from a lower state includes higher arrival rates, and thus shorter sojourn times. This can also be written as:

$$\sum_{i=n}^{\infty} a_{k-m}(i) \geq \sum_{i=n}^{\infty} a_k(i) \quad for\ n \geq 0 \tag{5.3}$$

and since $\sum_{i=0}^{\infty} a_j(i) = 1 \Rightarrow \sum_{i=n}^{\infty} a_j(i) = 1 - \sum_{i=0}^{n-1} a_j(i)$ it takes also a dual form:

$$\sum_{i=0}^{n} a_{k-m}(i) \leq \sum_{i=0}^{n} a_k(i) \quad for\ n \geq 0 \tag{5.4}$$

■

Lemma 5.3 *Let us now define $a_k^{'}(i/t)$ as the probability that starting from state E_k with an associated residual service time t, exactly i arrivals will occur during a pre-specified time interval. If $a_k(i)$ is defined as before, then:*

$$\sum_{i=n}^{\infty} a_k^{'}(i/t) \geq \sum_{i=n}^{\infty} a_k(i) \quad for\ n \geq 0 \tag{5.5}$$

Proof When considering residual service time (i.e. $a_k^{'}(i/t)$ probabilities), departures can occur during the interval thus the corresponding arrival rates $\lambda_i^{'}$ are unknown. However, departures will cause higher arrival rates to occur. The sojourn times will be again shorter when the residual service time is t than when we assume residual service time equal to ∞. For this reason Inequality (5.5) is valid. ■

Again since $\sum_{i=0}^{\infty} a_k^{'}(i/t) = 1$ and $\sum_{i=0}^{\infty} a_k(i) = 1$ Inequality (5.5) takes the dual form:

$$\sum_{i=0}^{n} a_k^{'}(i/t) \leq \sum_{i=0}^{n} a_k(i) \quad for\ n \geq 0 \tag{5.6}$$

Lemma 5.4 *Referring again to probabilities with associated residual service times, assume that s is the maximum number of departures. Then the followin inequality is valid:*

$$\sum_{i=n}^{\infty} a_k^{'}(i/t) \leq \begin{cases} \sum_{i=n}^{\infty} a_0(i), & n \geq 0, \quad k < s \\ \\ \sum_{i=n}^{\infty} a_{k-s}(i), & n \geq 0, \quad k \geq s \end{cases} \tag{5.7}$$

Proof Starting from state $k-s,\ when\ k \geq s$ and having ∞ residual service time, or from state 0, *when* $k < s$ and having ∞ residual service time, includes higher arrival rates than starting from state k and having any residual service time (i.e. allowing for departures), since the maximum number of departures that can occur are s when $k \geq s$ and k when $k < s$. As a result for the same reasoning as for Lemma 5.3, Inequality (5.7) is valid. ■

The dual form of Inequality (5.7) is:

$$\sum_{i=0}^{n} a_k^{'}(i/t) \geq \begin{cases} \sum_{i=0}^{n} a_0(i), & n \geq 0, \quad k < s \\ \\ \sum_{i=0}^{n} a_{k-s}(i), & n \geq 0, \quad k \geq s \end{cases} \tag{5.8}$$

In this way we have derived a set of useful inequalities which we are going to use in the proofs presented in the next sections.

5.4 Comparing the upper approximation with the lower approximation for an $M(t,n)/D_{=1}/s$ system

We look at the $M(t,n)/D_{=1}/s$ system. This system has two main characteristics:

1. Arrivals occur as a time-dependent Poisson process with balking
2. The service time is deterministic and lasts one unit of time

We give the equations that relate the probabilities of having n in the system at the $r+1$ epoch with the corresponding probabilities at the r epoch. Because all services in process at time r will be completed at time $r+1$, for the lower approximation the equations take the form:

$$P_{r+1}^{L}(n) = \begin{cases} \sum_{k=0}^{s} a_k(0) P_r^L(k), & n = 0 \\ \\ \sum_{k=0}^{s} a_k(n) P_r^L(k) + \sum_{k=1}^{n} a_{s+k}(n-k) P_r^L(s+k), & n \geq 1 \end{cases} \tag{5.9}$$

and as a result:

$$\sum_{n=0}^{m} P_{r+1}^{L}(n) = \sum_{n=0}^{m}\sum_{k=0}^{s} a_k(n) P_r^L(k) + \sum_{n=1}^{m}\sum_{k=1}^{n} a_{s+k}(n-k) P_r^L(s+k),\ n \geq 1 \quad (5.10)$$

where $P_r^L(k)$ is the probability that at epoch r there are k in the system in which we apply the lower approximation, $a_k(i)$ is the probability that exactly i arrivals will occur during a time unit (slot duration) when there are k in the system with ∞ residual service time, i.e. departures will not occur. The definition of $a_k(i)$ in this section is exactly the same as in Section 5.3.

Writing them in a compact form we use the following 'matrix notation' (where n can take any value):

$$\begin{bmatrix} P_{r+1}^L(0) \\ P_{r+1}^L(1) \\ \vdots \\ P_{r+1}^L(n) \end{bmatrix} = \begin{bmatrix} a_0(0) & \dots & a_s(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_s(1) & a_{s+1}(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_s(n) & a_{s+1}(n-1) & \dots & a_{s+n}(0) \end{bmatrix} \begin{bmatrix} P_r^L(0) \\ P_r^L(1) \\ \vdots \\ P_r^L(s+n) \end{bmatrix} \quad (5.11)$$

which can be written as:

$$\mathbf{P}_{r+1}^L(n) = \mathbf{A}_L \mathbf{P}_r^L(s+n) \quad (5.12)$$

with:

$$\mathbf{A}_L = \begin{bmatrix} a_0(0) & \dots & a_s(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_s(1) & a_{s+1}(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_s(n) & a_{s+1}(n-1) & \dots & a_{s+n}(0) \end{bmatrix}, \text{ and } \mathbf{P}_r^L(k) = \begin{bmatrix} P_r^L(0) \\ P_r^L(1) \\ \vdots \\ P_r^L(k) \end{bmatrix}$$

$\mathbf{A}_L$ is introduced in order to achieve a compact representation of $P_{r+1}^L(n)$, thus it does not have necessarily square form. In this case $\mathbf{A}_L$ has dimensions $(n+1)\times(s+n+1)$. For the upper approximation we have:

$$P_{r+1}^U(n) = \begin{cases} \sum_{k=0}^{s} a_0(0) P_r^U(k), & n = 0 \\ \sum_{k=0}^{s} a_0(n) P_r^U(k) + \sum_{k=1}^{n} a_k(n-k) P_r^U(s+k), & n \geq 1 \end{cases} \tag{5.13}$$

and as a result:

$$\sum_{n=0}^{m} P_{r+1}^U(n) = \sum_{n=0}^{m} \sum_{k=0}^{s} a_0(n) P_r^U(k) + \sum_{n=1}^{m} \sum_{k=1}^{n} a_k(n-k) P_r^U(s+k), n \geq 1 \tag{5.14}$$

where $P_r^U(k)$ is the probability that at epoch r there are k in the system in which we apply the upper approximation, and $a_n(k)$defined as before.

The difference between the equations for the lower approximation (Equation (5.9)) and the equations for the upper approximation (Equation (5.13)) is that in the upper approximation arrivals see the actual number in the system reduced by the number of departures which will occur during $[r, r+1]$. Hence for the upper approximation $a_k(n)$ is replaced by $a_0(n)$ and $a_{s+k}(n-k)$ is replaced by $a_k(n-k)$. In 'matrix notation' Equation (5.13) takes the form:

$$\begin{bmatrix} P_{r+1}^U(0) \\ P_{r+1}^U(1) \\ \vdots \\ P_{r+1}^U(n) \end{bmatrix} = \begin{bmatrix} a_0(0) & \dots & a_0(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_0(1) & a_1(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_0(n) & a_1(n-1) & \dots & a_n(0) \end{bmatrix} \begin{bmatrix} P_r^U(0) \\ P_r^U(1) \\ \vdots \\ P_r^U(s+n) \end{bmatrix} \tag{5.15}$$

which can also be written as

$$\mathbf{P}_{r+1}^U(n) = \mathbf{A}_U \mathbf{P}_r^U(s+n) \tag{5.16}$$

where:

$$\mathbf{A}_U = \begin{bmatrix} a_0(0) & \dots & a_0(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_0(1) & a_1(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_0(n) & a_1(n-1) & \dots & a_n(0) \end{bmatrix}, \quad \text{and} \quad \mathbf{P}^U_{r+1}(k) = \begin{bmatrix} P^U_{r+1}(0) \\ P^U_{r+1}(1) \\ \vdots \\ P^U_{r+1}(k) \end{bmatrix}$$

Theorem 5.1 *Let us consider the two probability distributions* $\{P^L_r\}$ *and* $\{P^U_r\}$ *defined by Equation (5.9) and Equation (5.13) respectively. Suppose that for each* $n \geq 0$ *at epoch* r *the following ordering for their cumulative probability distributions holds:*

$$\sum_{k=0}^{n} P^L_r(k) \geq \sum_{k=0}^{n} P^U_r(k) \tag{5.17}$$

i.e. $\mathbf{1}\mathbf{P}^L_r(n) \geq \mathbf{1}\mathbf{P}^U_r(n)$*, where* $\mathbf{1} = [\ 1 \quad \dots \quad 1\ ]$*, i.e.* $(n+1)$*-dimensional unit vector. Then same ordering is valid at epoch* $r+1$*:*

$$\sum_{k=0}^{n} P^L_{r+1}(k) \geq \sum_{k=0}^{n} P^U_{r+1}(k) \ \textit{i.e.}\ \mathbf{1}\mathbf{P}^L_{r+1}(n) \geq \mathbf{1}\mathbf{P}^U_{r+1}(n) \tag{5.18}$$

Proof 5.1 The first stage of this proof is to relate $P^L_r(i)$ to $P^U_r(i)$.

Since we assume that Equation (5.17) is valid for any value of n, it will also be valid for $n-1$. For this reason:

$$\begin{aligned} \textit{Let} \quad \beta_n &= \sum_{k=0}^{n} P^L_r(k) - \sum_{k=0}^{n} P^U_r(k) \\ \textit{and} \quad \beta_{n-1} &= \sum_{k=0}^{n-1} P^L_r(k) - \sum_{k=0}^{n-1} P^U_r(k) \\ \textit{then} \qquad & P^U_r(n) = P^L_r(n) - \beta_n + \beta_{n-1} \end{aligned}$$

where Equation (5.17) implies that $\beta_n \geq 0$ and $\beta_{n-1} \geq 0$. Also applying Equation (5.17) for $n = 0$ gives:

$$P^L_r(0) - P^U_r(0) = \beta_0 \geq 0$$

In this way $P_r^U(i)$ is given by:

$$P_r^U(i) = \begin{cases} P_r^L(0) - \beta_0, & for\ i = 0 \\ P_r^L(i) - \beta_i + \beta_{i-1}, & for\ i > 0 \end{cases} \tag{5.19}$$

which we can write in vector form as:

$$\begin{bmatrix} P_r^U(0) \\ P_r^U(1) \\ \vdots \\ P_r^U(s+n) \end{bmatrix} = \begin{bmatrix} P_r^L(0) - \beta_0 \\ P_r^L(1) - \beta_0 + \beta_1 \\ \vdots \\ P_r^L(s+n) - \beta_{s+n} + \beta_{s+n-1} \end{bmatrix} = \begin{bmatrix} P_r^L(0) \\ P_r^L(1) \\ \vdots \\ P_r^L(s+n) \end{bmatrix} - \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{s+n} \end{bmatrix} + \begin{bmatrix} 0 \\ \beta_0 \\ \vdots \\ \beta_{s+n-1} \end{bmatrix}$$

which can be written as

$$\mathbf{P}_r^U(s+n) = \mathbf{P}_r^L(s+n) - \boldsymbol{\beta}(s+n) + \boldsymbol{\beta}(s+n-1) \tag{5.20}$$

where

$$\boldsymbol{\beta}(s+n) = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{s+n} \end{bmatrix}, \qquad \boldsymbol{\beta}(s+n-1) = \begin{bmatrix} 0 \\ \beta_0 \\ \vdots \\ \beta_{s+n-1} \end{bmatrix}$$

However, $\boldsymbol{\beta}(s+n-1)$ can be written as a function of $\boldsymbol{\beta}(s+n)$. It is:

$$\boldsymbol{\beta}(s+n-1) = \begin{bmatrix} 0 \\ \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{s+n-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & & \vdots & & \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{s+n} \end{bmatrix}$$

$$i.e.\quad \boldsymbol{\beta}(s+n-1) = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ & & \vdots & & \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \boldsymbol{\beta}(s+n) \tag{5.21}$$

We are now in a position to show that Equation (5.18) is true. Because:

$$\mathbf{1}\mathbf{P}^L_{r+1}(n) \geq \mathbf{1}\mathbf{P}^U_{r+1}(n) \Leftrightarrow$$

$$\mathbf{1}\mathbf{P}^L_{r+1}(n) - \mathbf{1}\mathbf{P}^U_{r+1}(n) \geq 0 \Leftrightarrow (\text{ from (5.12) and (5.16)})$$

$$\mathbf{1}\mathbf{A}_L\mathbf{P}^L_r(s+n) - \mathbf{1}\mathbf{A}_U\mathbf{P}^U_r(s+n) \Leftrightarrow (\text{ from (5.20)})$$

$$\mathbf{1}\mathbf{A}_L\mathbf{P}^L_r(s+n) - \mathbf{1}\mathbf{A}_U\mathbf{P}^L_r(s+n) + \mathbf{1}\mathbf{A}_U\boldsymbol{\beta}(s+n) - \mathbf{1}\mathbf{A}_U\boldsymbol{\beta}(s+n-1) \geq 0 \Leftrightarrow$$

$$\mathbf{1}(\mathbf{A}_L - \mathbf{A}_U)\mathbf{P}^L_r(s+n) + \mathbf{1}\mathbf{A}_U\boldsymbol{\beta}(s+n) - \mathbf{1}\mathbf{A}_U\boldsymbol{\beta}(s+n-1) \geq 0 \tag{5.22}$$

we need to show that Equation (5.22) is true. The first term of Equation (5.22) is:

$$\mathbf{1}(\mathbf{A}_L - \mathbf{A}_U)\mathbf{P}^L_r(s+n) =$$

$$= \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \left(\begin{bmatrix} a_0(0) & \dots & a_s(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_s(1) & a_{s+1}(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_s(n) & a_{s+1}(n-1) & \dots & a_{s+n}(0) \end{bmatrix} - \right.$$

$$\left. - \begin{bmatrix} a_0(0) & \dots & a_0(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_0(1) & a_1(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_0(n) & a_1(n-1) & \dots & a_n(0) \end{bmatrix} \right) \mathbf{P}^L_r(s+n)$$

$$= \left(\begin{bmatrix} & & & & \\ [1 & \dots & 1] & & \end{bmatrix} \begin{bmatrix} a_0(0)-a_0(0) & \dots & 0 & \dots & 0 \\ a_0(1)-a_0(1) & \dots & a_{s+1}(0)-a_1(0) & \dots & 0 \\ & & \vdots & & \\ a_0(n)-a_0(n) & \dots & a_{s+1}(n-1)-a_1(n-1) & \dots & a_{s+n}(0)-a_n(0) \end{bmatrix} \right) \mathbf{P}^L_r(s+n)$$

$$= \left[\begin{array}{ccccccc} 0 & \sum_{i=0}^{n}[a_1(i)-a_0(i)] & \dots & \sum_{i=0}^{n-1}[a_{s+1}(i)-a_1(i)] & \dots & a_{s+n}(0)-a_n(0) \end{array} \right] \mathbf{P}_r^L(s+n)$$

$$= \sum_{k=1}^{s}\sum_{i=0}^{n}[a_k(i)-a_0(i)]P_r^L(k) + \sum_{k=s+1}^{s+n}\sum_{i=0}^{n+s-k}[a_k(i)-a_{k-s}(i)]P_r^L(k)$$

$$\left\{ \text{since } \{a_k\} \text{ is a probability distribution } \sum_{i=0}^{n} a_k(i) = 1 - \sum_{i=n+1}^{\infty} a_k(i) \right\}$$

$$= \sum_{k=1}^{s}\left[1 - \sum_{i=n+1}^{\infty} a_k(i) - 1 + \sum_{i=n+1}^{\infty} a_0(i)\right] P_r^L(k) + \sum_{k=s+1}^{s+n}\left[1 - \sum_{i=n+s-k+1}^{\infty} a_k(i) - 1 + \sum_{i=n+s-k+1}^{\infty} a_{k-s}(i)\right] P_r^L(k)$$

$$= \sum_{k=1}^{s}\sum_{i=n+1}^{\infty}[a_0(i)-a_k(i)]P_r^L(k) + \sum_{k=s+1}^{s+n}\sum_{i=n+s-k+1}^{\infty}[a_{k-s}(i)-a_k(i)]P_r^L(k)$$

and if we change the index in the last term by setting $j = k - s - 1$, we have:

$$\mathbf{1}(\mathbf{A}_L - \mathbf{A}_U)\mathbf{P}_r^L(s+n) =$$

$$= \sum_{k=1}^{s}\sum_{i=n+1}^{\infty}[a_0(i)-a_k(i)]P_r^L(k) + \sum_{j=0}^{n-1}\sum_{i=n-j}^{\infty}[a_{j+1}(i)-a_{j+s+1}(i)]P_r^L(j+s+1) \qquad (5.23)$$

The summation of the second and the third terms of Equation (5.22) is:

$$\mathbf{1}\mathbf{A}_U\boldsymbol{\beta}(s+n) - \mathbf{1}\mathbf{A}_U\boldsymbol{\beta}(s+n-1) =$$

$$\{ \text{ using Equation (5.21)} \}$$

$$= \mathbf{1}\left(\mathbf{A}_U - \begin{bmatrix} a_0(0) & \dots & 0 & 0 & \dots & 0 \\ a_0(1) & \dots & a_1(0) & 0 & \dots & 0 \\ a_0(2) & \dots & a_1(1) & a_2(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_1(n-1) & a_2(n-2) & \dots & a_n(0) \end{bmatrix} \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ & & \vdots & & \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \right) \boldsymbol{\beta}(s+n)$$

$$\texttt{column:} \quad \{s+1\} \qquad \{s+2\}$$

$$= \mathbf{1}\left(\mathbf{A}_U - \begin{bmatrix} a_0(0) & a_0(0) & \dots & 0 & 0 & \dots & 0 \\ a_0(1) & a_0(1) & \dots & a_1(0) & 0 & \dots & 0 \\ & & & \vdots & & & \\ a_0(n) & a_0(n) & \dots & a_1(n-1) & a_2(n-2) & \dots & 0 \end{bmatrix}\right)\boldsymbol{\beta}(s+n) \tag{5.24}$$

$$\texttt{column:} \quad \{s\} \qquad \{s+1\} \tag{5.25}$$

(5.26)

$$= \mathbf{1}\begin{bmatrix} 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & a_0(1)-a_1(0) & 0 & \dots & 0 \\ & & & \vdots & & \\ 0 & \dots & a_0(n)-a_1(n-1) & a_1(n-1)-a_2(n-2) & \dots & a_n(0) \end{bmatrix}\boldsymbol{\beta}(s+n) \tag{5.27}$$

$$\texttt{column:} \quad \{s\} \qquad \{s+1\} \tag{5.28}$$

$$= \left(\begin{bmatrix} 1 & \dots & 1 \end{bmatrix}\begin{bmatrix} 0 & \dots & a_0(0) & 0 & \dots & 0 \\ 0 & \dots & a_0(1)-a_1(0) & 0 & \dots & 0 \\ & & & \vdots & & \\ 0 & \dots & a_0(n)-a_1(n-1) & a_1(n-1)-a_2(n-2) & \dots & a_n(0) \end{bmatrix}\right)\boldsymbol{\beta}(s+n)$$

$$\texttt{column:} \quad \{s\} \qquad \{s+1\}$$

$$= \begin{bmatrix} 0 & 0 & \dots & \sum_{i=0}^{n} a_0(i) - \sum_{i=0}^{n-1} a_1(i) & \dots & \sum_{i=0}^{n-k} a_k(i) - \sum_{i=0}^{n-k-1} a_{k+1}(i) & \dots & a_n(0) \end{bmatrix}\boldsymbol{\beta}(s+n)$$

$$\texttt{column:} \quad \{s\} \qquad \{s+k\}$$

$$= \sum_{k=s}^{s+n-1}\left[\sum_{i=0}^{n-k+s} a_{k-s}(i) - \sum_{i=0}^{n-k+s-1} a_{k-s+1}(i)\right]\beta_k + a_n(0)\beta_{s+n}$$

$$= \sum_{k=s}^{s+n-1}\left[\sum_{i=0}^{n-k+s} a_{k-s}(i) + a_{k-s+1}(n-k+s) - \sum_{i=0}^{n-k+s} a_{k-s+1}(i)\right]\beta_k + a_n(0)\beta_{s+n}$$

$$= \sum_{k=s}^{s+n-1}\left\{a_{k-s+1}(n-k+s) + \sum_{i=0}^{n-k+s}\left[a_{k-s}(i) - a_{k-s+1}(i)\right]\right\}\beta_k + a_n(0)\beta_{s+n}$$

$$= \sum_{k=s}^{s+n}\left\{a_{k-s+1}(n-k+s) + \sum_{i=0}^{n-k+s}\left[a_{k-s}(i) - a_{k-s+1}(i)\right]\right\}\beta_k$$

$$i.e.\ \ \mathbf{1}\mathbf{A}_U\boldsymbol{\beta}(s+n) - \mathbf{1}\mathbf{A}_U\boldsymbol{\beta}(s+n-1) =$$
$$= \sum_{k=s}^{s+n}\left\{a_{k-s+1}(n-k+s) + \sum_{i=n-k+s+1}^{\infty}[a_{k-s+1}(i) - a_{k-s}(i)]\right\}\beta_k \tag{5.29}$$

Substituting (5.23) and (5.29) into Equation (5.22) we finally need to show that:

$$\sum_{k=0}^{s}\sum_{i=n+1}^{\infty}[a_0(i) - a_k(i)]P_r^L(k) + \sum_{j=0}^{n-1}\sum_{i=n-j}^{\infty}[a_{j+1}(i) - a_{j+s+1}(i)]P_r^L(j+s+1) +$$
$$+ \sum_{k=s}^{s+n}\left\{a_{k-s+1}(n-k+s) + \sum_{i=n-k+s+1}^{\infty}[a_{k-s+1}(i) - a_{k-s}(i)]\right\}\beta_k \geq 0 \tag{5.30}$$

By applying Equation (5.3) for $m = k$, since $n + 1 \geq 1$, and since $P_r^L(k) \geq 0$ we conclude that the first term is non-negative. Along the same lines, by applying Equation (5.3) for $m = s$, $k = l + s + 1$, since $n - l \geq 1$, and since $P_r^L(k) \geq 0$ we conclude that the second term is non-negative. Finally, by applying Equation (5.2) for $l = k - s + 1$, since $n - l + 1 = n + s - k \geq 0$, and since $\beta_k \geq 0$ we conclude that the third term is also non-negative, thus the above inequality is valid. ■

5.5 Formulation for the exact solution in an $M(t,n)/D_{=1}/s$ system

Similar formulation to the ones done for the lower and the upper approximations can be done for the exact solution. However we now need to take into account the residual service time in order to introduce the Markov chain. Integrating and summing over all the possible combinations of k and t at time r, in each case multiplied by the appropriate transition probabilities, we have:

$$P_{r+1}^E(m) = \begin{cases} \sum_{k=0}^{s}\int a_k'(0/t)P_r^E(k,t)dt, & m = 0 \\ \sum_{k=0}^{s}\int a_k'(m/t)P_r^E(k,t)dt + \sum_{k=1}^{m}\int a_{s+k}'(m-k/t)P_r^E(s+k,t)dt, & m \geq 1 \end{cases} \tag{5.31}$$

where $P_r^E(k)$ is the probability that at epoch r there are k in the exact system, and $a_k^{'}(m/t)$ is the probability that exactly i arrivals will occur during a time unit (slot duration) when there are k in the system with t residual service time, i.e. departures will occur. Note that t is a vector of residual service times and therefore the integral with respect to t indicates integration over the full vector space. Also the definition of $a_k^{'}(m/t)$ in this section is exactly the same as in Section 5.3.

Lemma 5.5 *Let us consider the probability distribution* $\{P_{r+1}^E\}$ *defined by Equation (5.31). The following inequalities are valid for the cumulative distribution of* P_{r+1}^E*:*

$$\sum_{m=0}^{n} P_{r+1}^E(m) \begin{cases} \leq \sum_{k=0}^{s}\sum_{m=0}^{n} a_k(m)P_r^E(k) + \sum_{k=s+1}^{s+n}\sum_{m=0}^{n-k+s} a_k(m)P_r^E(k) \\ \\ \geq \sum_{k=0}^{s}\sum_{m=0}^{n} a_0(m)P_r^E(k) + \sum_{k=s+1}^{s+n}\sum_{m=0}^{n-k+s} a_{k-s}(m)P_r^E(k) \end{cases}$$

where $a_k(i)$ *as defined in Section 5.4.*

Proof From Equation (5.31) we have:

$$\begin{aligned}
\sum_{m=0}^{n} P_{r+1}^E(m) &= \sum_{k=0}^{s}\int a_k^{'}(0/t)P_r^E(k,t)dt + \sum_{m=1}^{n}\sum_{k=0}^{s}\int a_k^{'}(m/t)P_r^E(k,t)dt + \\
&+ \sum_{m=1}^{n}\sum_{k=1}^{m}\int a_{s+k}^{'}(m-k/t)P_r^E(s+k,t)dt \\
&= \sum_{m=0}^{n}\sum_{k=0}^{s}\int a_k^{'}(m/t)P_r^E(k,t)dt + \sum_{m=1}^{n}\sum_{k=1}^{m}\int a_{s+k}^{'}(m-k/t)P_r^E(s+k,t)dt
\end{aligned}$$

{ swapping the order of the summation in the second term}

$$= \sum_{m=0}^{n}\sum_{k=0}^{s}\int a_k^{'}(m/t)P_r^E(k,t)dt + \sum_{k=1}^{n}\sum_{m=k}^{n}\int a_{s+k}^{'}(m-k/t)P_r^E(s+k,t)dt$$

{ setting $s+k$ in the second summation as a new variable k }

$$= \sum_{m=0}^{n}\sum_{k=0}^{s}\int a_k^{'}(m/t)P_r^E(k,t)dt + \sum_{k=s+1}^{s+n}\sum_{m=k-s}^{n}\int a_k^{'}(m-k+s/t)P_r^E(k,t)dt$$

{ setting $m-k+s$ in the second summation as a new variable m}

$$= \sum_{m=0}^{n}\sum_{k=0}^{s}\int a_k^{'}(m/t)P_r^E(k,t)dt + \sum_{k=s+1}^{s+n}\sum_{m=0}^{n-k+s}\int a_k^{'}(m/t)P_r^E(k,t)dt$$

{ changing the order of summations and integration in both terms}

$$i.e.\ \sum_{m=0}^{n} P_{r+1}^{E}(m) = \sum_{k=0}^{s} \int \left[\sum_{m=0}^{n} a_k'(m/t) \right] P_r^E(k,t)dt + \sum_{k=s+1}^{s+n} \int \left[\sum_{m=0}^{n-k+s} a_k'(m/t) \right] P_r^E(k,t)dt \qquad (5.32)$$

Applying Inequality (5.6) in relationship (5.32) we get:

$$\begin{aligned}
\sum_{m=0}^{n} P_{r+1}^{E}(m) &\leq \sum_{k=0}^{s} \int \left[\sum_{m=0}^{n} a_k(m) \right] P_r^E(k,t)dt + \sum_{k=s+1}^{s+n} \int \sum_{m=0}^{n-k+s} a_k(m) P_r^E(k,t)dt \\
&\{ \text{ the integrations now refers only to } P_r^E(k,t) \} \\
&= \sum_{k=0}^{s} \sum_{m=0}^{n} a_k(m) \int P_r^E(k,t)dt + \sum_{k=s+1}^{s+n} \sum_{m=0}^{n-k+s} a_k(m) \int P_r^E(k,t)dt \\
&= \sum_{k=0}^{s} \sum_{m=0}^{n} a_k(m) P_r^E(k) + \sum_{k=s+1}^{s+n} \sum_{m=0}^{n-k+s} a_k(m) P_r^E(k)
\end{aligned}$$

which proves the first part of this Lemma.

Along the same lines, using Inequality (5.8) in relationship (5.32) we have:

$$\begin{aligned}
\sum_{m=0}^{n} P_{r+1}^{E}(m) &\geq \sum_{k=0}^{s} \int \left[\sum_{m=0}^{n} a_0(m) \right] P_r^E(k,t)dt + \sum_{k=s+1}^{s+n} \int \sum_{m=0}^{n-k+s} a_{k-s}(m) P_r^E(k,t)dt \\
&\{ \text{ the integration now refers only to } P_r^E(k,t) \} \\
&= \sum_{k=0}^{s} \sum_{m=0}^{n} a_0(m) \int P_r^E(k,t)dt + \sum_{k=s+1}^{s+n} \sum_{m=0}^{n-k+s} a_{k-s}(m) \int P_r^E(k,t)dt \\
&= \sum_{k=0}^{s} \sum_{m=0}^{n} a_0(m) P_r^E(k) + \sum_{k=s+1}^{s+n} \sum_{m=0}^{n-k+s} a_{k-s}(m) P_r^E(k)
\end{aligned}$$

which proves the second part of this Lemma. ■

In the next sections we apply Lemma 5.5 in order to compare the cumulative probabilities of the exact solution with those of the approximations.

5.6 Comparing the exact solution with the upper approximation in an $M(t,n)/D_{=1}/s$ system

In this section we compare the exact solution with the upper approximation.

Theorem 5.2 *Let us consider the two probability distributions* $\{P_r^U\}$ *and* $\{P_r^E\}$ *defined by Equation (5.13) and Equation (5.31) respectively. Suppose that for each* $n \geq 0$ *at epoch* r *the following ordering for their cumulative probability distributions holds:*

$$\sum_{k=0}^{n} P_r^E(k) \geq \sum_{k=0}^{n} P_r^U(k) \tag{5.33}$$

We will show that the same ordering is valid at epoch $r+1$*, that is:*

$$\sum_{k=0}^{n} P_{r+1}^E(k) \geq \sum_{k=0}^{n} P_{r+1}^U(k) \tag{5.34}$$

Proof 5.2 This theorem is similar to Theorem 5.1 if $\{P^L\}$ is replaced by $\{P^E\}$. However $\{P^L\}$ is defined by Equation (5.9) and is not similar to Equation (5.31) which defines $\{P^E\}$. As a result this proof cannot be reduced to the previous one.

Nevertheless, some relations that involved only P_r^L, P_r^U, β_k and $\mathbf{A}_U$ are still valid if we replace P_r^L by P_r^E. We give two such relationships which are going to be used later on this proof. It is noted that, having made the previous correspondence, proving them again would be a repetition. In particular if $\hat{\beta}_n = \sum_{k=0}^{n} P_r^E(k) - \sum_{k=0}^{n} P_r^U(k)$ then, as in Equation (5.20) of Theorem 5.1:

$$\mathbf{P}_r^U(s+n) = \mathbf{P}_r^E(s+n) - \hat{\boldsymbol{\beta}}(s+n) + \hat{\boldsymbol{\beta}}(s+n-1) \tag{5.35}$$

where:

$$\mathbf{P}_r^E(s+n) = \begin{bmatrix} P_r^E(0) \\ P_r^E(1) \\ \vdots \\ P_r^E(s+n) \end{bmatrix}$$

Also, as in Equation (5.29) in Theorem 5.1:

$$\mathbf{1}\mathbf{A}_U\hat{\boldsymbol{\beta}}(s+n) - \mathbf{1}\mathbf{A}_U\hat{\boldsymbol{\beta}}(s+n-1) \geq 0 \tag{5.36}$$

We will show now that Equation (5.34) holds. Starting from the left hand side of Equation (5.34) and applying Lemma 5.5:

$$\begin{aligned}
\sum_{m=0}^{n} P_{r+1}^{E}(m) &\geq \sum_{k=0}^{s}\sum_{m=0}^{n} a_0(m)P_r^E(k) + \sum_{k=s+1}^{s+n}\sum_{m=0}^{n-k+s} a_{k-s}(m)P_r^E(k) \\
&\{ \text{ setting } k-s \text{ in the second summation as a new variable k } \} \\
&= \sum_{k=0}^{s}\sum_{m=0}^{n} a_0(m)P_r^E(k) + \sum_{k=1}^{n}\sum_{m=0}^{n-k} a_k(m)P_r^E(s+k) \\
&\{ \text{ setting } m+k \text{ in the second summation as a new variable m } \} \\
&= \sum_{k=0}^{s}\sum_{m=0}^{n} a_0(m)P_r^E(k) + \sum_{k=1}^{n}\sum_{m=k}^{n} a_k(m-k)P_r^E(s+k) \\
&\{ \text{ swapping the order of the summation in the second term} \} \\
&= \sum_{k=0}^{s}\sum_{m=0}^{n} a_0(m)P_r^E(k) + \sum_{m=1}^{n}\sum_{k=1}^{m} a_k(m-k)P_r^E(s+k)
\end{aligned}$$

$\{$ from the definition of $\mathbf{1}, \mathbf{A}_U, \mathbf{P}_r^E(s+n),$ we can write the above summation in matrix form$\}$

$$\left\{ \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} a_0(0) & \dots & a_0(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_0(1) & a_1(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_0(n) & a_1(n-1) & \dots & a_n(0) \end{bmatrix} \begin{bmatrix} P_r^E(0) \\ P_r^E(1) \\ \vdots \\ P_r^E(s+n) \end{bmatrix} \right\}$$

$$\begin{aligned}
&= \mathbf{1}\mathbf{A}_U\mathbf{P}_r^E(s+n) \qquad \{ \text{ using (5.35) } \} \\
&= \mathbf{1}\mathbf{A}_U[\mathbf{P}_r^U(s+n) + \hat{\boldsymbol{\beta}}(s+n) - \hat{\boldsymbol{\beta}}(s+n-1)] \\
&= \mathbf{1}\mathbf{A}_U\mathbf{P}_r^U(s+n) + \mathbf{1}\mathbf{A}_U[\hat{\boldsymbol{\beta}}(s+n) - \hat{\boldsymbol{\beta}}(s+n-1)]
\end{aligned}$$

$\{$ as the second term is positive from (5.36) $\}$

$$\geq \mathbf{1}\mathbf{A}_U\mathbf{P}_r^U(s+n)$$

{ from the definition of $\mathbf{1}, \mathbf{A}_U, \mathbf{P}_r^U(s+n)$, we can write this in a summation form}

$$\left\{ \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} a_0(0) & \dots & a_0(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_0(1) & a_1(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_0(n) & a_1(n-1) & \dots & a_n(0) \end{bmatrix} \begin{bmatrix} P_r^U(0) \\ P_r^U(1) \\ \vdots \\ P_r^U(s+n) \end{bmatrix} \right\}$$

$$= \sum_{k=0}^{s} \sum_{m=0}^{n} a_0(m) P_r^U(k) + \sum_{m=1}^{n} \sum_{k=1}^{m} a_k(m-k) P_r^U(s+k)$$

{ from Equation (5.14) }

$$= \sum_{m=0}^{n} P_{r+1}^U(m)$$

■

In this way we have showed the desired ordering between the cumulative probabilities of the exact solution and the upper approximation. As a result the upper approximation, i.e. the 'early departure' approximation, provides an upper bound of the actual congestion for $M(t,n)/D_{=1}/s$ systems. In the next section we compare the exact solution with the lower approximation.

5.7 Comparing the exact solution with the lower approximation in an $M(t,n)/D_{=1}/s$ system

In this section we show that for an $M(t,n)/D_{=1}/s$ system the lower approximation provides a lower bound of the actual congestion. For this proof we are going to use the following lemma.

Lemma 5.6 *If $\mathbf{A}_L$ is defined as in Section 5.4 and $\bar{\mathbf{b}}(s+n-1)$ is a column vector with positive elements defined analogous to $\bar{\mathbf{b}}(s+n-1)$ in Section 5.4 then:*

$$\mathbf{1}\mathbf{A}_L[\bar{\boldsymbol{\beta}}(s+n) - \bar{\boldsymbol{\beta}}(s+n-1)] \geq \sum_{k=0}^{s-1} \sum_{m=0}^{n} [a_k(m) - a_{k+1}(m)]\, \bar{\boldsymbol{\beta}}_k$$

Proof

$$\mathbf{1}\mathbf{A}_L\bar{\boldsymbol{\beta}}(s+n)-\mathbf{1}\mathbf{A}_L\bar{\boldsymbol{\beta}}(s+n-1)=$$

$$\left\{\text{However, }\bar{\boldsymbol{\beta}}(s+n-1)=\begin{bmatrix}0\\ \bar{\beta}_0\\ \bar{\beta}_1\\ \vdots\\ \bar{\beta}_{s+n-1}\end{bmatrix}=\begin{bmatrix}0&0&\dots&0&0\\ 1&0&\dots&0&0\\ 0&1&\dots&0&0\\ &&\vdots&&\\ 0&0&\dots&1&0\end{bmatrix}\begin{bmatrix}\bar{\beta}_0\\ \bar{\beta}_1\\ \bar{\beta}_2\\ \vdots\\ \bar{\beta}_{s+n}\end{bmatrix}=\begin{bmatrix}0&0&\dots&0&0\\ 1&0&\dots&0&0\\ 0&1&\dots&0&0\\ &&\vdots&&\\ 0&0&\dots&1&0\end{bmatrix}\bar{\boldsymbol{\beta}}(n+s)\right\}$$

$$=\mathbf{1}\mathbf{A}_L\bar{\boldsymbol{\beta}}(s+n)-\mathbf{1}\left(\begin{bmatrix}a_0(0)&\dots&0&0&\dots&0\\ a_0(1)&\dots&a_{s+1}(0)&0&\dots&0\\ a_0(2)&\dots&a_{s+1}(1)&a_{s+2}(0)&\dots&0\\ &&&\vdots&&\\ a_0(n)&\dots a_{s+1}(n-1)&&a_{s+2}(n-2)&\dots a_{s+n}(0)&\end{bmatrix}\begin{bmatrix}0&0&\dots&0&0\\ 1&0&\dots&0&0\\ 0&1&\dots&0&0\\ &&\vdots&&\\ 0&0&\dots&1&0\end{bmatrix}\right)\bar{\boldsymbol{\beta}}(n+s)$$

$$\{s+1\}\qquad\{s+2\}$$

$$=\mathbf{1}\mathbf{A}_L\bar{\boldsymbol{\beta}}(s+n)-\mathbf{1}\begin{bmatrix}a_1(0)&a_2(0)&\dots&0&0&\dots&0\\ a_1(1)&a_2(1)&\dots&a_{s+1}(0)&0&\dots&0\\ &&&\vdots&&&\\ a_1(n)&a_2(n)&\dots&a_{s+1}(n-1)&a_{s+2}(n-2)&\dots&0\end{bmatrix}\bar{\boldsymbol{\beta}}(s+n)$$

$$\{s\}\qquad\{s+1\}$$

$$=\mathbf{1}\begin{bmatrix}a_0(0)-a_1(0)&\dots&a_s(0)&0&\dots&0\\ a_0(1)-a_1(1)&\dots&a_s(1)-a_{s+1}(0)&0&\dots&0\\ &&&\vdots&&\\ a_0(n)-a_1(n)&\dots&a_s(n)-a_{s+1}(n-1)&a_{s+1}(n-1)-a_{s+2}(n-2)&\dots&a_{s+n}(0)\end{bmatrix}\bar{\boldsymbol{\beta}}(s+n)$$

$$\{s\}\qquad\{s+1\}$$

$$= \left(\mathbf{1} \begin{bmatrix} a_0(0)-a_1(0) & \dots & a_s(0) & 0 & \dots & 0 \\ a_0(1)-a_1(1) & \dots & a_s(1)-a_{s+1}(0) & 0 & \dots & 0 \\ & & & \vdots & & \\ a_0(n)-a_1(n) & \dots & a_s(n)-a_{s+1}(n-1) & a_{s+1}(n-1)-a_{s+2}(n-2) & \dots & a_{s+n}(0) \end{bmatrix} \right) \bar{\boldsymbol{\beta}}(s+n)$$

$$\{s\} \qquad\qquad \{s+1\}$$

$$= \left[\sum_{i=0}^{n} [a_0(i)-a_1(i)] \dots \sum_{i=0}^{n} a_s(i) - \sum_{i=0}^{n-1} a_{s+1}(i) \dots \sum_{i=0}^{n-k} a_{s+k}(i) - \sum_{i=0}^{n-k-1} a_{s+k+1}(i) \dots a_{s+n}(0) \right] \bar{\boldsymbol{\beta}}(s+n)$$

$$\{s\} \qquad\qquad \{s+k\}$$

$$= \sum_{k=0}^{s-1} \sum_{i=0}^{n} [a_k(i) - a_{k+1}(i)] \bar{\beta}_k + \sum_{k=s}^{s+n-1} \left[\sum_{i=0}^{n-k+s} a_k(i) - \sum_{i=0}^{n-k+s-1} a_{k+1}(i) \right] \bar{\beta}_k + a_{s+n}(0) \bar{\beta}_{s+n}$$

$$= \sum_{k=0}^{s-1} \sum_{i=n+1}^{\infty} [a_{k+1}(i) - a_k(i)] \bar{\beta}_k + \sum_{k=s}^{s+n-1} \left[\sum_{i=0}^{n-k+s} a_k(i) + a_{k+1}(n-k+s) - \sum_{i=0}^{n-k+s} a_{k+1}(i) \right] \bar{\beta}_k + a_{s+n}(0) \bar{\beta}_{s+n}$$

$$= \sum_{k=0}^{s-1} \sum_{i=n+1}^{\infty} [a_{k+1}(i) - a_k(i)] \bar{\beta}_k + \sum_{k=s}^{s+n-1} \left\{ a_{k+1}(n-k+s) + \sum_{i=0}^{n-k+s} [a_k(i) - a_{k+1}(i)] \right\} \bar{\beta}_k + a^e_{s+n}(0) \bar{\beta}_{s+n}$$

$$= \sum_{k=0}^{s-1} \sum_{i=n+1}^{\infty} [a_{k+1}(i) - a_k(i)] \bar{\beta}_k + \sum_{k=s}^{s+n} \left\{ a_{k+1}(n-k+s) + \sum_{i=0}^{n-k+s} [a_k(i) - a_{k+1}(i)] \right\} \bar{\beta}_k$$

$$= \sum_{k=0}^{s-1} \sum_{i=n+1}^{\infty} [a_{k+1}(i) - a_k(i)] \bar{\beta}_k + \sum_{k=s}^{s+n} \left\{ a_{k+1}(n-k+s) + \sum_{i=n-k+s+1}^{\infty} [a_{k+1}(i) - a_k(i)] \right\} \bar{\beta}_k$$

{the second summation is positive since it is summation of positive terms. Indeed $\bar{\beta}_k \geq 0$}

{is multiplied by the term in the brackets. This term is positive }

{ by applying Inequality 5.2 for $r = k+1$ and n replaced by $n+s$}

$$\geq \sum_{k=0}^{s-1} \sum_{i=n+1}^{\infty} [a_{k+1}(i) - a_k(i)] \bar{\beta}_k = \sum_{k=0}^{s-1} \left\{ \left[\sum_{i=n+1}^{\infty} a_{k+1}(i) \right] - \left[\sum_{i=n+1}^{\infty} a_k(i) \right] \right\} \bar{\beta}_k$$

$$= \sum_{k=0}^{s-1} \left\{ 1 - \left[\sum_{i=0}^{n} a_{k+1}(i) \right] - 1 + \left[\sum_{0}^{n} a_k(i) \right] \right\} \bar{\beta}_k = \sum_{k=0}^{s-1} \sum_{i=0}^{n} [a_k(i) - a_{k+1}(i)] \bar{\beta}_k$$

■

Theorem 5.3 *Let us consider the two probability distributions* $\{P_r^E\}$ *and* $\{P_r^L\}$ *defined by Equation (5.31) and Equation (5.9) respectively. Suppose that for each* $n \geq 0$ *at epoch* r *the following ordering for their cumulative probability distributions holds:*

$$\sum_{k=0}^{n} P_r^L(k) \geq \sum_{k=0}^{n} P_r^E(k) \tag{5.37}$$

We will show that the same ordering is valid at epoch $r+1$*, that is:*

$$\sum_{k=0}^{n} P_{r+1}^L(k) \geq \sum_{k=0}^{n} P_{r+1}^E(k) \tag{5.38}$$

Proof 5.3 What we want to prove here is again similar to Theorem 5.1 if $\{P^U\}$ is replaced by $\{P^E\}$. However, $\{P^U\}$ has a different formulation than $\{P^E\}$ and as a result a new proof is provided.

Nevertheless, some relations that involved only P_r^L, P_r^U, β_k and $\mathbf{A}_U$ are still valid if we replace P_r^U by P_r^E. We give two such relationships which are going to be used later on this proof. It is noted that, having made the previous correspondence, proving them again would be a repetition. In particular if $\bar{\beta}_n = \sum_{k=0}^{n} P_r^L(k) - \sum_{k=0}^{n} P_r^E(k)$ then, as in Equation (5.20) of Theorem 5.1:

$$\mathbf{P}_r^E(s+n) = \mathbf{P}_r^L(s+n) - \bar{\boldsymbol{\beta}}(s+n) + \bar{\boldsymbol{\beta}}(s+n-1) \tag{5.39}$$

$$i.e. \quad P_r^E(k) = \begin{cases} P_r^L(0) - \bar{\beta}_0, & k = 0 \\ P_r^L(k) - \bar{\beta}_k + \bar{\beta}_{k-1}, & k \geq 1 \end{cases} \tag{5.40}$$

Starting from the left hand side of Equation (5.38) and according to Lemma 5.5 we

have:

$$\sum_{m=0}^{n} P_{r+1}^{E}(m) \leq \sum_{k=0}^{s}\sum_{m=0}^{n} a_k(m)P_r^E(k) + \sum_{k=s+1}^{s+n}\sum_{m=0}^{n-k+s} a_k(m)P_r^E(k)$$

{ applying Inequality (5.4) to the first term for $m = k - s$ }

$$\leq \sum_{k=0}^{s}\left[\sum_{m=0}^{n} a_s(m)\right] P_r^E(k) + \sum_{k=s+1}^{s+n}\sum_{m=0}^{n-k+s} a_k(m)P_r^E(k)$$

{ adding and subtracting the term $a_k(m)$ in the first term and setting in the second term $k - s$ as k}

$$= \sum_{k=0}^{s}\sum_{m=0}^{n} \left[a_s(m) + a_k(m) - a_k(m)\right] P_r^E(k) + \sum_{k=1}^{n}\sum_{m=0}^{n-k} a_{s+k}(m)P_r^E(s+k)$$

{ splitting the first term and setting $m + k$ in the second summation as a new variable m }

$$= \sum_{k=0}^{s}\sum_{m=0}^{n} a_k(m)P_r^E(k) + \sum_{k=0}^{s}\sum_{m=0}^{n} \left[a_s(m) - a_k(m)\right] P_r^E(k) + \sum_{k=1}^{n}\sum_{m=k}^{n} a_{s+k}(m-k)P_r^E(s+k)$$

{swapping the order of the summation in the third term}

$$= \sum_{k=0}^{s}\sum_{m=0}^{n} a_k(m)P_r^E(k) + \sum_{k=0}^{s}\sum_{m=0}^{n} \left[a_s(m) - a_k(m)\right] P_r^E(k) + \sum_{m=1}^{n}\sum_{k=1}^{m} a_{s+k}(m-k)P_r^E(s+k)$$

{combining the first and the third terms and using the definition of $\mathbf{A}_L, \mathbf{P}_r^E(s+n)$}

$$\left\{ \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} a_0(0) & \dots & a_s(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_s(1) & a_{s+1}(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_s(n) & a_{s+1}(n-1) & \dots & a_{s+n}(0) \end{bmatrix} \begin{bmatrix} P_r^E(0) \\ P_r^E(1) \\ \vdots \\ P_r^E(s+n) \end{bmatrix} \right\}$$

$$i.e.\ \sum_{m=0}^{n} P_{r+1}^{E}(m) \leq \mathbf{1}\mathbf{A}_L\mathbf{P}_r^E(s+n) + \sum_{k=0}^{s}\sum_{m=0}^{n} \left[a_s(m) - a_k(m)\right] P_r^E(k) \qquad (5.41)$$

The second term in the (5.41) is:

$$\sum_{k=0}^{s}\sum_{m=0}^{n} \left[a_s(m) - a_k(m)\right] P_r^E(k) = \{ \text{ using Equation (5.40}\}$$

$$= \sum_{k=1}^{s}\sum_{m=0}^{n} \left[a_s(m) - a_k(m)\right] \left[P_r^L(k) - \bar{\beta}_k + \bar{\beta}_{k-1}\right] + \sum_{m=0}^{n} \left[a_s(m) - a_0(m)\right] \left[P_r^L(0) - \bar{\beta}_0\right]$$

$$= \sum_{k=0}^{s}\sum_{m=0}^{n} \left[a_s(m) - a_k(m)\right] P_r^L(k) - \sum_{k=0}^{s}\sum_{m=0}^{n} \left[a_s(m) - a_k(m)\right] \bar{\beta}_k + \sum_{k=1}^{s}\sum_{m=0}^{n} \left[a_s(m) - a_k(m)\right] \bar{\beta}_{k-1}$$

{ setting $k-1$ in the last term as a new variable k }

$$= \sum_{k=0}^{s}\sum_{m=0}^{n}[a_s(m)-a_k(m)]\,P_r^L(k) - \sum_{k=0}^{s}\sum_{m=0}^{n}[a_s(m)-a_k(m)]\,\bar{\beta}_k + \sum_{k=0}^{s-1}\sum_{m=0}^{n}[a_s(m)-a_{k+1}(m)]\,\bar{\beta}_k$$

{ noting that for $k=s$ the second term is zero }

$$= \sum_{k=0}^{s}\sum_{m=0}^{n}[a_s(m)-a_k(m)]\,P_r^L(k) - \sum_{k=0}^{s-1}\sum_{m=0}^{n}[a_s(m)-a_k(m)]\,\bar{\beta}_k + \sum_{k=0}^{s-1}\sum_{m=0}^{n}[a_s(m)-a_{k+1}(m)]\,\bar{\beta}_k$$

$$= \sum_{k=0}^{s}\sum_{m=0}^{n}[a_s(m)-a_k(m)]\,P_r^L(k) + \sum_{k=0}^{s-1}\sum_{m=0}^{n}[a_k(m)-a_{k+1}(m)]\,\bar{\beta}_k$$

{ as Inequality (5.4) for $m=k-s$, implies that the first term is positive }

$$\leq \sum_{k=0}^{s-1}\sum_{m=0}^{n}[a_k(m)-a_{k+1}(m)]\,\bar{\beta}_k$$

Substituting this result into (5.41):

$$\sum_{m=0}^{n}P_{r+1}^E(m) \leq \mathbf{1}\mathbf{A}_L\mathbf{P}_r^E(s+n) + \sum_{k=0}^{s-1}\sum_{m=0}^{n}[a_k(m)-a_{k+1}(m)]\,\bar{\beta}_k$$

{ using Equation (5.39}

$$= \mathbf{1}\mathbf{A}_L[\mathbf{P}_r^L(s+n) - \bar{\boldsymbol{\beta}}(s+n) + \bar{\boldsymbol{\beta}}(s+n-1)] + \sum_{k=0}^{s-1}\sum_{m=0}^{n}[a_k(m)-a_{k+1}(m)]\,\bar{\beta}_k$$

$$= \mathbf{1}\mathbf{A}_L\mathbf{P}_r^L(s+n) - \left[\mathbf{1}\mathbf{A}_L[\bar{\boldsymbol{\beta}}(s+n) - \bar{\boldsymbol{\beta}}(s+n-1)] - \sum_{k=0}^{s-1}\sum_{m=0}^{n}[a_k(m)-a_{k+1}(m)]\,\bar{\beta}_k\right]$$

{ as the second term is always negative according to Lemma 5.6 }

$$\leq \mathbf{1}\mathbf{A}_L\mathbf{P}_r^L(s+n)$$

{ from the definition of $\mathbf{1}, \mathbf{A}_L, \mathbf{P}_r^L(s+n)$, we can write the above summation in matrix notation}

$$\left\{ \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} a_0(0) & \dots & a_s(0) & 0 & \dots & 0 \\ a_0(1) & \dots & a_s(1) & a_{s+1}(0) & \dots & 0 \\ & & & \vdots & & \\ a_0(n) & \dots & a_s(n) & a_{s+1}(n-1) & \dots & a_{s+n}(0) \end{bmatrix} \begin{bmatrix} P_r^L(0) \\ P_r^L(1) \\ \vdots \\ P^L r(s+n) \end{bmatrix} \right\}$$

$$= \sum_{k=0}^{s}\sum_{m=0}^{n}a_k(m)P_r^L(k) + \sum_{m=1}^{n}\sum_{k=1}^{m}a_{s+k}(m-k)P_r^L(s+k)$$

{from Equation (5.13)}

$$= \sum_{m=0}^{n}P_{r+1}^L(m)$$

■

We have showed that the lower approximation, i.e. the 'late departure' approximation, provides a lower bound of the actual congestion for $M(t,n)/D_{=1}/s$ systems. As a result for this category of systems our approximations bound the actual solution. More empirical results are provided in the next chapter where we develop a simulation model the output of which we compare with the approximations.

5.8 Summary

In this chapter we undertook a theoretical investigation of the bounding behaviour of the two approximations via a series of lemmas and proofs. Due to the complexity of the problem we have limited our investigation to an $M(t,n)/D/s$ system, i.e. a system with a deterministic service time distribution.

The formulation of the two approximations is given in Section 5.4 and we have shown (Theorem 5.1) that an $M(t,n)/D/s$ discrete-time system with 'early departures' (upper approximation) has at each epoch a smaller cumulative distribution of the number in the system compared to a discrete-time system with 'late departures' (lower approximation).

We have shown in Section 5.6 (Theorem 5.2) that an $M(t,n)/D/s$ discrete-time system with 'early departures' has at each epoch a smaller cumulative distribution of the number in the system than the actual system, and as a result is always more congested than the actual one. In Section 5.7 (Theorem 5.3) that an $M(t,n)/D/s$ discrete-time system with 'late departures' has at each epoch a larger cumulative distribution of the number in the system than the actual system, and as a result is always less congested than the actual one. Thus for an $M(t,n)/D/s$ system we have proved that the two approximations behave as bounds.

Note that it was shown early in the chapter that ordering of cumulative distributions implied inverse ordering of distribution means. Hence the more powerful results above also imply results for mean numbers in the system.

In the next chapter we develop a simulation model in order to provide a broader investigation of this bounding behaviour.

Chapter 6

Empirical investigation of the bounding behaviour of the two approximations vs a simulation model

6.1 Introduction

In the previous chapter we introduced two approximations in order to include balking in the DTM approach, and showed theoretically that for some limited cases they behave as bounds. To achieve a broader investigation for a wider set of cases, and due to the lack of analytical models, a simulation model was developed and employed.

This chapter describes this simulation model, validates it by comparison with known analytical models (e.g. negative exponential inter-arrival and service times), and proceeds to use it to evaluate the two approximations. In particular this is done in order to investigate first whether the two approximations behave as bounds, and second the nature of the 'bounds'.

6.2 Simulation characteristics

In order to implement this simulation model, we chose not to use a simulation package, or a simulation language. Instead the model was developed in a programming language (C++). The reason for this is that the performance measures of interest (e.g. mean queue length) can be calculated by a procedural programme, and that nowadays every programming language has in its library a function for generating random numbers that is the 'heart' of simulation. Also, a computational model provides more flexibility than a simulation package for using and altering functional parameters of the systems which we are trying to model. Another advantage of the simulation model versus a simulation package is the full control and transparency of the random number generator. On the other hand, these systems do not possess sophisticated characteristics that would require us to use a simulation language.

The parameters of the simulation model are the number of servers and the pre-specified probability distributions for the arrivals, services and balking. Representative observations for these random variables are produced using a random number generator, a method described in the literature (for example see [14]) as Monte Carlo simulation.

One run of a Monte Carlo simulation represents a single sample path of the input process of the queueing system under consideration. As a result, we get a single sample path of the output, which can be viewed as one of many possible realisations of the system's performance. Hence the behaviour of the system can only be properly described in terms of the probability distribution of the full set of realisations. At any point of time this distribution will have mean, standard deviation, percentage points etc, all of which can be estimated from the simulated sample.

The number of runs needed for the simulation model to give accurate results, is unknown and usually case driven. This is a disadvantage of simulation methods since a lot of trials are needed to specify the appropriate number of runs to achieve the desired accuracy. Some other disadvantages of simulation methods were mentioned in Section 2.5. However, it is well known that the larger the number of different runs

the larger the accuracy of the results. For this reason our simulation results stem from a large number of runs, and for this model the values used were of order 10^5 or higher.

This simulation is mainly event oriented although in parallel a time oriented process takes place. This is because updating the system's structural parameters (i.e. time-dependent arrival rate and number of servers) and recording the system's state takes place at the equally spaced epochs (for comparison with the DTM model). Each time an event occurs (arrival or departure) the system is updated, however the times of interest are the epochs and it is then that we record the system's state. We will see how this is achieved in the next section, where we describe the simulation model.

6.3 Simulation model

In this section we give a description of the implementation of the simulation model. The simulation structure is fairly simple and is described briefly. When using simulation to produce accurate results, the random number generator is crucial, and is described next. Finally the method of incorporating balking into the model is outlined.

6.3.1 Simulation structure

The simulation is event-based and starts with an empty system at time $t = 0$. The time counter is moved to the time that the next event occurs, unless an epoch is met during the inter-event time, in which case the system's state needs to be recorded and the system's parameters (i.e. arrival rate and number of servers) are updated. By moving the time counter we mean that the inter-event elapsed time is subtracted from each of the residual service times and the inter-arrival time. To decide on whether the next event is an arrival or a departure we compare the inter-arrival time with the minimum residual time.

If the next event is a departure, one of the updated residual service times will be zero, indicating a free server. When a queue exists, one call from the queue is moved

to this server by reducing the queue size by one and by calculating a new residual time by sampling from the service time distribution. The time of the next event is then calculated by comparing the updated inter-arrival time with the updated and the new residual service times.

If the next event is an arrival we use a balking function to decide whether or not it joins the system. In the case that it joins the system, the queue size is increased by one unless there is a null residual service time in which case a new residual service time is calculated by sampling from the service time distribution. If more than one servers are free the first null residual service time is used as for this simulation it is not important how we allocate jobs to servers. Whether or not the arrival balks the next step is to calculate a new inter-arrival time. The time of the next event is now calculated by comparing the new inter-arrival time with the updated residual service times.

From the above we conclude that a careful consideration of all possible events and actions which occur in systems with balking was enough to structure this simulation. We have found more challenging dealing with the generation of random numbers which is described next. Also, in Section 6.3.3 we describe in more detail how we have implemented the state-dependent balking procedure. Finally, for the interested reader the programme is available in Appendix B.

6.3.2 Random number generator

As mentioned in section 6.2 the crux of every simulation model is the random number generator. C++ has a function named rand() which returns a random integer number between 0 and $RAND\_MAX - 1$. The rand() function generates a sequence of integers $I_1, I_2, I_3, \ldots$, by using the recurrence relation

$$I_{j+1} = aI_j + b \ \ (mod\ m)$$

where $m = RAND\_MAX$, and is called the modulus, and a, b are positive integers. The period of this recurrence relationship cannot be greater than m. A problem that

arises is that $RAND\_MAX$ is often not very large, and $ANSI\ C$ standard requires only that it be at least 32767. In our case it was actually 32767. This means that there are at most 32768 different values to use, so running the simulation more than 32768 times will produce repetitions of previous runs. For this reason rand() has raised a lot of criticisms, and is no longer considered a good random number generator.

Park and Miller [64] propose a 'Minimal Standard' random number generator, that has accumulated a large amount of successful use. The generator is not claimed to be perfect, however it has a period of $2^{31} - 2 = 2.1 \times 10^9$ which is sufficient for our runs. This random generator was implemented as a function and was included in all our simulation programmes.

Another subtle point that concerns the random number generator is that one has to be careful on using 'seeds'. C++ programmers are encouraged to use as a seed the $time(NULL)$ function, that returns an integer, which is the number of seconds that elapsed since a specific date in the past. Following this advice, each time a simulation run was executed the random number was invoked with this seed. However, the execution time of each run was much shorter than a second, hence consecutive runs would often be exactly the same. For example if the runtime of 10^5 simulation runs was 10^3 seconds, the number of different runs would be 10^3. This problem was identified while analysing early outputs, and was overcome by using the 'Park and Miller' random number generator. This changes the seed in a different way, has been extensively tested, and did not seem to produce any surprising effects in our investigations.

6.3.3 Simulation and balking

There are two possible ways to achieve sampling from a non-homogeneous Poisson processes. Either sample from different distributions each time the arrival rate changes (either because of its time-dependent nature, or because an event occurs) or sample from a fixed distribution that corresponds to the maximum arrival rate, and only allow some of the arrivals (based on the balking rate) to enter the system. This

latter method is mentioned in the literature as thinning and is strongly suggested compared to the first method which can give misleading results when a low arrival rate precedes a high arrival rate [41]. We agree that this statement is valid for event oriented simulations, however this issue does not arise in time oriented simulations.

For this implementation a mixture of these methods were used. This is because the inhomogeneity in the post-balking arrival rate stems from two sources, i.e. time-dependence in the pre-balking arrival rate (which is time oriented) and state-dependence in the post-balking arrival rate (which is simulated event oriented). In the DTM algorithm the time dependent pre-balking arrival rate is assumed constant during the time between two epochs. The same assumption is made for the simulation model. Hence at epochs where there is a change in the pre-balking arrival rate function we need to sample from a different distribution. In between two successive epochs, each time an arrival or departure occurs the post-balking arrival rate changes (due to an increase/decrease in balking probability). Because this procedure is event based, the thinning method had to be used. The maximum pre-balking arrival rate during any slot is the arrival rate at the beginning of the slot as calculated from the arrival rate function. Note that we have chosen to use a mixture of these methods instead of an overall thinning method as the latter method is computationally slower.

The equivalence between the two sampling methods stems from the Poisson process definition. When post-balking arrivals occur at random at a rate λ_n, where n depends on the number of customers in the system, the probability that someone will enter the system during $(t, t+\delta t)$ equals the probability that someone arrives during this time interval with rate λ_n, and from the Poisson definition this equals $\lambda_n \delta t$.

Alternatively, if we assume that the arrivals occur at a maximum possible rate λ_{max}, and that the probability of a successful entry is λ_n/λ_{max} (it is noted here that since λ_{max} is the pre-balking arrival rate, the quantity λ_n/λ_{max} takes values in $[0, 1]$, and therefore can represent probability),then the probability of having an entry during $(t, t+\delta t)$ is given by the product of the probability of having an arrival during this

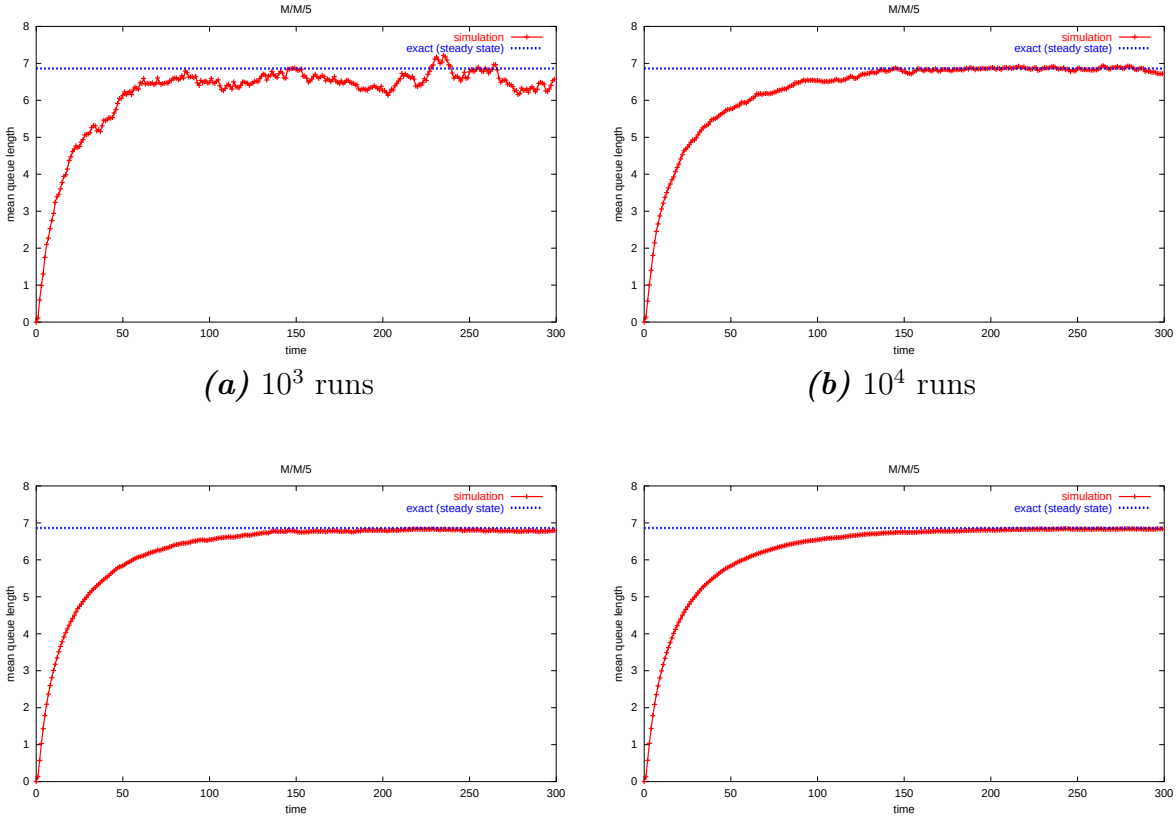
interval and the probability of having a successful entry. In other words:

$$\begin{aligned} Prob[entry\ in\ (t, t+\delta t)] &= Prob[arrival\ in\ (t, t+\delta t)] \times Prob[successful\ entry] \\ &= \lambda_{max}\delta t \times \frac{\lambda_n}{\lambda_{max}} = \lambda_n \delta t \end{aligned}$$

as required.

6.4 Validation of the simulation model

In this section we validate the simulation model. This is done by comparing systems where exact results are available with the simulation output. Figure 6.1 shows through an example how (as would be expected) the average queue lengths become less variable as the number of simulation runs increases. It is obvious that the results remain almost unchanged when the number of runs is 10^5 or higher.



(a) 10^3 runs ***(b)*** 10^4 runs

(c) 10^5 runs ***(d)*** 10^6 runs

Figure 6.1: Simulation results of different number of runs for an $M/M/5$ system.

6.4.1 M/D/s and M/M/s steady state

We start the validation of the simulation model with the $M/D/s$ steady-state results for which results are tabulated, and the $M/M/s$ steady state for which an analytical result exists. Figure 6.2 shows the simulation results for M/D/s and M/M/s systems compared with the exact results for these systems. In each case $10^6/2$ simulation runs were performed.

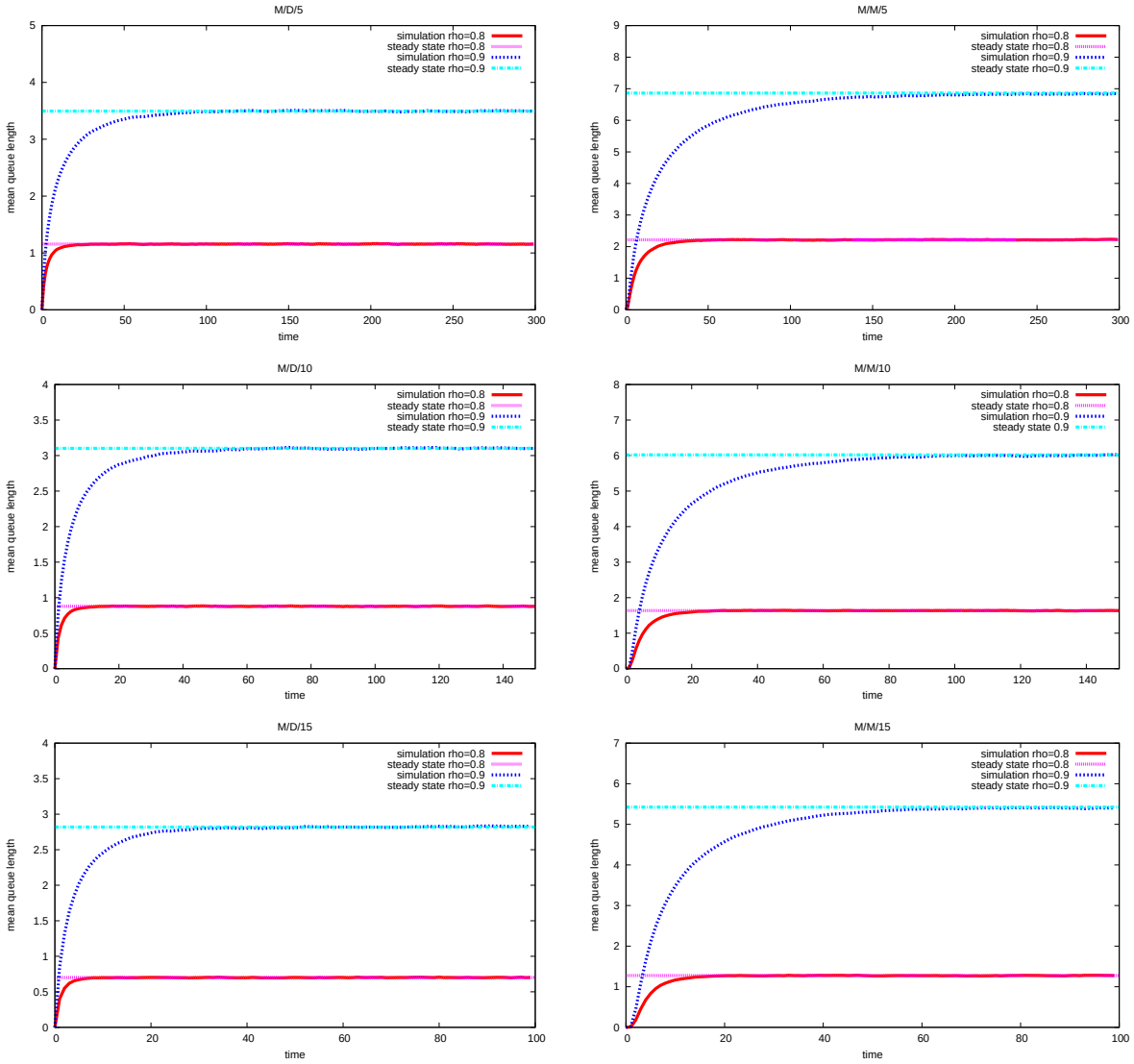


Figure 6.2: The mean queue length behaviour for constant and markovian service time distribution, starting from empty.

Figure 6.2 implies that the simulation model provides a very good estimation of the mean queue length, for these systems. To see how accurate this estimation is we provide a statistical analysis of the output. We select one of these systems, for

example the $M/D/5$ with $\rho = 0.8$, and we look at the confidence intervals of the mean queue length for the epochs after the steady state has been achieved (since there we know the exact mean queue length, which for this case is 1.1562).

Let us assume that we run the simulation n times. At each epoch i, each time we run the simulation a different possible sample queue length is observed, leading to a distribution of the queue length for this point of time. The programme (in Appendix B) is designed to keep a track of these queue lengths (by using a cumulative vector) in order to give as output the average queue length $(\sum_{i=1}^{n} L_i)/n$, it was easily modified to cumulate the L_i^2, so the quantity $(\sum_{i=1}^{n} L_i^2)/n$ is also calculated. The sample standard deviation can now be calculated from the usual formula:

$$\begin{aligned} s_i &= \sqrt{\frac{\sum_{i=1}^{n}(L_i - (\sum_{i=1}^{n} L_i/n))^2}{n-1}} = \sqrt{\frac{1}{n-1}\left[(\sum_{i=1}^{n} L_i^2) - \frac{(\sum_{i=1}^{n} L_i)^2}{n}\right]} \\ &= \sqrt{\frac{n}{n-1}\left[\left(\frac{\sum_{i=1}^{n} L_i^2}{n}\right) - \left(\frac{\sum_{i=1}^{n} L_i}{n}\right)^2\right]} \end{aligned}$$

Due to the central limit theorem, which is valid due to the large number of simulations (here $n = 10^6/2$), we know that the average queue length will follow a normal distribution with mean equal to the actual mean and standard deviation estimated by the sample standard deviation divided by the square root of the sample size. Based on this, 95% confidence intervals at the epochs $i = 100, \ldots, 300$ were calculated and our interest was on whether the actual mean queue length (1.1562) was located in these intervals for 95% of the cases. This was the case, as only in 12 out of 200 cases the estimated value was outside the CI. Some example cases can be seen in Table 6.1. The narrowness of the CI indicates the high accuracy of the simulation model.

6.4.2 $M(n)/M/s$ at steady state

We now test the simulation model against cases where the arrival rate depends on the number in the system. More specifically we look at the machine interference

Epoch	Estimated Mean Queue Length	Confidence Interval (CI)	* indicates values $\notin$ CI
215	1.1542	(1.148482655 , 1.159917345)	
216	1.14985	(1.144144700 , 1.155555300)	*
217	1.15145	(1.145746398 , 1.157153602)	
218	1.1544	(1.148684020 , 1.160115980)	
219	1.15586	(1.150142799 , 1.161577201)	
220	1.15607	(1.150351136 , 1.161788864)	
221	1.15553	(1.149811554 , 1.161248446)	
222	1.1554	(1.149668087 , 1.161131913)	
223	1.15538	(1.149656339 , 1.161103661)	
224	1.15481	(1.149082651 , 1.160537349)	
225	1.15649	(1.150762887 , 1.162217113)	
226	1.15396	(1.148241207 , 1.159678793)	
227	1.15272	(1.146997782 , 1.158442218)	
228	1.15645	(1.150721967 , 1.162178033)	
229	1.15528	(1.149550951 , 1.161009049)	
230	1.15605	(1.150319328 , 1.161780672)	

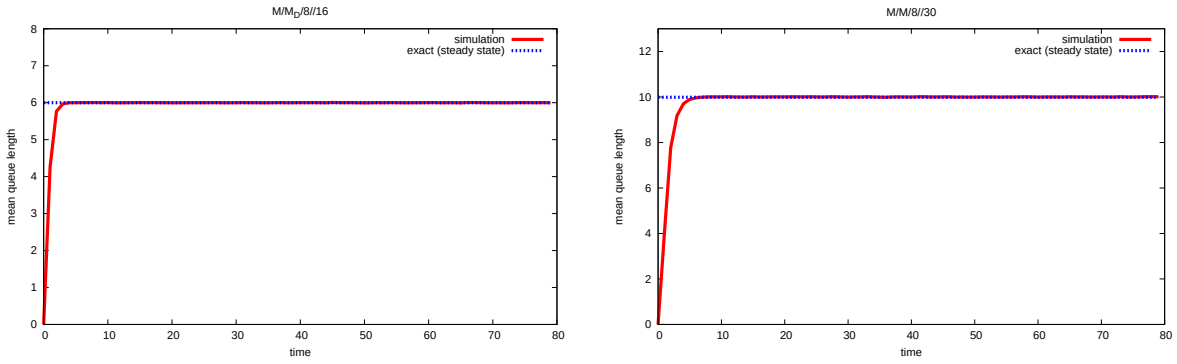
Table 6.1: Estimated mean queue lengths and confidence intervals

problem. For this problem exact steady-state results exist when the service times follow a negative exponential distribution. These results can be looked up either in tables (see for example [65]), or can be obtained by solving the finite number of balance equations for the steady state probabilities.

Figure 6.3 presents two examples of this comparison. In the first example we have $N = 16$ machines, $s = 8$ servers, mean time to breakdown 0.5 units, and mean repair time = 2 units. For this reason when there are n machines in the system the arrival rate is $2(16 - n)$ breakdowns per unit of time, and can be rewritten as $32(1 - n/16)$, thus the balking factor is introduced with the term $(1 - n/16)$. For population $N = 16$, service factor $X = \frac{1/\mu}{1/\mu+1/\lambda} = 0.8$, and 8 servers, the tables give an efficiency factor $F = 0.625$. Using this result the average number of units waiting for service is $N \times (1 - F) = 6$.

In the second example $N = 30$, $s = 10$, and both mean time to breakdown and repair time is 1 unit. The arrival rate now is $(30 - n) = 30(1 - n/30)$ breakdowns per unit of time, which implies that the balking function is $(1 - n/30)$. From the tables for $N = 30$, service factor $X = \frac{1/\mu}{1/\mu+1/\lambda} = 0.5$, and 10 servers, we get the efficiency factor $F = 0.667$. Thus, the average number in the queue is $N \times (1 - F) = 9.99$.

The simulation model gave very accurate estimations for the steady state mean queue length, when the arrival rate is state dependent. Figure 6.3 illustrates this accuracy for the two examples described above. We can also note that in these systems the time to reach the steady state is short.



(a) *(b)*

Figure 6.3: Mean queue length behaviour for two machine interference systems. **(a)** arrival rate $2(16-n)$ per unit and service rate 0.5 per unit; **(b)** arrival rate $(30-n)$ per unit and service rate 1 per unit

6.5 Bounding behaviour of the approximations

We have now developed a simulation model which as we have shown estimates accurately the actual solution. In the rest of this chapter we compare the two approximations with the actual solution, as it is estimated by the simulation model, in order to see whether they provide bounds, and how the bounds behave. This task was undertaken in chapter 4 for some limited cases in which theoretical comparison was feasible. However, there is need to examine a wider set of cases in order to be confident about the bounding behaviour of the two approximations. We also investigate the factors which affect the size of the gap between the two bounds. Our interest is to see how these factors can be altered in order to bring these bounds closer together. This will help us to establish rules on how to control the accuracy of our approximations.

For this investigation we start from an arbitrary system with parameters that could occur in practice. We then change one of these parameters keeping the others

constant, and observe the effect on the behaviour of the bounds. We use an $M(n)/D/8$ system that has $s = 8$ servers, deterministic service time distribution equal to one unit of time, arrival rate = 20 customers per unit of time, and state-dependent balking when $n \geq s$ introduced by the term 0.9^{1+q} (where n is the number in the system, and q is the number in the queue).

As we have seen in chapter 4 the difference between the two approximations is that at each point of time arriving customers in the 'upper' approximation assume early departures (i.e. see a quieter system than the actual one), while arriving customers in the 'lower' approximation assume late departures (i.e. see a more busy system than the actual one). For this reason the average number of departures during a step is expected to be a determinant of the gap between the approximations.

Let us assume busy systems with average service time equal to one unit of time. In comparison to the 'lower' approximation, arrivals in the 'upper' approximation will find an additional number of empty places to occupy, either in the queue or in service, equal to the number of departures during this step. When these systems are busy the number of departures per step is:

$$\left[\frac{departures}{step}\right] = s \times \mu \times step \tag{6.1}$$

where s is the number of servers, μ is the service rate (departures per server per unit of time), and *step* is the number of units of time between two epochs (moments of time at which the system state is updated).

We note that because μ refers to departures per unit of time, any change in μ can also be seen as a change in the unit of time and hence the step size. For example doubling the service rate is the same as keeping the same service rate but doubling the unit of time and hence the step size. As a result we only need to study systems with different step sizes and there is no need to study systems with different service rates. can be used in order to control the accuracy of the approximations.

We next present results for a range of numbers of servers, step sizes and variances of service time. A summary of the characteristics of the systems modelled and the

figures containing the results is given in Table 6.2 for quick reference. In the following sections we comment on the effect of each parameter, and then use the results to recommend how formula (6.1) can be used in order to control the accuracy of the approximations.

6.5.1 Bounding behaviour of the approximations

First and foremost in all the systems studied the two approximations behave as bounds to the actual solution. This can be seen in Figures 6.4-6.13, which present results for different number of servers, different step sizes, and different variances of the service time distribution.

For this reason for the remainder of this chapter we are able to focus on how the system's parameters affect the size of the gap between the two approximations and as a result an upper limit on the errors involved in using the approximations.

6.5.2 Changing the number of servers

Figure 6.4 shows the effect of changing the number of servers. We notice from the graphs in this figure that the gaps between the approximations change considerably when the number of servers is changed. This is expected since, according to Equation (6.1) the difference between the two approximations for busy systems should scale with the number of servers. Since $\mu = 1$ and $step = 1$ in all cases we expect the gap between the two approximations to be proportional to the number of servers for busy systems. Indeed, in Figure 6.4(a) 4 servers were used and the gap equals 4, in Figure 6.4(b) 5 servers were used and the gap equals 5, and in Figure 6.4(c) 6 servers were used and the gap equals 6. However, when the system becomes less busy the differences do not continue to increase in proportion to the number of servers. For example in Figure 6.4(d) 8 servers were used and the gap is just less than 8, while in the last two graphs (Figure 6.4(e), Figure 6.4(f)) the systems are much

Figure number	step size	number of servers (s)	maximum rho (ρ)	Balking coefficient	variance of service time
Figure 6.4(a)	1	4	4	0.9	0
Figure 6.4(b)	1	5	3.2	0.9	0
Figure 6.4(c)	1	6	2.66	0.9	0
Figure 6.4(d)	1	8	2	0.9	0
Figure 6.4(e)	1	10	1.6	0.9	0
Figure 6.4(f)	1	12	1.33	0.9	0
Figure 6.5(a)	1	8	2	0.9	0
Figure 6.5(b)	0.5	8	2	0.9	0
Figure 6.5(c)	0.25	8	2	0.9	0
Figure 6.5(d)	0.125	8	2	0.9	0
Figure 6.6(a)	1	8	1	0.9	0
Figure 6.6(b)	0.5	8	1	0.9	0
Figure 6.6(c)	0.25	8	1	0.9	0
Figure 6.6(d)	0.125	8	1	0.9	0
Figure 6.7(a)	1	8	0.75	0.9	0
Figure 6.7(b)	0.5	8	0.75	0.9	0
Figure 6.7(c)	0.25	8	0.75	0.9	0
Figure 6.7(d)	0.125	8	0.75	0.9	0
Figure 6.8	0.5	8	2	0.9	0
Figure 6.8	0.5	8	2	0.9	0.2
Figure 6.8	0.5	8	2	0.9	0.4
Figure 6.8	0.5	8	2	0.9	0.6
Figure 6.9	0.5	8	2	0.9	0.6
Figure 6.9	0.5	8	2	0.9	0.8
Figure 6.9	0.5	8	2	0.9	1
Figure 6.9	0.5	8	2	0.9	2
Figure 6.10	0.5	8	1	0.9	0
Figure 6.10	0.5	8	1	0.9	0.2
Figure 6.10	0.5	8	1	0.9	0.4
Figure 6.10	0.5	8	1	0.9	0.6
Figure 6.11	0.5	8	1	0.9	0.6
Figure 6.11	0.5	8	1	0.9	0.8
Figure 6.11	0.5	8	1	0.9	1
Figure 6.11	0.5	8	1	0.9	2
Figure 6.12	0.5	8	0.75	0.9	0
Figure 6.12	0.5	8	0.75	0.9	0.2
Figure 6.12	0.5	8	0.75	0.9	0.4
Figure 6.12	0.5	8	0.75	0.9	0.6
Figure 6.13	0.5	8	0.75	0.9	0.6
Figure 6.13	0.5	8	0.75	0.9	0.8
Figure 6.13	0.5	8	0.75	0.9	1
Figure 6.13	0.5	8	0.75	0.9	2

Table 6.2: Summary of the queueing system characteristics associated with results in Figures 6.4-6.13.

less busy, since we have 10 and 12 servers, and the gap is much smaller than the one predicted by the above formula.

6.5.3 Changing the step size

We now change the step size, i.e. the duration of time between two sequential epochs. In general systems with different step sizes have different epochs. However, if one step can be written as a rational multiple of the other we will have some time moments appearing as epochs to both systems. If we assume that the rational number expressing the ratio between the larger and the smaller step is p/q, the frequency with which the epochs of the system with the larger step appear in the system with the smaller step is $1/p$. For this reason if we select steps that are integer multiples of the smallest step size (so $q = 1$), we will have all the epochs of the systems with the larger steps appearing as epochs in the systems with smaller steps.

In this set of experiments, see Figures 6.5-6.7, the step sizes are changed while keeping the other parameters constant. According to Equation (6.1) we expect the gap between the approximations to scale with the step size. Since $s = 8$ and $\mu = 1$ we expect this gap, for busy systems, to be proportional to 8 times the step size.

This pattern can be clearly seen in Figures 6.5-6.7. In all these figures graphs (b) are produced by using half the step used for graphs (a). We observe that the distance between the approximations reduces by half. Along the same lines, graphs (c) use $step = 0.25$ so compared with graphs (b) which use $step = 0.5$ we can again see the distance between the two approximations reduces by a factor 0.5. Finally, graphs (d) use $step = 0.125$ and as a result have half of the gap that appears in graphs (c) which use $step = 0.25$. We therefore conclude that by controlling the step size we can control the size of the gap between the approximations. This is an important finding, since by controlling the step size we can actually control the accuracy of the approximations.

6.5.4 The effect of the variance of the service time distribution

In this section we are interested in the effect that the variance of the service time has on the approximations. We allow the variance to take a range of values $(0, 0.2, 0.4, 0.6, 0.8, 1, 2)$. These values correspond to a coefficient of variation between $[0, \sqrt{2}]$, which covers a wide range of distributions. This set of experiments is done for 'busy' systems (Figures 6.8-6.9), 'quiet' systems (Figures 6.10-6.11), and 'very quiet' systems (Figures 6.12-6.13). We note that for the 'quiet' and 'very quiet' systems we present the number in the system instead of the number in the queue.

It is clear for all these cases that changing the variance of the service time distribution has no discernible impact on the difference between the two approximations, i.e. the accuracy is unaffected by the variance of the service time distribution.

In fact we also note that the variance of the service time distribution also seems to have very little impact on the actual level of congestion in these systems. This is most probably caused by the balking procedure, since for systems without balking increasing the variance would tend to increase the number in the system, and the number in the queue. However, we will not attempt to explain at this point why this might happen, as we undertake this task in chapter 6.

6.6 Conclusions

Our major findings from this chapter are the following:

- The two approximations bound the actual mean queue length.
- For busy systems the difference between the two approximations is proportional to the number of departures per step, which for these systems is given by the formula: $\left[\frac{departures}{step}\right] = s \times \mu \times step$, where s is the number of servers, μ is the service rate (departures per server per unit of time), and $step$ is the number of units of time between two epochs.

- For this reason by controlling the step size we can actually control the difference between the two bounds, and as a consequence the accuracy of the approximations.

Thus, in terms of practical use of the approximate models, we suggest use of the models can start with an arbitrary step size. If 'k' times better accuracy is needed (i.e. the gap between the approximations needs to be reduced 'k' times) we recommend the user to re-run the models with new step size 'k' times smaller than the initial one.

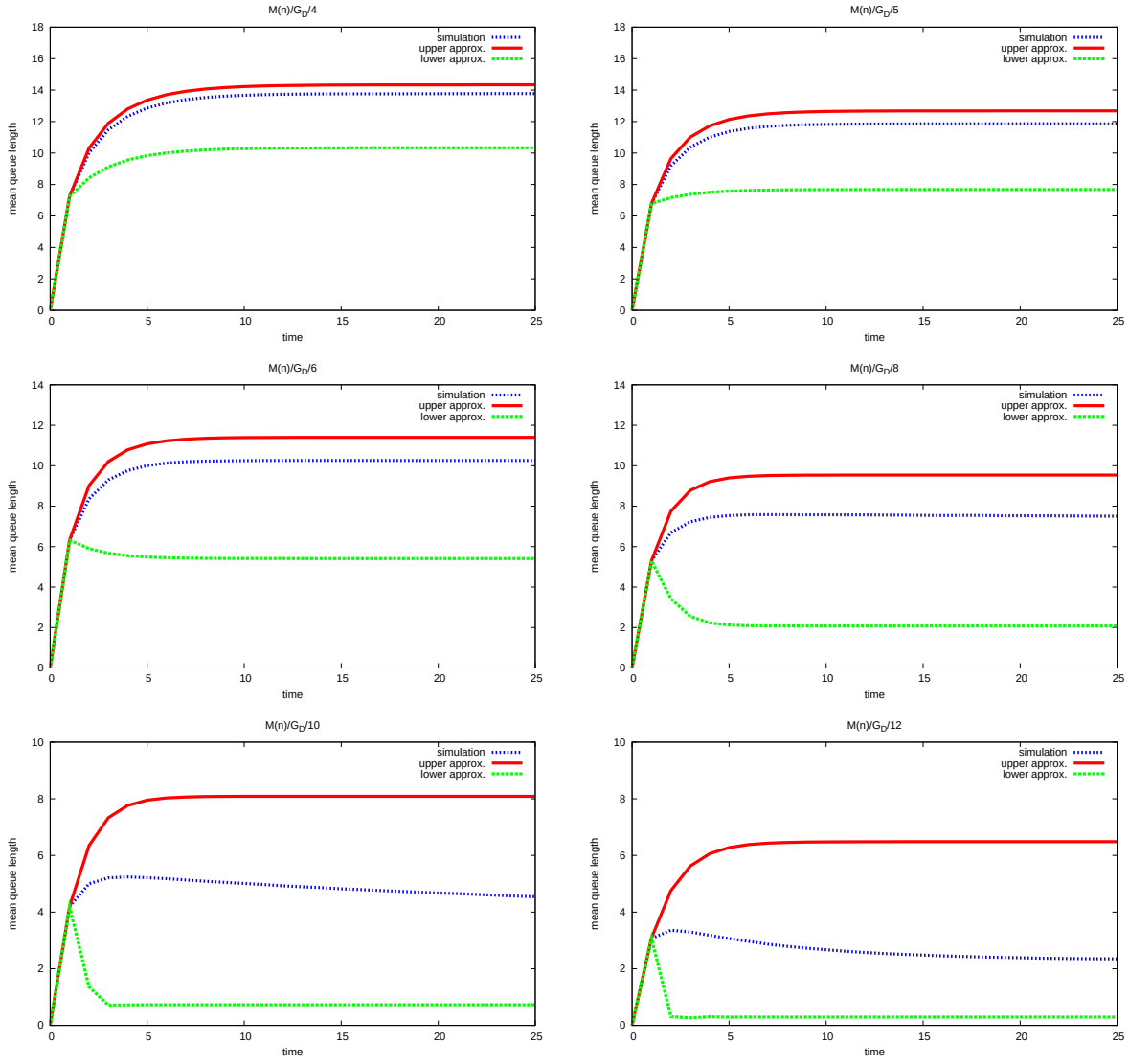
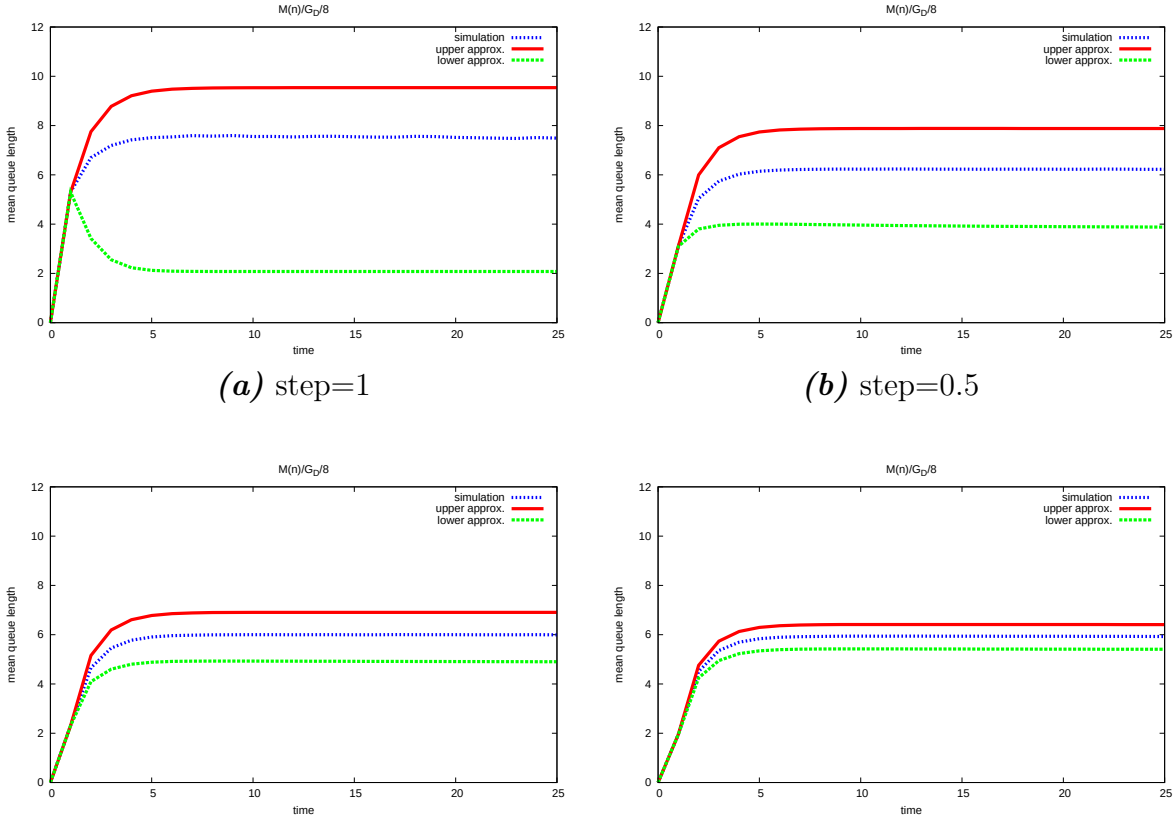


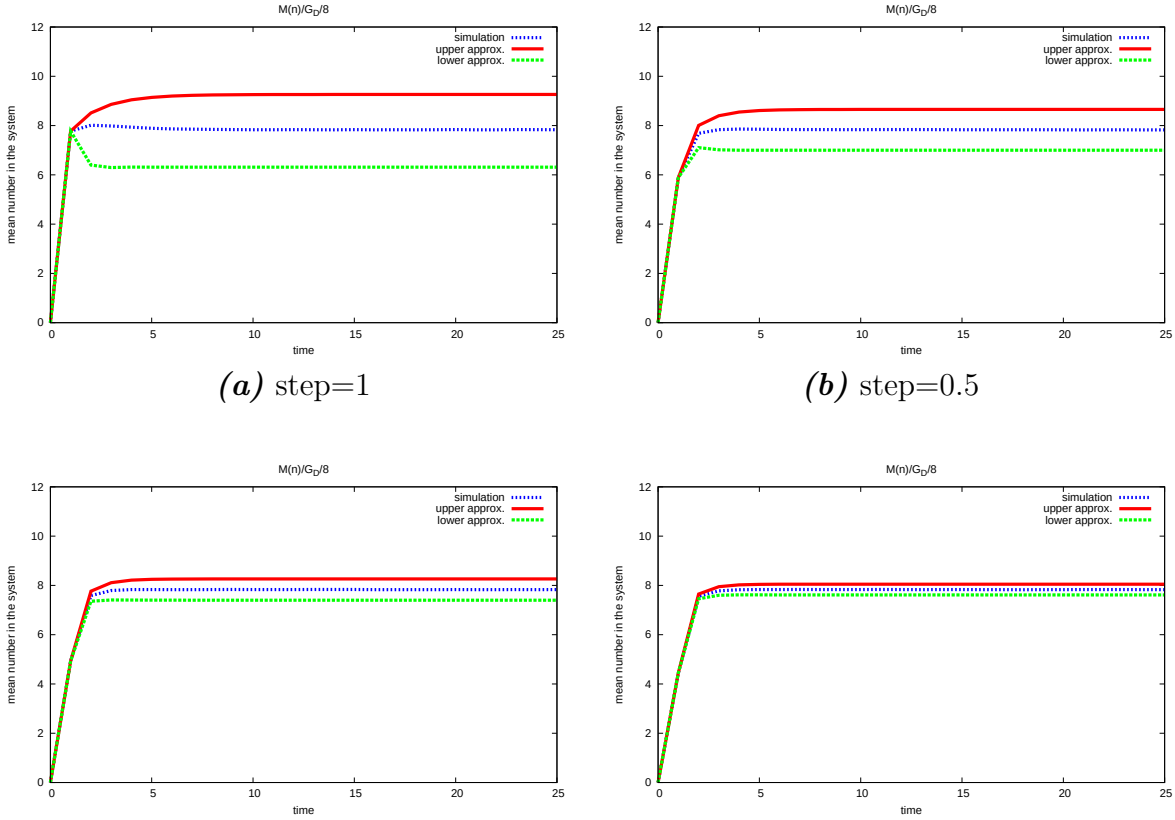
Figure 6.4: Mean queue length using approximations and simulation model for $M(\lambda_t(n)/D/8$ with $\lambda_n(t) = 2(0.9)^{1+n}$



(a) step=1 ***(b)*** step=0.5

(c) step=0.25 ***(d)*** step=0.125

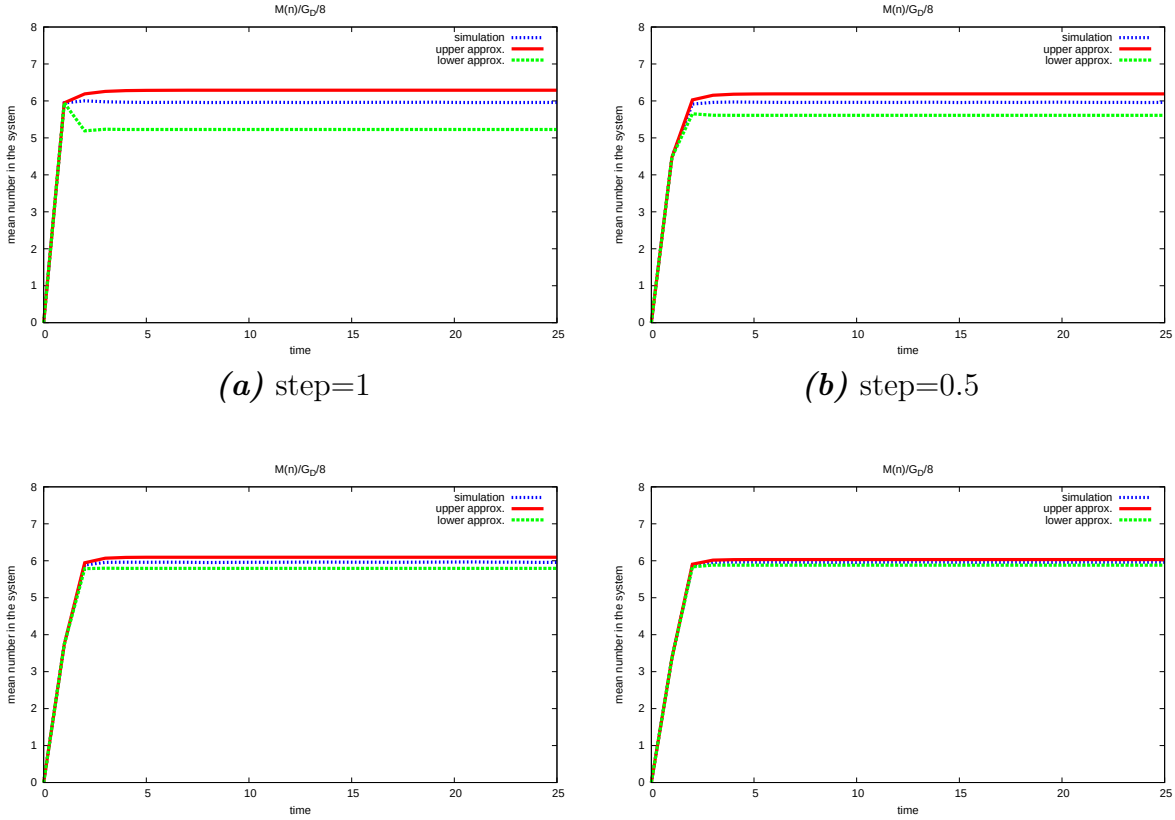
Figure 6.5: System with s=8 servers, maximum rho=2 (when no balking), and geometrical balking function 0.9^{1+n}, where n is the number in the queue, for different step sizes.



(a) step=1 ***(b)*** step=0.5

(c) step=0.25 ***(d)*** step=0.125

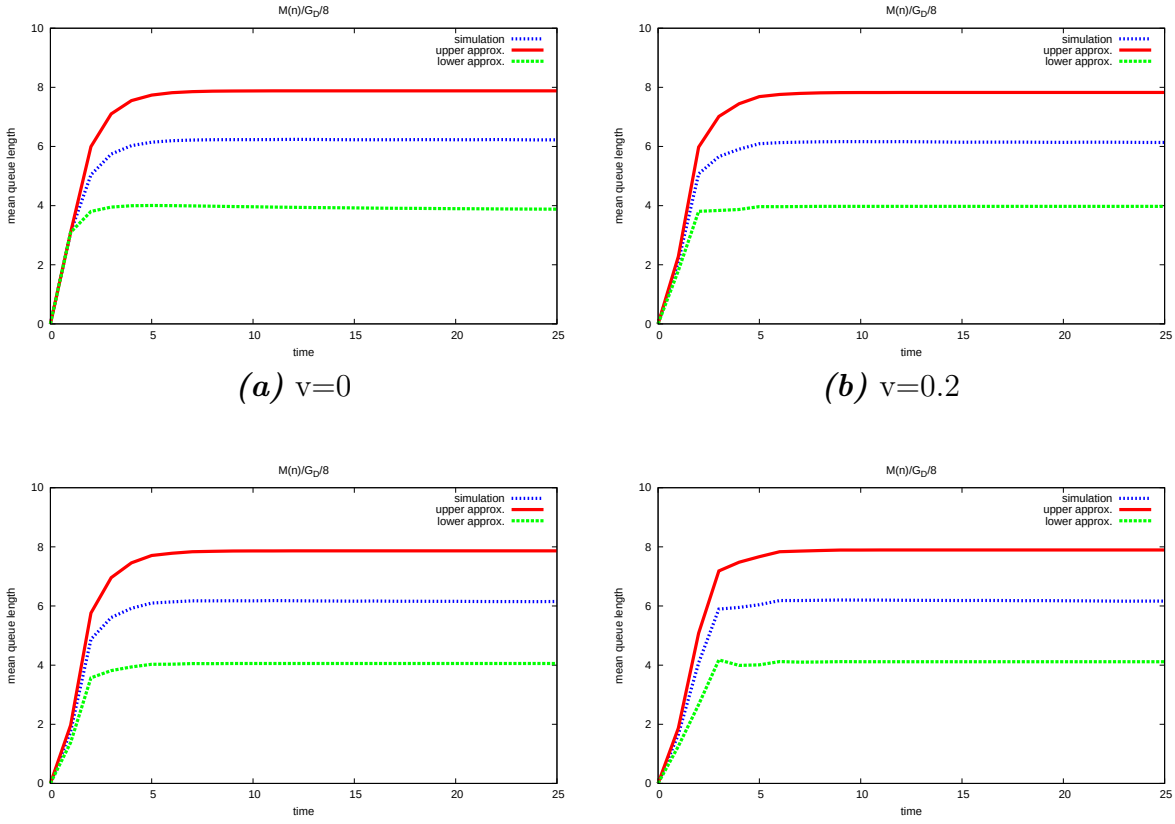
Figure 6.6: System with s=8 servers, maximum rho=1 (when no balking), and geometrical balking function 0.9^{1+n}, where n is the number in the queue, for different step sizes.



(a) step=1 ***(b)*** step=0.5

(c) step=0.25 ***(d)*** step=0.125

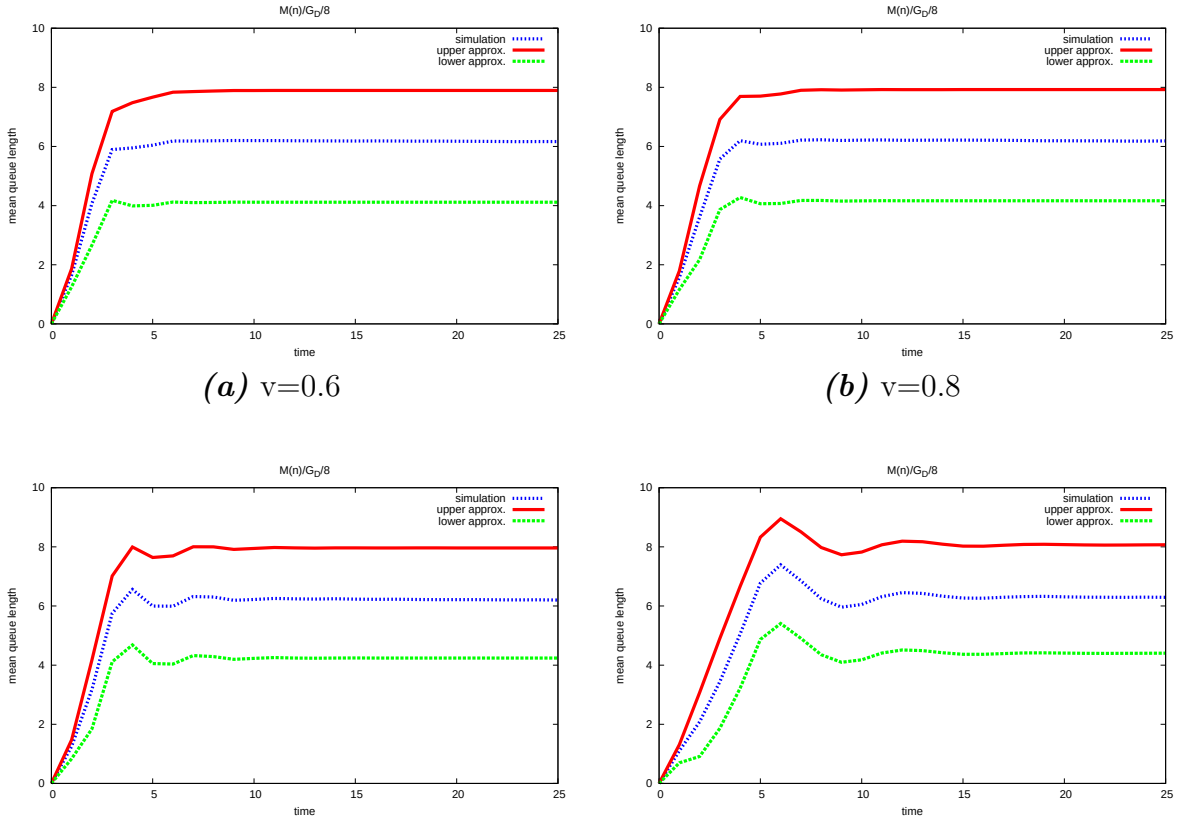
Figure 6.7: System with s=8 servers, maximum rho=0.75 (when no balking), and geometrical balking function 0.9^{1+n}, where n is the number in the queue, for different step sizes.



(a) v=0 ***(b)*** v=0.2

(c) v=0.4 ***(d)*** v=0.6

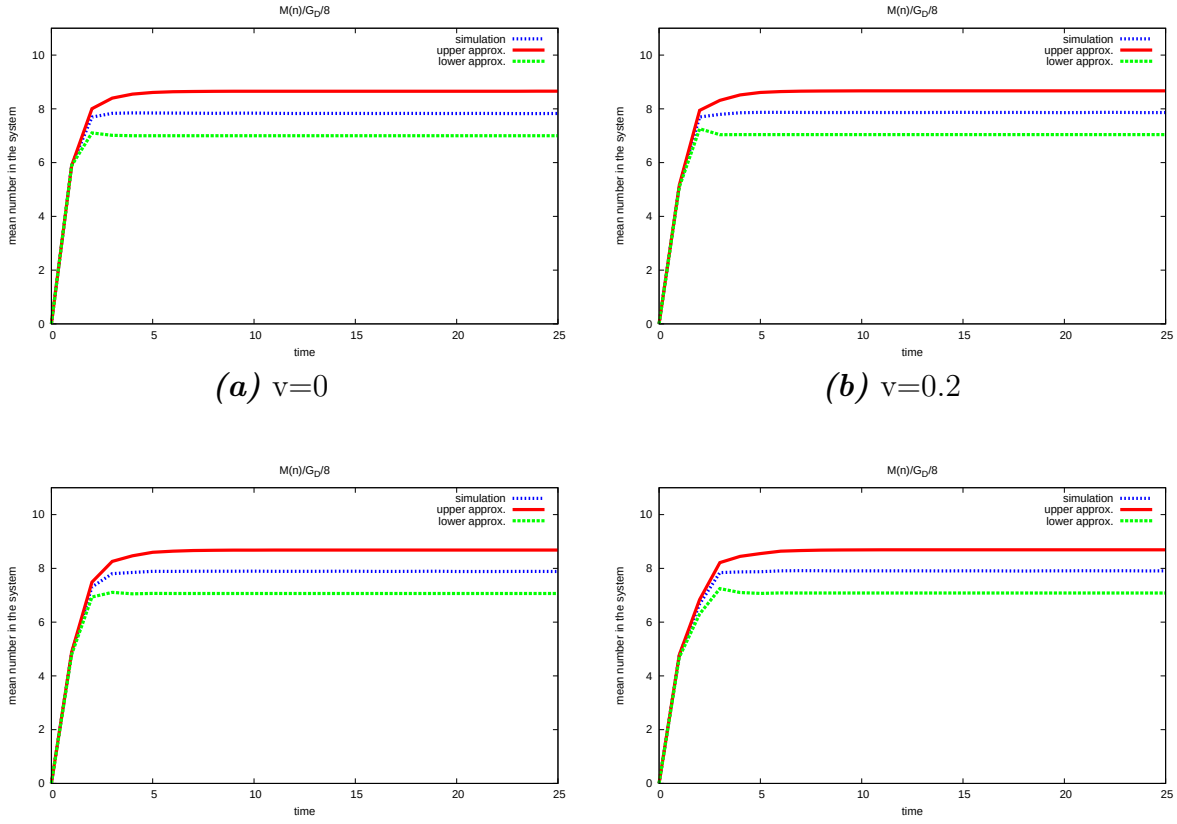
Figure 6.8: System with s=8 servers, step=0.5, maximum rho=2 (when no balking), and geometrical balking function 0.9^{1+n}, where n is the number in the queue, for different variances.



(a) v=0.6 ***(b)*** v=0.8

(c) v=1 ***(d)*** v=2

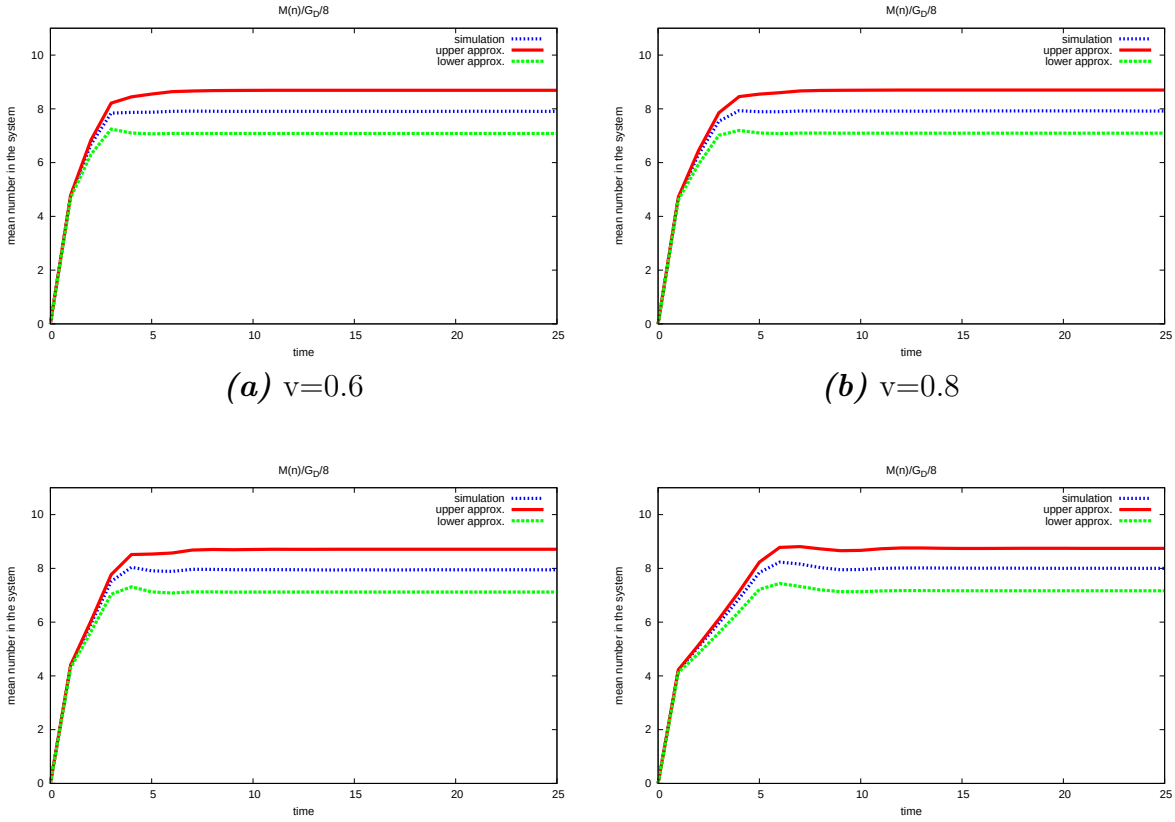
Figure 6.9: System with s=8 servers, step=0.5, maximum rho=2 (when no balking), and geometrical balking function 0.9^{1+n}, where n is the number in the queue, for different variances.



(a) v=0 ***(b)*** v=0.2

(c) v=0.4 ***(d)*** v=0.6

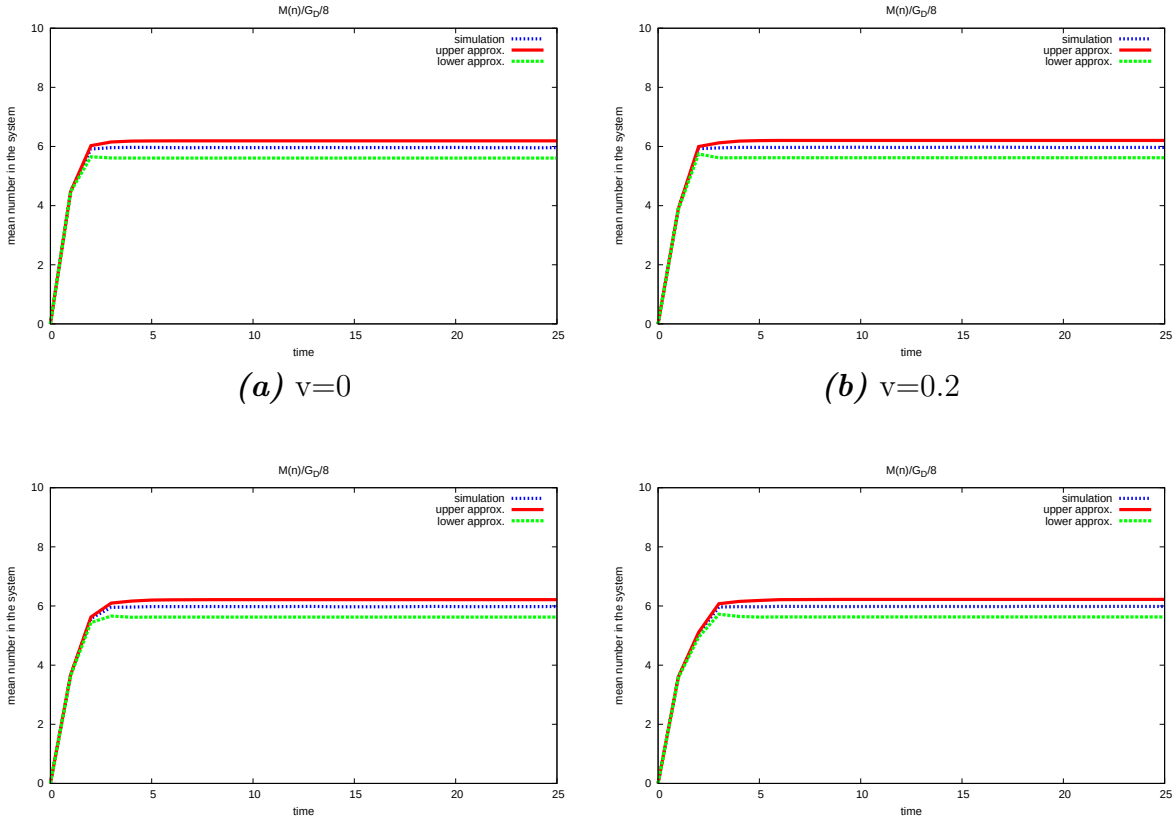
Figure 6.10: System with s=8 servers, step=0.5, maximum rho=1 (when no balking), and geometrical balking function 0.9^{1+n}, where n is the number in the queue, for different variances.



(a) v=0.6 ***(b)*** v=0.8

(c) v=1 ***(d)*** v=2

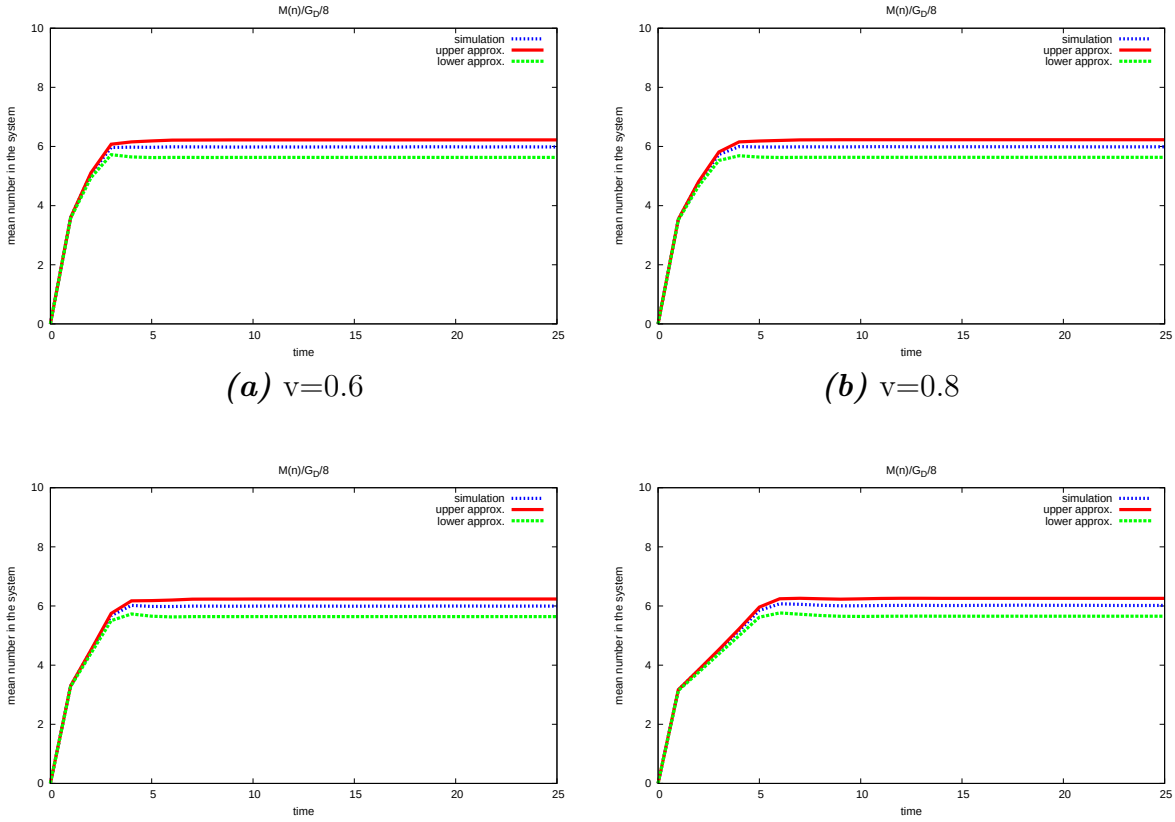
Figure 6.11: System with s=8 servers, step=0.5, maximum rho=1 (when no balking), and geometrical balking function 0.9^{1+n}, where n is the number in the queue, for different variances.



(a) v=0 ***(b)*** v=0.2

(c) v=0.4 ***(d)*** v=0.6

Figure 6.12: System with s=8 servers, step=0.5, maximum rho=0.75 (when no balking), and geometrical balking function 0.9^{1+n}, where n is the number in the queue, for different variances.



(a) v=0.6 ***(b)*** v=0.8

(c) v=1 ***(d)*** v=2

Figure 6.13: System with s=8 servers, step=0.5, maximum rho=0.75 (when no balking), and geometrical balking function 0.9^{1+n}, where n is the number in the queue, for different variances.

Chapter 7

Systems with state-dependent balking

7.1 Introduction

Up to now we have developed the theoretical and algorithmic framework to incorporate balking in the DTM algorithm, and we have ended up with two approximations of controllable accuracy. This enables us to use these approximate models to study systems with balking, and see how they perform. We start with an example system, and change the key parameters there in order to see how balking affects the performance of different systems. In this way a set of empirical results are obtained which are used to identify potentially interesting findings.

In particular these results are used to comment on: the lag between the arrival peak and the peak in congestion, the mean and percentiles of the number in the system, the distribution of the number in the system, and the effect of the service time distribution.

These insights are then used to establish two important findings for the practical application of the pointwise stationary approximation approach to systems with balking.

7.2 Test cases

In order to study systems with balking we set an example case and change key parameters, so that a wide range of systems are included in our investigations. The arrivals are assumed Poisson, and when facing a busy system balking occurs in a geometrical form.

The basic system under consideration has $s = 8$ servers and mean service time $\bar{t} = 6$ minutes. The arrival rate is non-homogeneous and varies sinusoidally with time. The overall average arrival rate takes 3 different values so that quiet, busy, and very busy systems are studied. The amplitude of the sine wave is also changed in order to include systems with mild variation ($\pm 20\%$ of the average value), and with larger variation ($\pm 40\%$ of the average value).

Figure 7.1 shows the basic form of the arrival rate used for our results. The system starts empty and receives an increasing arrival rate, which very soon, after 45 minutes reaches a peak value. The sine wave starts when this first peak appears. Thus the period of the sine wave was selected so that 2 peaks appear during our observation time, in this case corresponding to morning and evening rush hours. This arrival rate could describe a call centre which operates from 7 : 30 in the morning till 9 : 30 in the evening, having a peak arrival rate at about 8 : 00 (8 : 15) in the morning, and another one at 5 : 00 (5 : 15) in the evening. The period of the sinusoidal arrival rate is 9 hours and the system is observed during the 14 operating hours.

In order to make the time reference easier we set the unit of time equal to the mean service time. For this reason the unit of time is 6 minutes and thus the mean service rate is now 1 customer per unit of time. For the rest of this chapter we are going to measure time in these new units instead of referring to the actual time. For example the system is observed for 140 units of time, which corresponds to the 14 operating hours.

The step size used in the approximate algorithms is one eighth of the basic time unit, that is 0.75 minutes (or 45 seconds), so that the approximations are quite close. The first peak is achieved much faster than the second one. This will enable us to

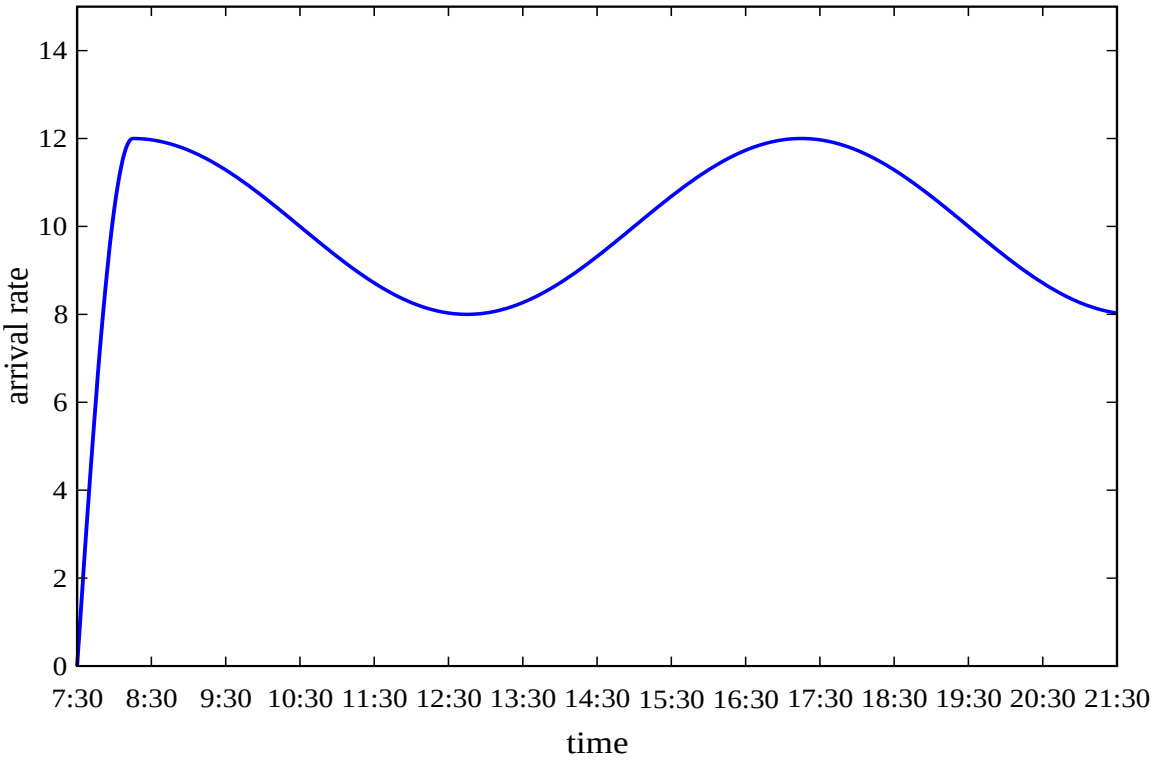


Figure 7.1: Arrival rate used for our set of results.

study 2 different types of peaks, however we have in mind that starting conditions also affect the first one.

The set of results produced is included at the end of this chapter (Figures 7.7 - 7.24). The input characteristics of each figure are summarised in Table 7.1 for quick reference. As can be seen from this table our results include various arrival rates and strengths of balking, as well variation to the amplitude of the sine wave. The purpose of this is to have a reasonably general set of empirical results in order to be as objective as possible when drawing conclusions based on them.

In the following paragraphs we describe the layout of each of Figures 7.7 - 7.24. Each figure consists of five graphs. Graph (a) shows the arrival rate used, the resulting upper approximation (red line), and lower approximation (green line) of the mean number in the system produced by our approximate models. The large dots indicate values of mean number in the system estimated by the pointwise stationary approximation, and will be discussed later. In graph (b) we wanted to show the lower and upper approximation together with the 95% and 5% percentile curves. To avoid confusion caused by the representation of many lines, as each approximation

has two associated percentile curves, we show one upper 95% (i.e. the 95% percentile curve that corresponds to the upper approximation) and one lower 5% (i.e. the 5% percentile curve that corresponds to the lower approximation).

Figure number	Balking coefficient	Average arrival rate	Amplitude of the sine
Figure 7.7	0.8 (strong)	8 (quiet)	±20% (mild)
Figure 7.8	0.8 (strong)	8 (quiet)	±40% (large)
Figure 7.9	0.8 (strong)	10 (busy)	±20% (mild)
Figure 7.10	0.8 (strong)	10 (busy)	±40% (large)
Figure 7.11	0.8 (strong)	18 (very busy)	±20% (mild)
Figure 7.12	0.8 (strong)	18 (very busy)	±40% (large)
Figure 7.13	0.9 (medium)	8 (quiet)	±20% (mild)
Figure 7.14	0.9 (medium)	8 (quiet)	±40% (large)
Figure 7.15	0.9 (medium)	10 (busy)	±20% (mild)
Figure 7.16	0.9 (medium)	10 (busy)	±40% (large)
Figure 7.17	0.9 (medium)	18 (very busy)	±20% (mild)
Figure 7.18	0.9 (medium)	18 (very busy)	±40% (large)
Figure 7.19	0.95 (weak)	8 (quiet)	±20% (mild)
Figure 7.20	0.95 (weak)	8 (quiet)	±40% (large)
Figure 7.21	0.95 (weak)	10 (busy)	±20% (mild)
Figure 7.22	0.95 (weak)	10 (busy)	±40% (large)
Figure 7.23	0.95 (weak)	18 (very busy)	±20% (mild)
Figure 7.24	0.95 (weak)	18 (very busy)	±40% (large)

Table 7.1: Summary of the queueing system characteristics associated with results in Figures 7.7-7.24.

Graphs (c), (d), and (e) show snapshots of the distribution of the number in the system taken at different points of time. These are indicated by the vertical lines. (The fitted Poisson and Normal curves will be explained later.) At any time point the distributions of the number in the system were very similar for both approximations. For this reason we show only the distributions associated with one of the approximations, and here we select the upper one. Graph (c) shows the distribution of the number in the system after $t = 30$ units of time, where the arrival rate has a medium value. On the same lines graph (d) shows this distribution at $t = 52.5$ units, where the arrival rate takes its minimum value, and graph (e) at $t = 97.5$ units, where the arrival rate takes its maximum value.

7.3 Performance of systems with balking

In this section we comment on those findings from our empirical results that we consider interesting. The way which we are going to describe our results is via contrasting balking systems with non-balking systems, which can be associated with the relevant literature.

In subsection 7.3.1 we are interested in how the congestion peak lags behind the arrival rate peak in non-stationary queueing systems with balking. This lag is important in practice as it can imply that the indicators of inadequate staffing levels often do not fully materialise until too late. It is also important in modelling terms as it is one of the main factors that affects the performance of PSA. Indeed, as we have seen in chapter 2, PSA models the behaviour of the system at each point of time using a stationary model with arrival rate equal to the value of the arrival rate function at that moment. As a consequence it fails to model this lag, and this adds to its imprecision, as seen in Figure 7.2. For this reason a lagged PSA has been proposed by Green and Kolesar [66], although the size of the lag can only be estimated.

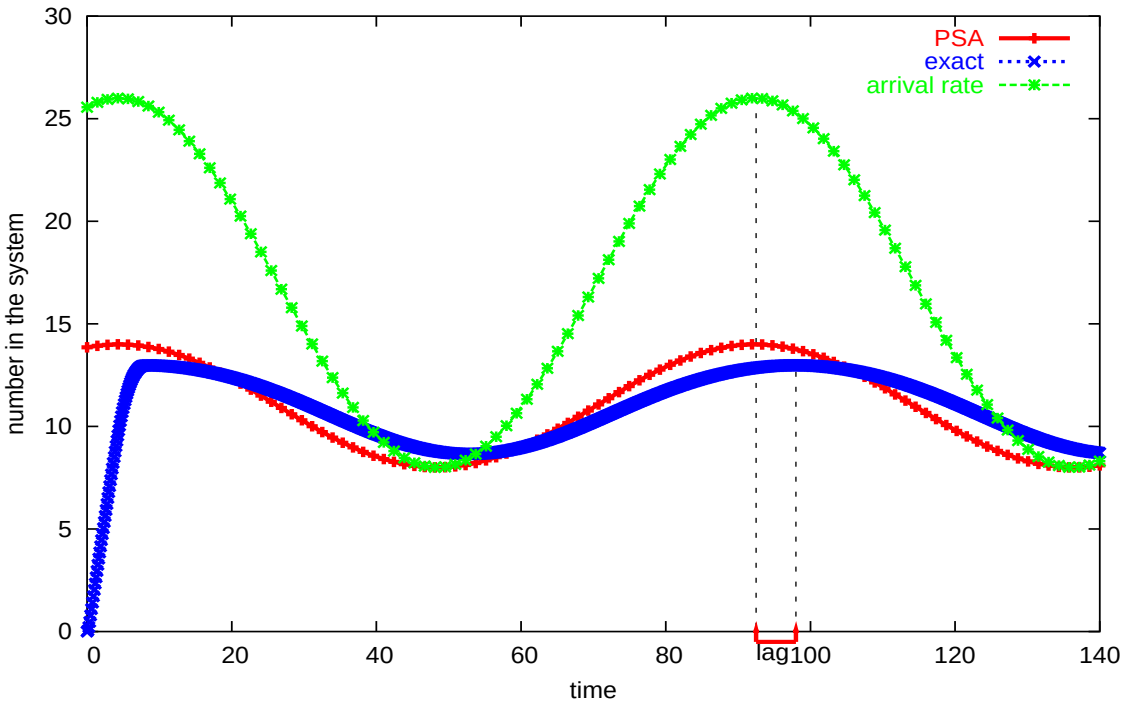


Figure 7.2: Lag between the peak of the actual solution and the peak in the PSA

In subsection 7.3.2 we then look at how balking affects the systems' performance, by looking the effect on the mean and the percentile curves for the number in the system. In subsection 7.3.3 we observe that the distribution of the number in the system is very close to normal, and give explanations why this happens. Finally subsection 7.3.4 investigates the effect of the service time distribution.

7.3.1 Lag between arrival peak and peak in congestion

It has been noted by many researchers that non-stationary queueing systems reflect a peak in the arrival rate in terms of a lagged peak in congestion, see for example [24], [67], [36], [35]. One obvious reason for this phenomenon, is the delay introduced by the service queueing process. However, the way in which the service process, or other factors, affect the magnitude of this lag, does not appear to have been studied systematically. We are interested in identifying these factors, as this should help us predict how the lag behaves in systems with balking. We mention below some findings from the relevant literature.

Eick et al. [36, 35] studied $M(t)/G/\infty$ queues. For these queues the number in the system has a Poisson distribution, and an exact expression of its mean exists due to Palm [68], and Khintchine [69]. Using this result the authors derive an expression for the lag for $M(t)/G/\infty$ queues with sinusoidal arrival rates. The sinusoidal arrival rate affects the lag only by its frequency. The other factor that determines the lag is the service time distribution. For example if the sinusoidal arrival rate is given by $\lambda(t) = \bar{\lambda} + \beta sin(\gamma t)$, and the service time follows a negative exponential distribution with mean equal to μ, the lag is $l = cot^{-1}(\mu/\gamma)/\gamma$, which is a decreasing function of γ. This means that sinusoidal arrival rates with high frequencies result in shorter lags than sinusoidal arrival rates with lower frequencies. On similar lines, if the service is deterministic with mean rate equal to μ then the lag equals $\mu/2$, and thus in this case it is independent of all the characteristics of the sinusoidal arrival rate.

The relatively simple results obtained for these systems are due to the fact that in infinite servers systems different customers do not interfere with each other. The

results obtained for infinite server systems have also been used to estimate the lags in the finite server systems. Green and Kolesar [66] propose a lagged PSA for estimating the peak congestion in $M(t)/M/s$ systems with periodic arrival rates. They first estimate the size of the time lag using an infinite server system, and then assume that the arrival rate in the finite server system is displaced by this amount along the time axis. The authors suggest that the lag predicted by the infinite server model underestimates the actual lag in the finite server system. Green and Kolesar [66], also report that in finite server systems, the lag increases as the peak probability of delay increases, and as the event frequency i.e. the average number of arrivals and service completions per period, decreases.

Up to now we have summarised the factors which affect the lag in systems without balking. In the rest of this section we focus on systems with balking. Looking at our full set of results we notice that systems which are subject to balking do not suffer from major lags. Moreover, systems with stronger balking seem to have shorter lags, and lags virtually vanish when very strong balking is present. For example if we compare the lag for the peak which occurs at time $t = 97.5$, in Figure 7.21(a) and in Figure 7.15(a) we have $Lag_{0.95} = 2.375$ units (≈ 14 minutes) for the weak balking system, and $Lag_{0.9} = 1$ unit ($= 6$ minutes) for the one with medium balking. If we continue this comparison with the corresponding system with strong balking, i.e. Figure 7.9(a), the lag is $Lag_{0.8} = 0.5$ units ($= 3$ minutes). This kind of comparison can be done among all systems with different balking levels, and the pattern that the stronger the balking the smaller the lag is confirmed by all cases. We also note that in each figure, see for example Figure 7.21, the lag associated with the first peak, at time $t = 7.5$ units, is larger than the one associated with the second peak, at time 97.5 units. The difference between these two peaks is that the arrival rate curve which leads to the first peak is much steeper than the one that leads to the second peak. However, because starting conditions might still have an effect at $t = 7.5$ units, we focused on the peak at time $t = 97.5$ units.

We can see from Figures 7.7-7.12(a) that for $b = 0.8$ (strong balking) there is no obvious visible difference between the time at which the peak arrival rate occurs, and

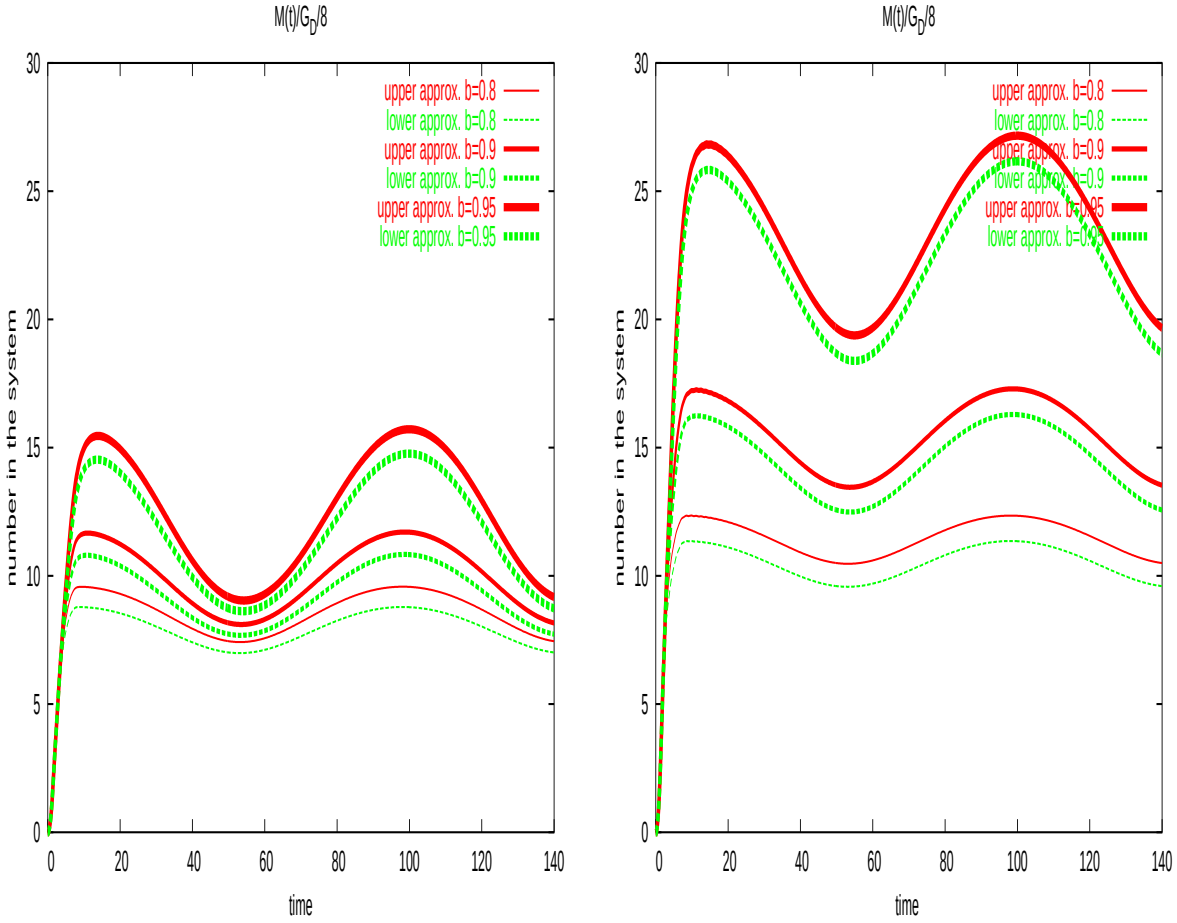
the time at which the peak number in the system occurs. In an effort to explain why systems with stronger balking have smaller lags, we notice that the peak probability of delay decreases for systems with stronger balking. As we have seen in the previous paragraphs, according to [66] the lag should decrease as well. This is because the probability of delay increases with the traffic intensity and with the average arrival rate [24]. These quantities decrease as balking increases. In systems with stronger balking less people will join when encountering the same congestion, and thus, these systems appear to have smaller average arrival rate and traffic intensity. As a result the probability of delay decreases leading to smaller lags as balking increases.

The above finding, that the stronger the balking the smaller the delay between the traffic intensity peak and the the number in the system peak could be used as an indication of balking occurrence. For example, in a call centre where management does not keep records of the people that abandon the system when encountering a busy line, observing how the peaks in congestion reflect the corresponding peaks in traffic intensity, would provide an indication on whether strong balking is present.

7.3.2 Mean and percentiles curves

When there is low or no balking (because the system is quiet, or because the strength of balking is weak) the curve of the mean number in the system has a similar shape to the arrival rate curve, though a time lag might be present as we have seen in the previous section. When balking is present, it tends to eliminate the increasing parts, resulting in smoother curves. For example in Figure 7.3 we contrast the different levels in congestion that we get due to different strengths of balking, for systems with 'busy' arrival rates (results taken from Figures 7.9, 7.15, and 7.21) and for systems with 'very busy' arrival rates (results taken from Figures 7.11, 7.17, and 7.23). As expected systems with higher arrival rates are more affected by the different levels of balking.

In graphs noted as (b) in Figures 7.7-7.24 we present the mean number in the system together with two percentile curves. By showing these percentiles we want to



(a) systems with medium arrival rate ***(b)*** systems with high arrival rate

Figure 7.3: Effect of balking on mean number in the system

give a fuller idea of how the system performs. The mean indicates what will be the average number in the system, however a call centre manager (for example) might also be interested in possible states in which the system can be found, e.g. customers facing too long queues, or servers being idle. Since our algorithm is able to calculate not only the mean number in the system, but also its distribution, it is easy to calculate any percentiles of interest. We can see by looking at any of the (b) graphs that the shape of the 5% and 95% percentile curves follows the shape of the mean curves. This can be expected since, as we discuss later, the number in the system follows a normal distribution approximately. In order to make this comparison easier we give an example in Figure 7.10(b) and in Figure 7.19(b). There, at some random points, we have highlighted these differences. We notice that in each graph $a^U \approx b^U \approx c^U$

and $a^L \approx b^L \approx c^L$.

Also, for each approximation, the distance between the percentile curve and the corresponding mean, seems to be more or less the same. For example in Figure 7.10(b) and in Figure 7.19(b) $a^U \approx a^L$, $b^U \approx b^L$ and $c^U \approx c^L$. This indicates that the distributions of the number in the system for the two approximations are fairly similar and symmetric. This is again consistent with their approximately Normal distributions which we will see later.

Finally, we can also notice that the distance between the percentile curve and the corresponding mean becomes smaller as the strength of balking increases, which indicates that the standard deviation of the number in the system decreases for systems with stronger balking. For example the highlighted distances in Figure 7.10(b), which refers to strong balking, are smaller than the highlighted distances in Figure 7.7(b) that refers to weak balking.

7.3.3 Distribution of the number in the system

In this section we are concerned with the probability distribution of the number in the system. Since we are dealing with non-stationary systems this distribution is time dependent. We initially remind the reader of available results concerning the distribution of the number in the system, when arrival rates are not state dependent, which we then combine with balking ideas, in order to conjecture what to expect for the probability distributions of non-stationary systems with balking.

(a) Approximately Normal distribution

If we temporarily ignore balking, the systems of interest fall in the general category of multi-server non-stationary $M(t)/G/s$ systems. However, there is very little information about the distribution of the number in the system for these systems. This is because exact or approximate analysis is difficult for non-stationary systems. Even in studies about $M(t)/M/s$ systems with sinusoidal arrival rates, the above issue is not addressed [24]. Most of the times the main concern is the mean number

in the system and estimations even for the variance are not available, see for example [24]. The only time-dependent system for which the distribution of the number in the system is known, is the $M(t)/G/\infty$, see for example [36].

In our attempt to investigate the time-dependent distribution of the number in the system, we consider the congestion level at the point of time of interest as a key factor. This is motivated by the relevant literature, where distributions of the number in the system can be approximated or estimated for extreme levels of congestion. For this reason we consider 3 categories of congestion: very busy, busy, and quiet systems.

We are first concerned with times at which the system is very busy. Worthington [70] suggested that for a stationary finite population system, $M(n)/G/s//N$, with high traffic intensity, the distribution of the number in the system can be approximated by an appropriate normal distribution. (We remind the reader here that finite population systems can be also seen as systems with balking which decreases linearly with the number in the system, while in the systems we consider, balking appears only when a queue is formed, and the strength of this balking depends geometrically on the number in the queue.) The explanation proposed in [70] for the normal approximation for $M(n)/G/s//N$ busy systems, is achieved in two steps. First, the system is approximated with s independent single-server queues ($M(n)/G/1//\frac{N}{s}$), since the servers are (almost) always busy. Then, the random variable which represents the number in the overall system, is calculated as the summation of s random variables, each of which represents the number in one of the s subsystems. Due to the central limit theorem a summation of s independent identically distributed (i.i.d.) random variables follows a Normal distribution when s is large. In fact the approximation gave small errors even when s was not large. The mean and the variance of this distribution were then estimated empirically.

There is no reason why a similar argument should not hold for any busy system with balking and with many servers, although different calculations would be needed to estimate the mean and the variance. As a result, under heavy traffic assumptions, we conjecture that the number in the system follows a normal distribution. In order to see whether this is consistent with our results, we focus on our results for systems

and times where the congestion is very high. This selection is based on the mean number in the system and is assisted by comparing the lower percentile curve with the number of servers. Results for systems and times of high congestion are presented in Figures 7.17(e), 7.18(e), 7.21(e), 7.22(e), 7.23(c, d, e), 7.24(c, e). From these graphs it is obvious that the normal distribution closely matches the probability distribution of the number in the system. We also note that in all cases the Poisson distribution which matches the mean has larger variance than the actual distribution, thus, the mean is an upper bound for the variance for busy systems.

Moving on the antipode side, which is very quiet systems, we expect them to behave as $M(t)/G/\infty$ systems. This is because in very quiet systems the queue is negligible, and thus we can assume that an arrival always finds a free server. Moreover, very quiet systems with balking will behave in the same way, as balking only happens in our case when the number in the system exceeds the number of servers. In $M(t)/G/\infty$ systems the number in the system follows a Poisson distribution. The mean for the stationary $M/G/\infty$ systems is $m = \frac{\lambda}{\mu}$ [13], and for the non-stationary $M(t)/G/\infty$ systems is $m(t) = \frac{E[\lambda(t-S_e)]}{\mu}$ [36], that is a weighted average of arrival rates before time t. Hence we expect that when the mean number in the system is substantially smaller than the number of servers, the number in the system will match a Poisson distribution.

We remind the reader here that Poisson distributions with large means can be approximated by normal distributions. We refer again to our set of results, and concentrate on systems and times when they are very quiet. The selection of very quiet systems is based on the mean number in the system and is assisted by comparing the upper percentile curve with the number of servers. Such systems can be found in Figures 7.7(d), 7.8(d), 7.10(d), 7.13(d), 7.14(d), 7.16(d), 7.19(d), 7.20(d), 7.22(d). In all these cases both a Poisson and a normal distribution can be used in order to describe the number in the system.

The last category is systems which are neither very busy nor very quiet. There were no relevant results in the literature for these systems, and as a result we have no theoretical grounds to conjecture on the distribution of the number in the sys-

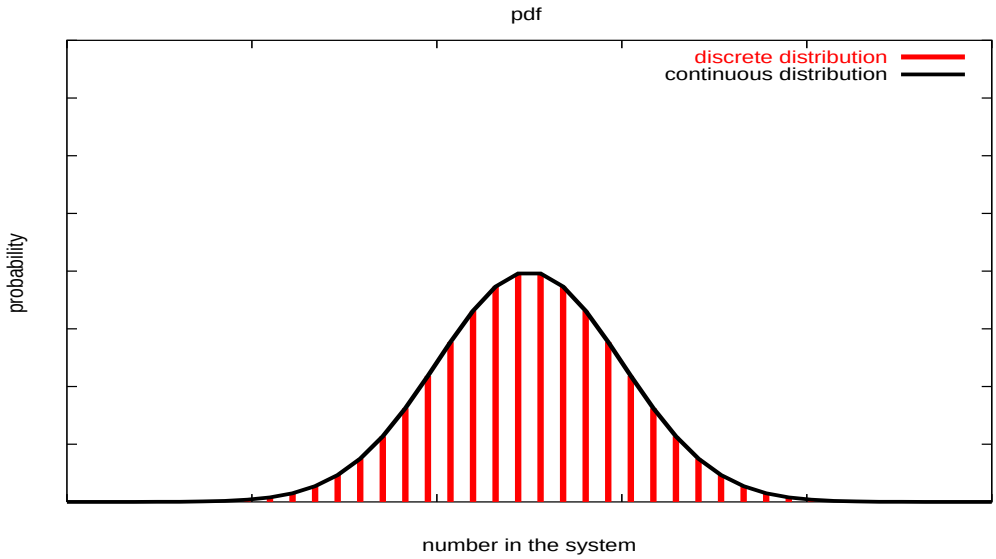


Figure 7.4: Discrete distribution obtained by taking integer samples from a normal distribution.

tem. However, the graphs from our results that correspond to this situation are Figures 7.7(c, e), 7.8(c, e), 7.9(c, d, e), 7.10(c, e), 7.11(c, d, e), 7.12(c, d, e), 7.13(c, e), 7.14(c, e), 7.15(c, d, e), 7.16(c, e), 7.17(c, d), 7.18(c, d), 7.19(c, e), 7.20(c, e), 7.21(c, d), 7.22(c), and as can be seen the normal distribution again provides a good approximation for the distribution of the number in the system.

In conclusion, the probability distribution of the number in the system closely matches a normal density function at integer values. All our empirical results have showed this matching. In other words we can form the discrete distribution of the number in the system by allocating the values of a continuous normal function at integer values, as in Figure 7.4. Hence if $P_t(n)$ denotes the probability that there are n customers in the system at time t,

$$P_t(n) \approx \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(n-m)^2}{2\sigma^2}}$$

where m is the mean and σ the standard deviation of the number in the system. Note that strictly we need to prove that $\sum_0^\infty P_t(n) = 1$, for $\{P_t(n)\}$ to represent a probability distribution. This is done in Appendix C for distributions with mean values that are not close to zero.

(b) Standard deviation

As noted in Section 7.3.2, and by further comparison of Figures 7.7-7.12 with Figures 7.13-7.18 and Figures 7.19-7.24 we can also observe that systems with stronger balking have smaller standard deviations of number in the system. We give an explanation of this phenomenon by referring again to the nature of balking systems. This nature is an example of systems with negative feedback. Indeed the fact that increased congestion discourages new arrivals to enter the system, indicates presence of a negative feedback mechanism. In general negative feedback systems act to maintain their homeostasis, that is, to keep themselves in a constant state. For systems with balking the negative feedback appears only when the system's state is relatively high. However, this is still homeostatic behaviour for upward trends. In this way balking limits the possible states in which the system can be found. As a result systems with balking have smaller standard deviations, and the stronger the balking, the smaller the standard deviation of the number in the system.

7.3.4 The effect of the distribution of service time

In this section we remind the reader about the known effects of the service time distribution for general queueing systems, and then focus on systems with balking. For general queueing systems knowing only the mean of the service time distribution is not enough to have an accurate description of how the system performs. For example for an $M/G/1$ system the mean number in the system is given by the Pollaczek-Khinchin mean-value formula [13]:

$$m = \frac{\lambda}{\mu} + \lambda^2 \frac{\frac{1}{\mu^2} + \sigma^2}{2(1 - \frac{\lambda}{\mu})}$$

This relationship indicates that the mean number in the system depends not only on the mean service rate μ, but also on the standard deviation σ of the service time distribution. Moreover, the mean number in the system increases as the variation in the service time distribution increases.

For more than one server an analytic expression for the mean number in the system, or in the queue does not exist, unless the service time distribution is negative exponential. However, there are tabulated values for the steady state mean queue length for $M/D/s$ systems, see for example [15]. In Table 7.2 we compare these values with the corresponding results for $M/M/s$ systems, for different traffic intensities. We again note that the systems with the higher variance of service time (i.e. negative exponential service time has $variance = mean^2$) have higher congestion levels than those with the lower variance of service time (i.e. deterministic service time has 0 variance). Indeed, the steady state mean queue lengths of the systems with markovian service time distributions are almost twice the mean queue lengths of the corresponding systems with deterministic service time distributions.

traffic intensity (rho)	$M/M/1$	$M/D/1$	$M/M/8$	$M/D/8$	$M/M/15$	$M/D/15$
0.5	0.5	0.25	0.059	0.03728	0.01129	0.00801
0.8	3.2	1.6	1.8306	0.9725	1.2768	0.70123
0.9	8.1	4.05	6.313	3.2398	5.4237	2.8198
0.95	18.05	9.025	16.039	8.1163	14.952	7.6099
0.99	98.01	49.005	95.812	48.014	94.556	47.433

Table 7.2: Comparison of steady state mean queue length for systems that differ in the service time distribution.

When dealing with non-stationary systems we do not know whether the service time distribution has such a distinctive effect as for the steady state of stationary systems. While modelling a call centre without balking, Chassioti and Worthington [71] gave an example of the effect of the service time distribution in non-stationary systems. Their results suggested that accurate representation of higher than first moment of the service time distribution is not so important as other factors, although changing the variance of the service time distribution still had a noticeable effect on congestion. This effect seemed to be smaller than for stationary systems.

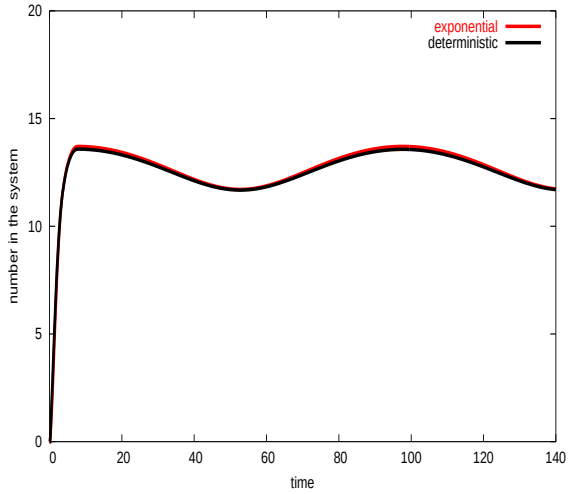
We now focus on systems with balking. Balking constrains the range of possible states, since the higher the number in the system the more difficult it becomes for new arrivals to join this system, and thus, high states which would occur if balking was

not present, will not occur as often. This reduces the mean number in the system, and also the standard deviation of the number in the system, since the range of low states does not increase.

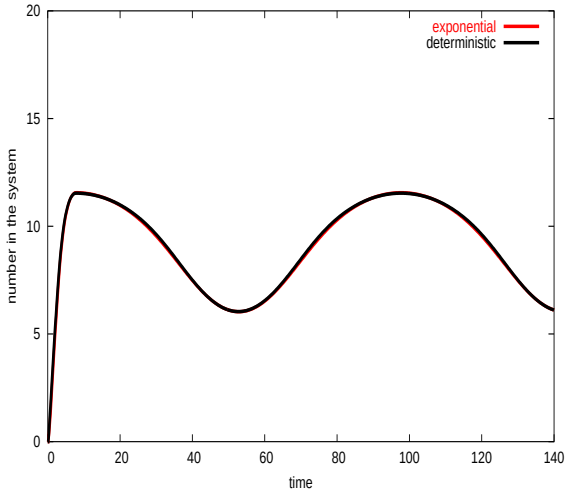
Let us now consider two systems with the same arrival process and with service time distributions with the same mean but different higher moments. Due to the variability in service times the servers in the system with the more variable service time distribution will be idle for more time than the ones in the less variable system. However, both systems are subject to the same workload. As a result, at time t, the number in the system, which is the number arrived minus the number that have been served, will be larger for the more variable system. Hence for systems without balking variability in service time tends to increase the mean number in the system.

However, in balking systems, increased congestion increases balking, and hence tends to reduce the mean number in the system. Thus, we conclude that we have two competing mechanisms, variability versus balking. As a result, we expect that the increased variability in the service time distribution will have smaller effect for systems with balking.

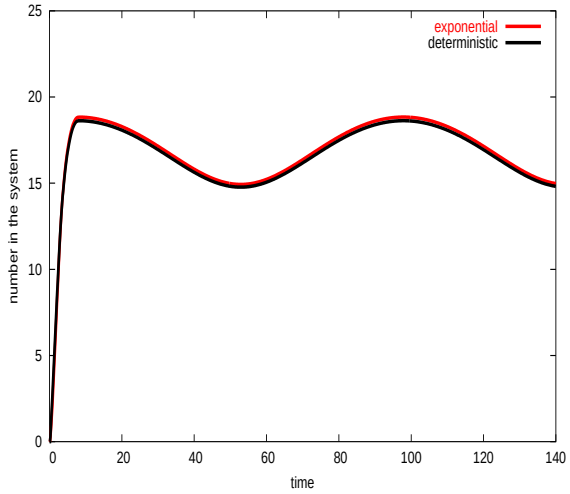
This can be seen clearly from Figure 7.5 in which the effect of the service time distribution is presented for one of the approximations (here we show the upper approximation) for systems with different balking coefficients. In all cases two outputs were obtained, one for a deterministic service time distribution and one for a discrete analogue of a negative exponential one. From these graphs we observe the close proximity of the mean number in the system for systems with different service time distributions. This reinforces our previous results presented in Chapter 6 in Figures 6.8-6.13, which concerned systems with balking coefficient equal to 0.9 and different service time distributions, to support our conjecture that systems with balking are insensitive to the service time distribution.



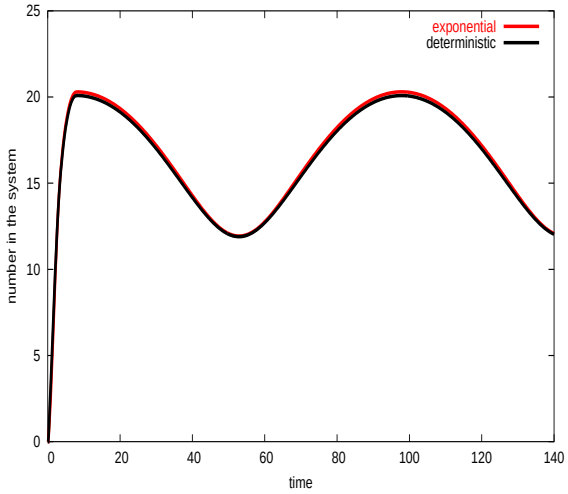
(a) balking coef.=0.8, sine amp.=±20%



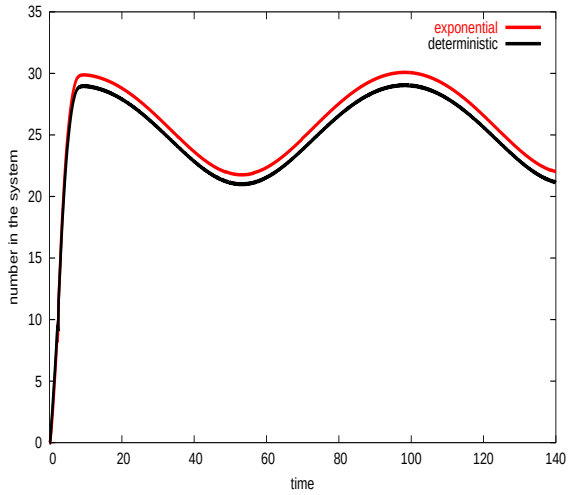
(b) balking coef.=0.8, sine amp.=±40%



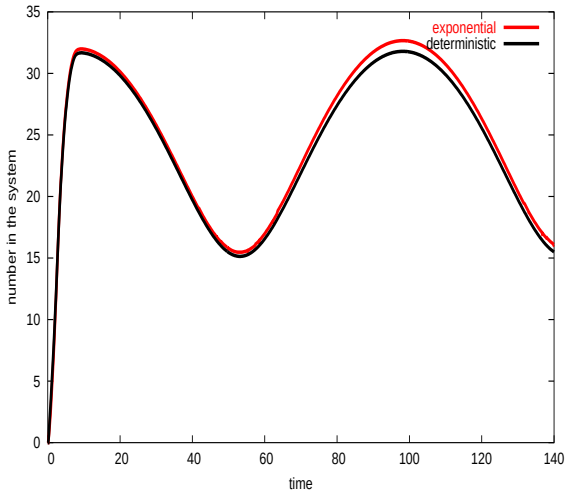
(c) balking coef.=0.9, sine amp.=±20%



(d) balking coef.=0.9, sine amp.=±40%



(e) balking coef.=0.95, sine amp.=±20%



(f) balking coef.=0.95, sine amp.=±40%

Figure 7.5: Systems with deterministic and negative exponential service time distributions for different balking coefficients and sine variations.

7.4 The PSA for systems with balking

The basis of the pointwise stationary approximation (PSA) is that it assumes that at each moment steady-state is achieved, and it uses the instantaneous arrival rate in order to estimate the system's performance at that instant. As a result PSA tends to perform poorly for systems that achieve steady state slowly (e.g. where peaks and troughs in congestion lag significantly behind peaks and troughs in arrival rates), and cannot be applied at all in cases where the system does not settle to steady state. In this section we are interested in how successfully the PSA approach can be applied for systems with balking.

Our results to date enable us to make two quite strong statements about the likely usefulness of the PSA approach in time-dependent systems with balking. In particular:

1. Although the instantaneous offered load in a balking system may be greater than 1, the system will settle to steady state due to the negative feedback that balking imposes on it. Hence PSA can always be applied for systems with balking.

2. Because lags between arrival rates and congestion levels reduce as balking increases, there is reason to hope that PSA may be particularly applicable in systems with strong balking.

This second conjecture is well supported by our empirical results. In all of Figures 7.7(a) to 7.24(a) the large dots indicate the PSA result for the mean number in the system using both the upper and lower approximations are indicated at $t = 30$, $t = 52.5$, $t = 75$, $t = 97.5$ (i.e. two symmetric points a minimum and a maximum). We can notice that for strong balking (i.e. b=0.8, Figures 7.7(a) to 7.12(a)) the PSA estimations coincide with the values that are calculated from our approximations, while when balking is weak (i.e. b=0.8, Figures 7.19(a) to 7.24(a)) the PSA values are not as accurate, however they are still very close to the ones produced by the approximations.

7.5 Calculation of steady-state measures for very busy systems

In the previous section we saw that PSA generally performs well for systems with strong balking. This indicates that at each point of time these systems behave more or less as if at steady state. For this reason we can assume that at each point of time steady state has been achieved, and hence that the mean arrival rate should match the mean departure rate.

If we now restrict ourselves to busy systems, i.e. systems in which the servers are (almost) always very busy, then the above steady state finding implies that at each point of time

$$E\{\lambda_t(n)\} = s\mu \tag{7.1}$$

where s is the number of servers, and μ is the service rate. The arrival rate, at time t, when there are n customers in the system, is:

$$\lambda_t(n) = \begin{cases} \lambda_t, & n < s, \\ \lambda_t b^{n-s+1}, & n \geq s \end{cases}$$

We have seen from our results in section 7.3.3 that the distribution of the number in the system could be approximated by a normal density function at integer values. Since here we are considering very busy systems, $P_t(n)$ (probability that we have n customers in the system at time t) can also be written as:

$$P_t(n) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(n-m)^2}{2\sigma^2}} \tag{7.2}$$

The mean arrival rate at time t is:

$$E\{\lambda_t(n)\} = \sum_{n=0}^{\infty} \lambda_t(n) P_t(n)$$

Hence, from Equation (7.2) we have:

$$\begin{aligned} E\{\lambda_t(n)\} &= \sum_{n=0}^{s-1} \lambda_t \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(n-m)^2}{2\sigma^2}} + \sum_{n=s}^{\infty} \lambda_t b^{n-s+1} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(n-m)^2}{2\sigma^2}} \\ &= \sum_{n=0}^{s-1} \lambda_t (1 - b^{n-s+1}) \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(n-m)^2}{2\sigma^2}} + \sum_{n=0}^{\infty} \lambda_t b^{n-s+1} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(n-m)^2}{2\sigma^2}} \end{aligned}$$

The first term takes very small values since the probabilities for $n < s$ can be considered negligible for very busy systems. Thus:

$$\begin{aligned} E\{\lambda_t(n)\} &\approx \lambda_t b^{-s+1} \frac{1}{\sigma\sqrt{2\pi}} \sum_{n=0}^{\infty} b^n e^{-\frac{(n-m)^2}{2\sigma^2}} \\ &= \frac{\lambda_t b^{-s+1}}{\sigma\sqrt{2\pi}} \sum_{n=0}^{\infty} e^{ln(b)^n} e^{-\frac{(n-m)^2}{2\sigma^2}} \\ &= \frac{\lambda_t b^{-s+1}}{\sigma\sqrt{2\pi}} \sum_{n=0}^{\infty} e^{nln(b)} e^{-\frac{n^2-2nm+m^2}{2\sigma^2}} \\ &= \frac{\lambda_t b^{-s+1}}{\sigma\sqrt{2\pi}} \sum_{n=0}^{\infty} e^{-\frac{n^2-2n(m+\sigma^2 ln(b))+m^2}{2\sigma^2}} \\ &= \frac{\lambda_t b^{-s+1}}{\sigma\sqrt{2\pi}} \sum_{n=0}^{\infty} e^{-\frac{n^2-2n(m+\sigma^2 ln(b))+(m+\sigma^2 ln(b))^2-\sigma^4 ln^2(b)-2m\sigma^2 ln(b)}{2\sigma^2}} \\ &= \frac{\lambda_t b^{-s+1} e^{\frac{ln(b)(\sigma^2 ln(b)+2m)}{2}}}{\sigma\sqrt{2\pi}} \sum_{n=0}^{\infty} e^{-\frac{[n-(m+\sigma^2 ln(b))]^2}{2\sigma^2}} \end{aligned}$$

However $\frac{1}{\sigma\sqrt{2\pi}} \sum_{n=0}^{\infty} e^{-\frac{[n-(m+\sigma^2 ln(b))]^2}{2\sigma^2}}$ is the summation of the normal probabilities at integer values, with mean= $m + \sigma^2 ln(b)$ and variance= σ^2. As established earlier, these probabilities sum to 1. As a result from the above equation we have :

$$E\{\lambda_t(n)\} \approx \lambda_t b^{-s+1} e^{\frac{ln(b)(\sigma^2 ln(b)+2m)}{2}}$$

We substitute this expression in Equation (7.1) in order to find an estimation for

the mean :

$$\lambda_t b^{-s+1} e^{\frac{ln(b)(\sigma^2 ln(b)+2m)}{2}} = s\mu \Rightarrow$$
$$e^{\frac{ln(b)(\sigma^2 ln(b)+2m)}{2}} = \frac{s\mu b^{s-1}}{\lambda_t} \Rightarrow$$
$$\frac{ln(b)(\sigma^2 ln(b)+2m)}{2} = ln(\frac{s\mu}{\lambda_t}) + ln(b)^{s-1} \Rightarrow$$
$$\frac{1}{2}\sigma^2 ln(b) + m = \frac{1}{ln(b)}\left[ln(\frac{s\mu}{\lambda_t}) + (s-1)ln(b)\right] \Rightarrow$$
$$\frac{1}{2}\sigma^2 ln(b) + m = \frac{1}{ln(b)} ln(\frac{s\mu}{\lambda_t}) + (s-1) \Rightarrow$$
$$m = (s-1) + \frac{1}{ln(b)} ln(\frac{s\mu}{\lambda_t}) - \frac{1}{2}\sigma^2 ln(b) \tag{7.3}$$

The above relationship, though it assumes very busy systems and existence of non-trivial balking, is very useful since it provides an analytical formula that can be easily applied in order to calculate the mean number of customers in the system at any point of time. It can also be seen to contain a self validation property, since in cases that the estimated m is not large enough (compared to the number of servers) the estimation should be discarded as misleading as it does not refer to a very busy system.

The practitioner who wants to apply PSA could use the above formula for very busy systems with balking. The only dilemma is what value of σ to use as a standard deviation of the number in the system. Indeed the standard deviation is unknown, however extreme values can be used to get a range of possible levels of congestion. This is because we have seen from our empirical results that the Poisson distribution was more variant than the actual distribution for busy systems, so $\sigma^2 \leq m$; and obviously $\sigma^2 \geq 0$.

If we assume $\sigma^2 = m$ in Equation (7.3) we have an upper bound for m:

$$m_U = (s-1) + \frac{ln(s\mu) - ln(\lambda_t)}{ln(b)} - \frac{m_U\, ln(b)}{2}$$
$$\Rightarrow \quad m_U = \frac{1}{1+\frac{ln(b)}{2}}\left[(s-1) + \frac{ln(s\mu) - ln(\lambda_t)}{ln(b)}\right] \tag{7.4}$$

If we assume $\sigma^2 = 0$ in Equation (7.3), we get a lower bound for m:

$$m_L = (s-1) + \frac{1}{ln(b)} ln(\frac{s\mu}{\lambda_t})$$

However the derivation of Equation (7.3) required σ to be 'big'. Hence it is safer to deal with $\sigma^2 = 0$ directly, by noting that it implies a deterministic distribution of number in the system. In this case Equation (7.1) gives:

$$s\mu = \lambda_t b^{m_L - s + 1}$$

which also leads to:

$$m_L = (s-1) + \frac{1}{ln(b)} ln(\frac{s\mu}{\lambda_t}) \tag{7.5}$$

Formulae (7.4) and (7.5) are clearly very easy to evaluate, and only require the number of servers, the mean service rate, the balking coefficient and the arrival rate function. We therefore look briefly at the quality of results these formulae might produce in practice.

In Table 7.3 we compare the mean value of the number in the system as calculated from above formulae (m_L, m_U), with the mean values calculated by our DTM approximations for different strengths of balking. The systems under consideration are the same as in Figures 7.11, 7.17, 7.23, i.e. very busy, with mild variation of the amplitude of the arrival rates. We select time $t = 30$ units, for this comparison.

Balking coefficient (b)	Estimates of Mean (using formulae (7.4) & (7.5))	Lower & Upper bounds for Mean (using DTM approximations)
0.8 (strong)	(10.634, 11.969)	(10.5874 , 11.5491)
0.9 (medium)	(14.696, 15.514)	(14.718 , 15.7147)
0.95 (weak)	(22.81, 23.41)	(23.2764 , 24.2764)

Table 7.3: Mean queue lengths estimated by the formulae and by the DTM approximations.

We can observe that the formulae give values very close to the ones calculated by our algorithms. Note that we already know from our empirical work that the true values of mean number in the system lie in the DTM ranges. Whilst the lower and

upper values from the formulae do not necessarily bound the true value, they clearly give potentially useful approximations.

A further comparison between the actual values and the values from the above formulae are presented in Figure 7.6. The results are for the same system as in Figure 7.23: a very busy system which experiences weak balking ($b = 0.95$) with mild variation of the amplitude of the sine. According to our earlier results the actual solution will be between the upper and the lower approximations. From Figure 7.6 we see that the curves for the two formulae intertwine with the two approximations and give generally very close values to them.

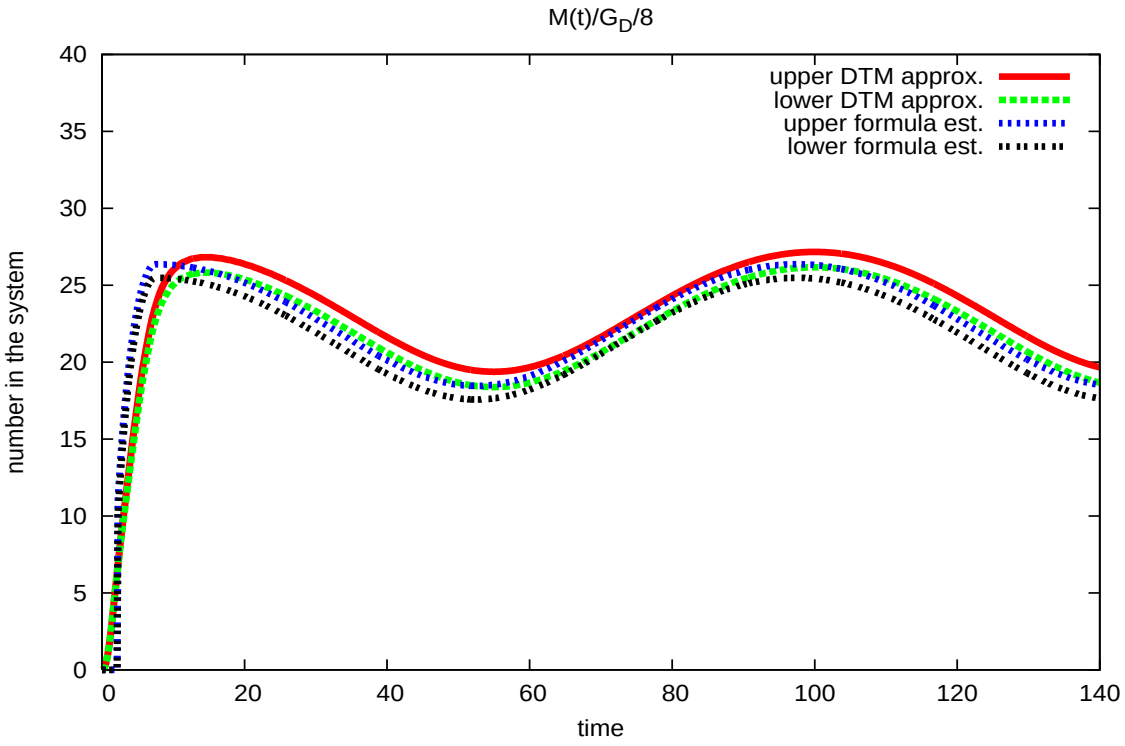


Figure 7.6: Comparison between exact results and formula (7.3).

In conclusion this method clearly has the potential to provide a good quality indication of the system's performance for busy call centres which experience sufficiently high levels of balking. The two formulae are very simple to evaluate, and the information requirements are minimal, i.e. the instantaneous arrival rate, mean service time, number of servers and balking factor.

7.6 Summary

In this chapter we have studied the performance of systems with balking by using the DTM approximations proposed in previous chapters. The systems under consideration included a range of realistic arrival rates and strengths of balking.

Based on this set of empirical results a number of conjectures have been generated, investigated and discussed:

(i) stronger balking leads to smaller lags between peaks in arrival rates and the corresponding peaks in congestion levels (see Section 7.3.1);

(ii) other things being equal, balking has a more marked effect for systems with higher arrival rates (see Section 7.3.2);

(iii) the probability distribution of number in the system for systems with balking closely matches a Normal density function at integer values (see Section 7.3.3, 7.3.2);

(iv) systems with balking are insensitive to second and higher moments of the distribution of service time (see Section 7.3.4).

(v) the PSA performs better for systems with balking than for systems without balking (see Section 7.4);

(vi) for very busy balking systems the PSA can be reduced to a pair of simple formulae of reasonably accuracy for the mean number in the system (see Section 7.5).

Initially the lag between a peak in the arrival rate and a peak in congestion was studied. Unlike systems that do not experience balking, systems with balking have insignificant lags. In Section 7.3.1 we conjectured that the stronger the balking the smaller the lag between the arrival rate peak and the congestion peak, and we have given reasons on why this happens.

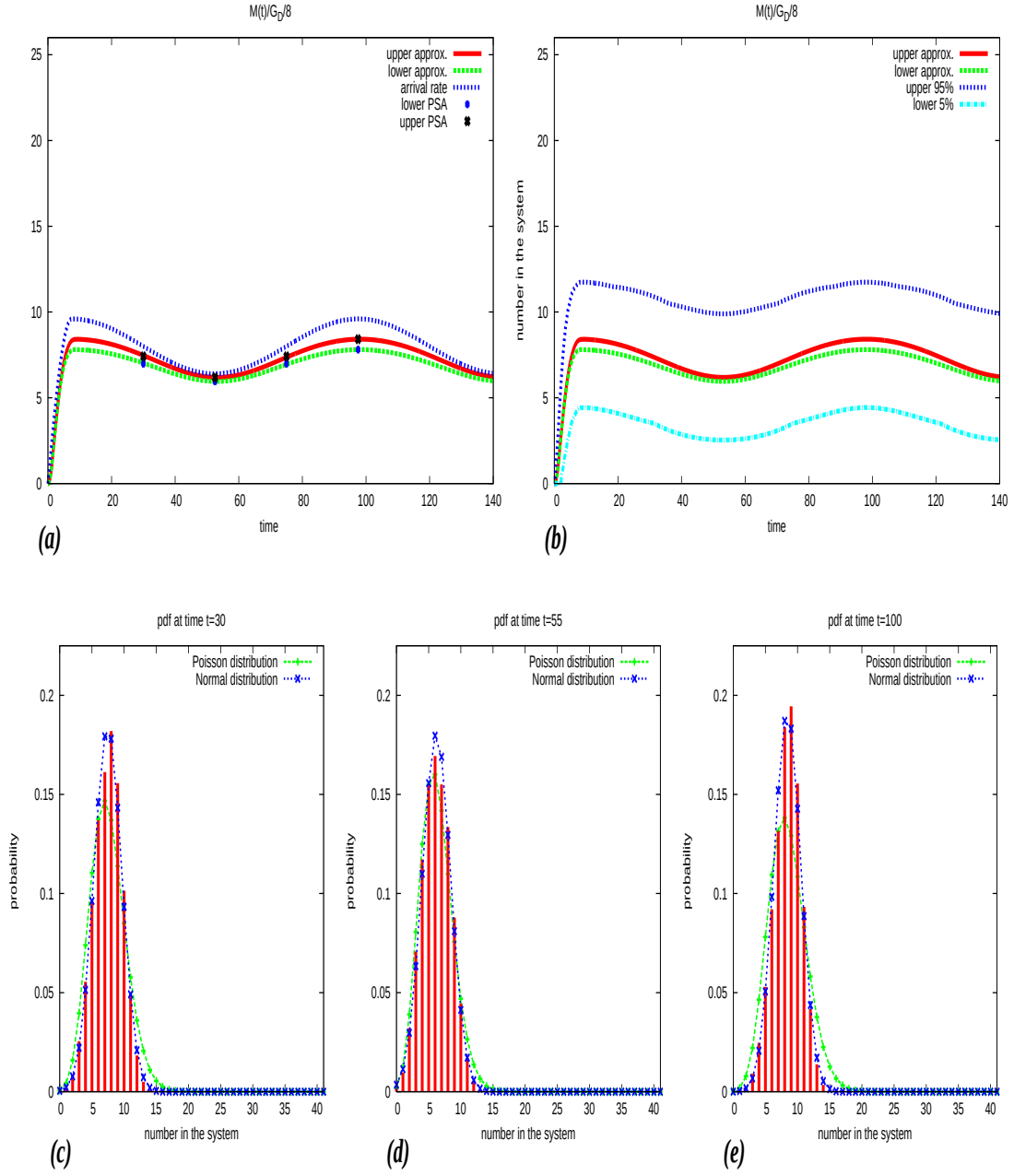


Figure 7.7: A quiet system that has a sinusoidal arrival rate with mean $\lambda = 8$, mild variation defined by the amplitude which is 20% of the mean, service rate $\mu = 1$, $s = 8$ servers, and strong balking defined by the balking coefficient $b = 0.8$.

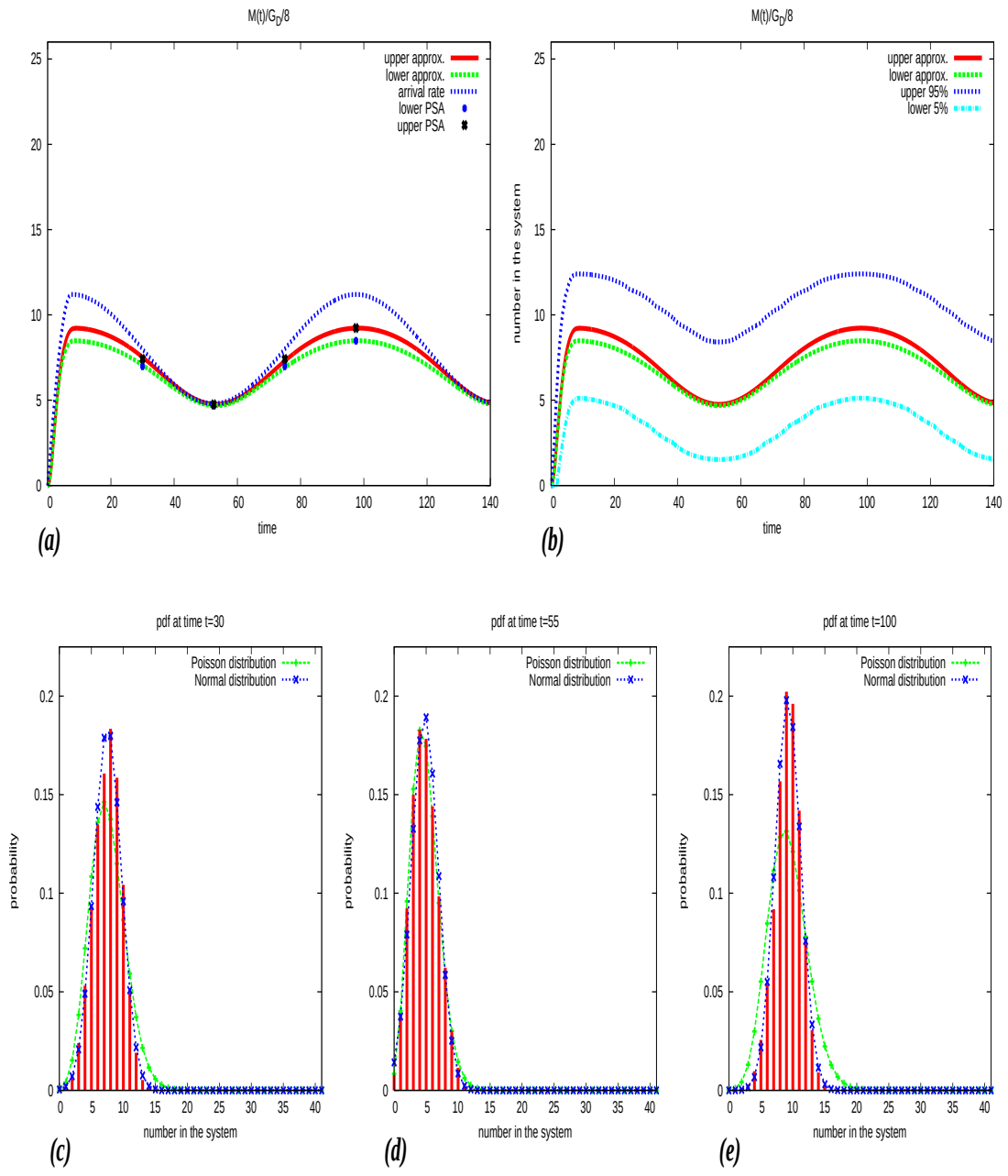


Figure 7.8: A quiet system that has a sinusoidal arrival rate with mean $\lambda = 8$, variation defined by the amplitude which is 40% of the mean, service rate $\mu = 1$, $s = 8$ servers, and strong balking defined by the balking coefficient $b = 0.8$.

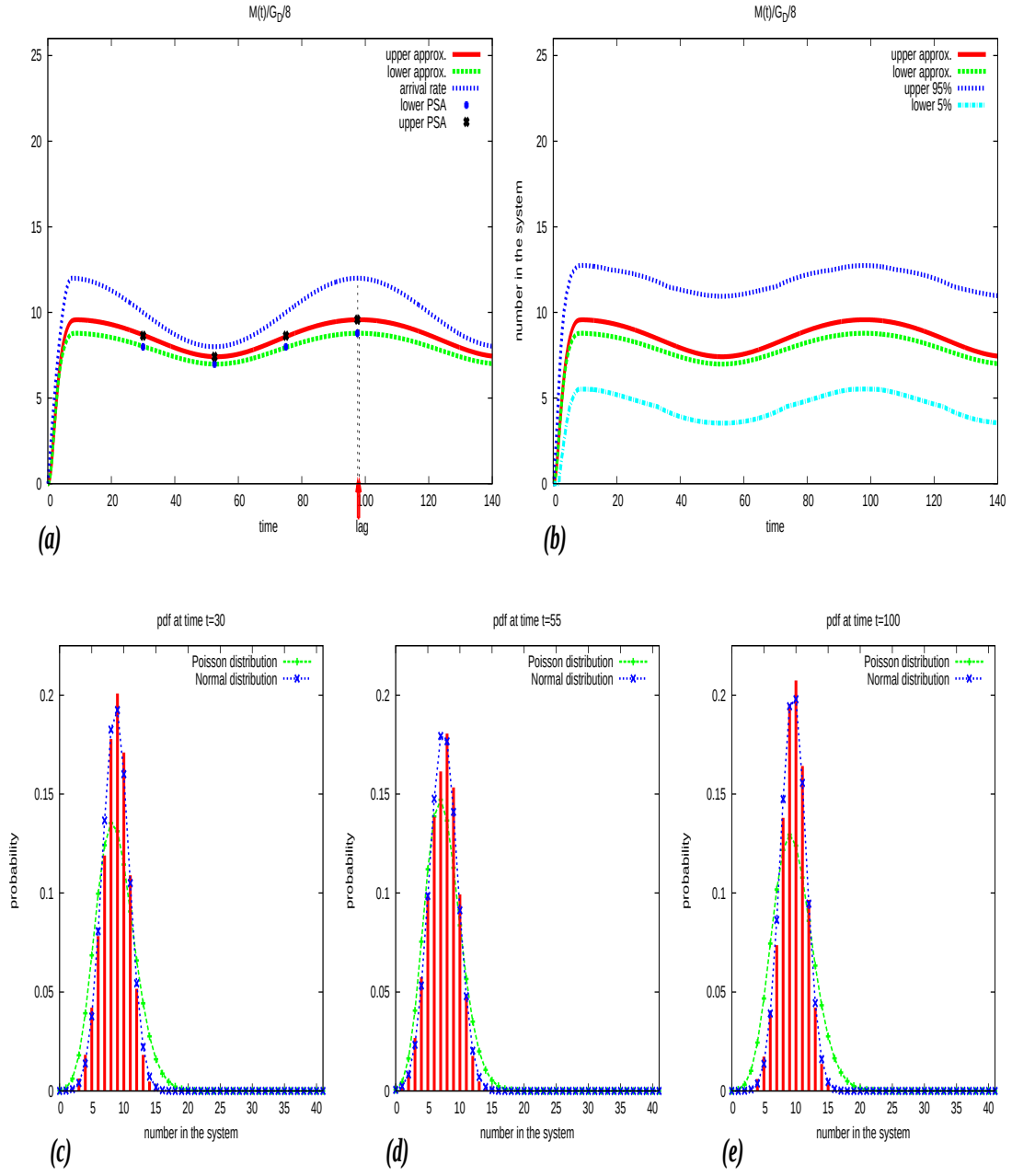


Figure 7.9: A medium busy system that has a sinusoidal arrival rate with mean $\lambda = 10$, mild variation defined by the amplitude which is 20% of the mean, service rate $\mu = 1$, $s = 8$ servers, and strong balking defined by the balking coefficient $b = 0.8$.

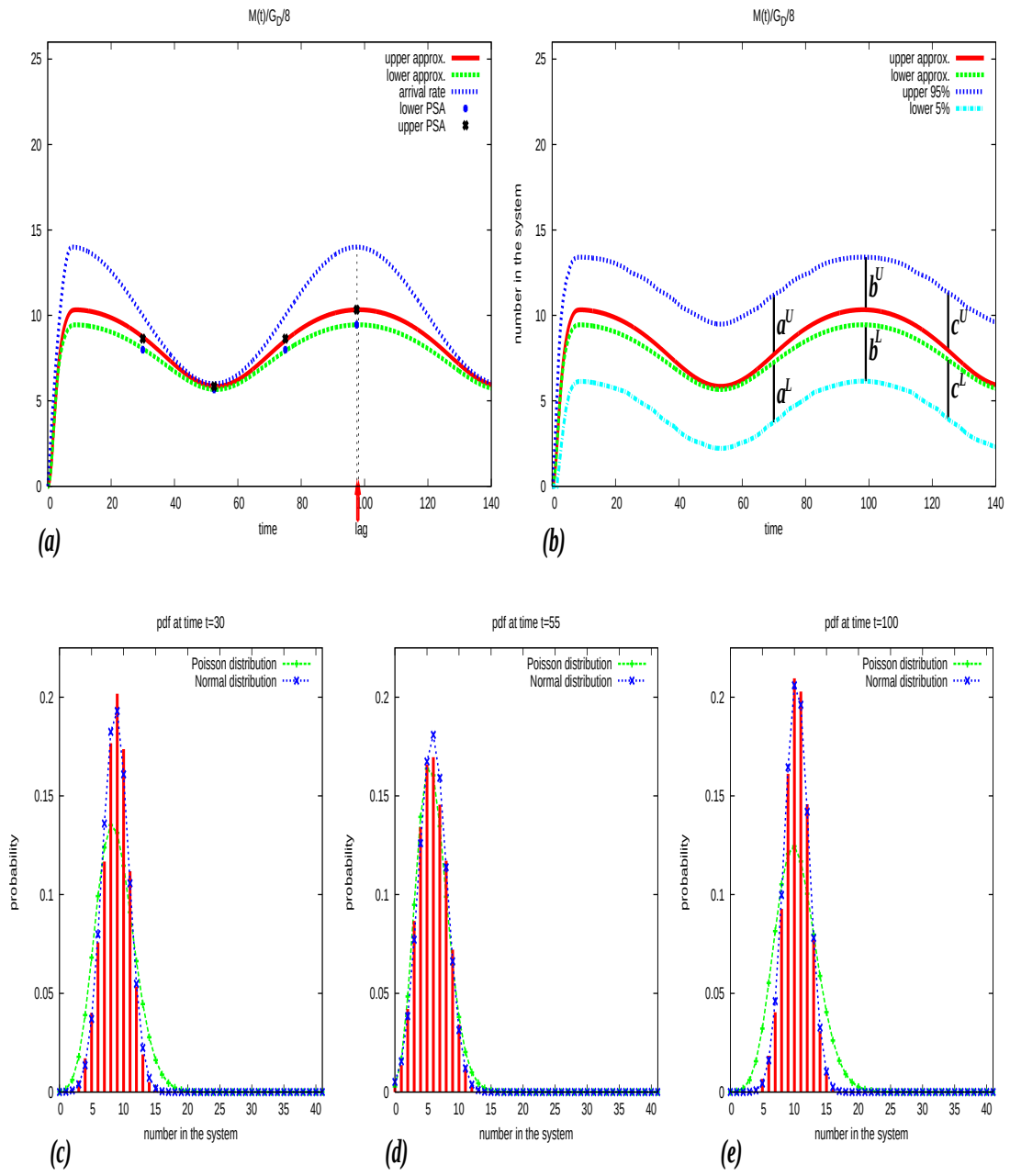


Figure 7.10: A medium busy system that has a sinusoidal arrival rate with mean $\lambda = 10$, variation defined by the amplitude which is 40% of the mean, service rate $\mu = 1$, $s = 8$ servers, and strong balking defined by the balking coefficient $b = 0.8$.

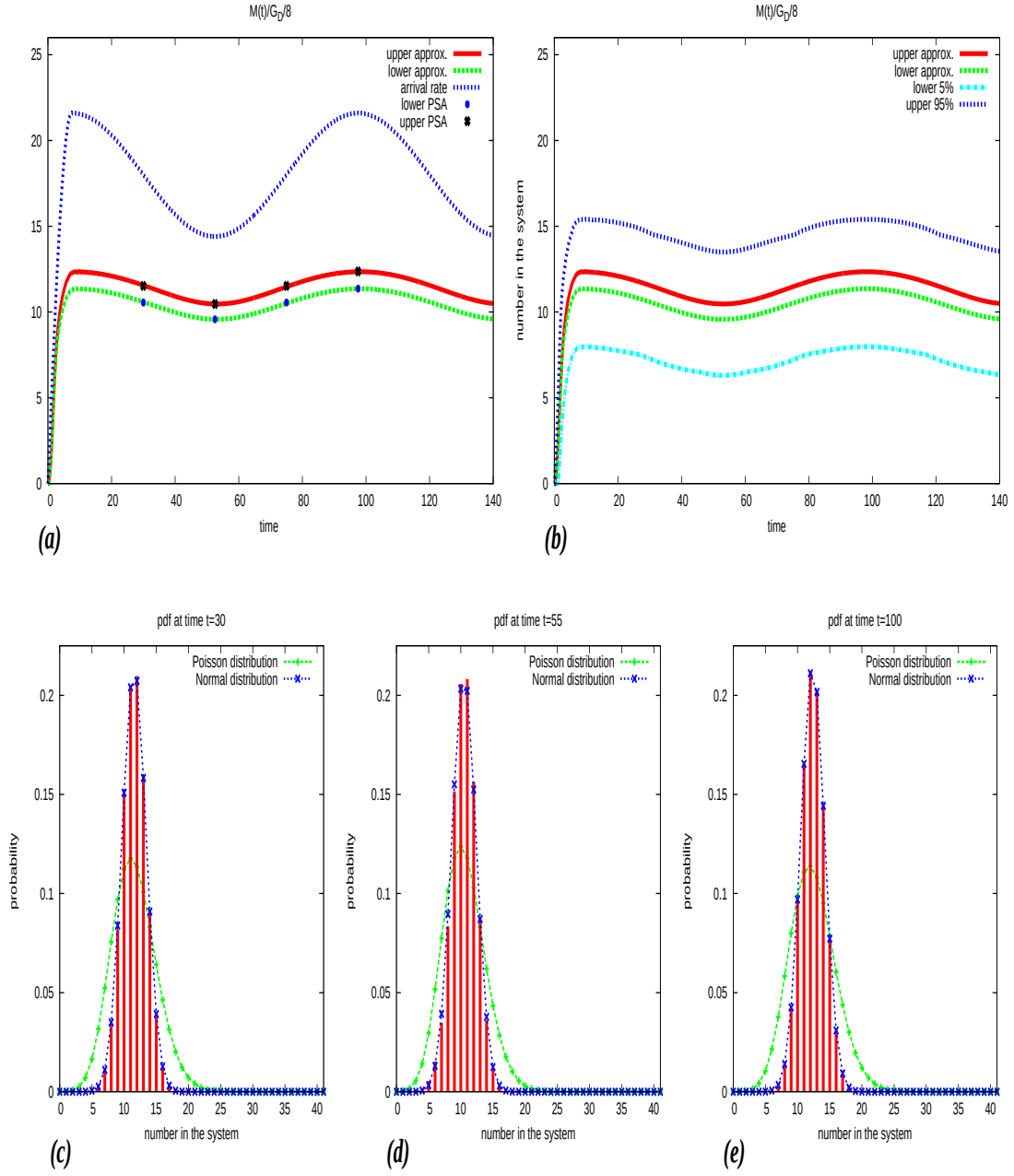


Figure 7.11: A very busy system that has a sinusoidal arrival rate with mean $\lambda = 18$, mild variation defined by the amplitude which is 20% of the mean, service rate $\mu = 1$, $s = 8$ servers, and strong balking defined by the balking coefficient $b = 0.8$.

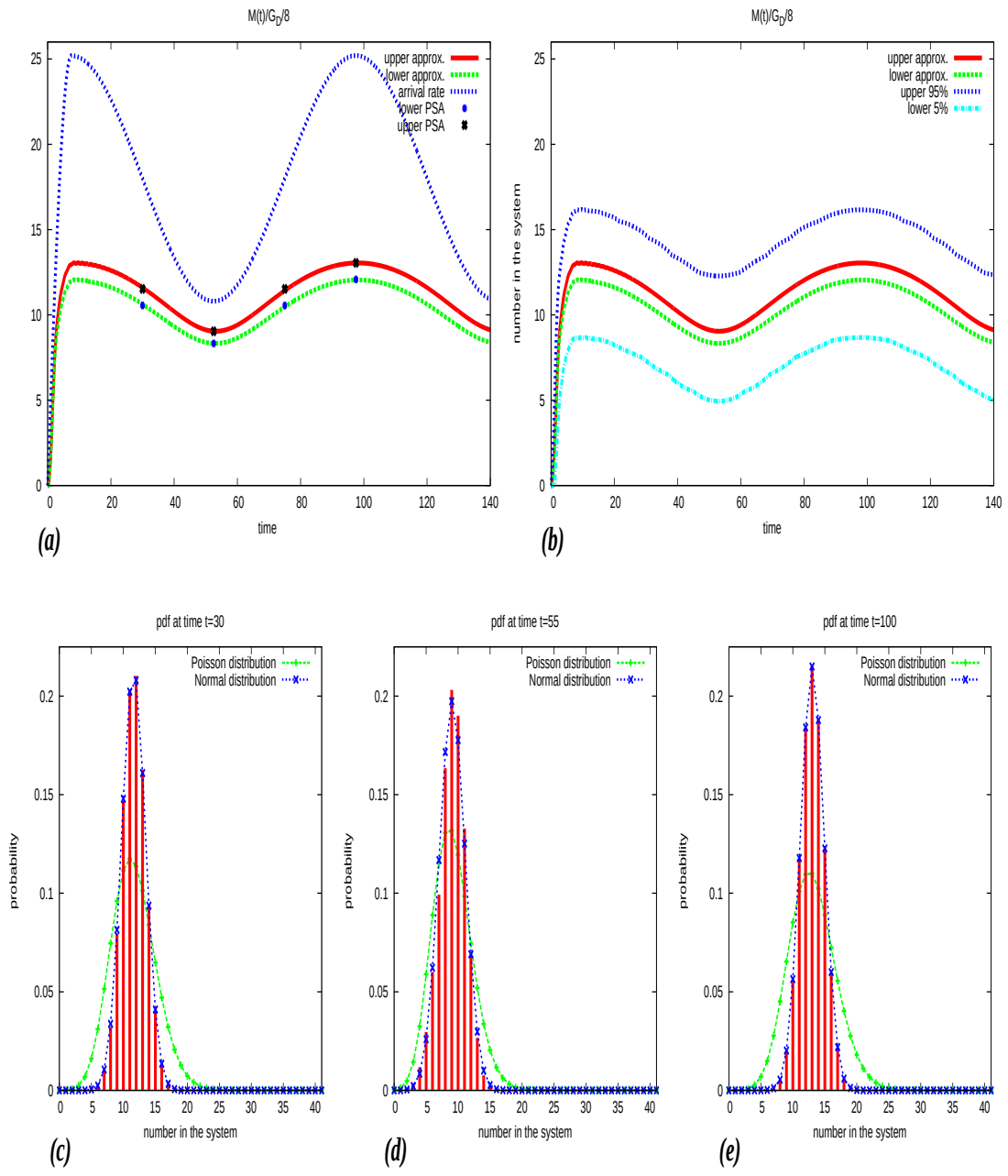


Figure 7.12: A very busy system that has a sinusoidal arrival rate with mean $\lambda = 18$, variation defined by the amplitude which is 40% of the mean, service rate $\mu = 1$, $s = 8$ servers, and strong balking defined by the balking coefficient $b = 0.8$.

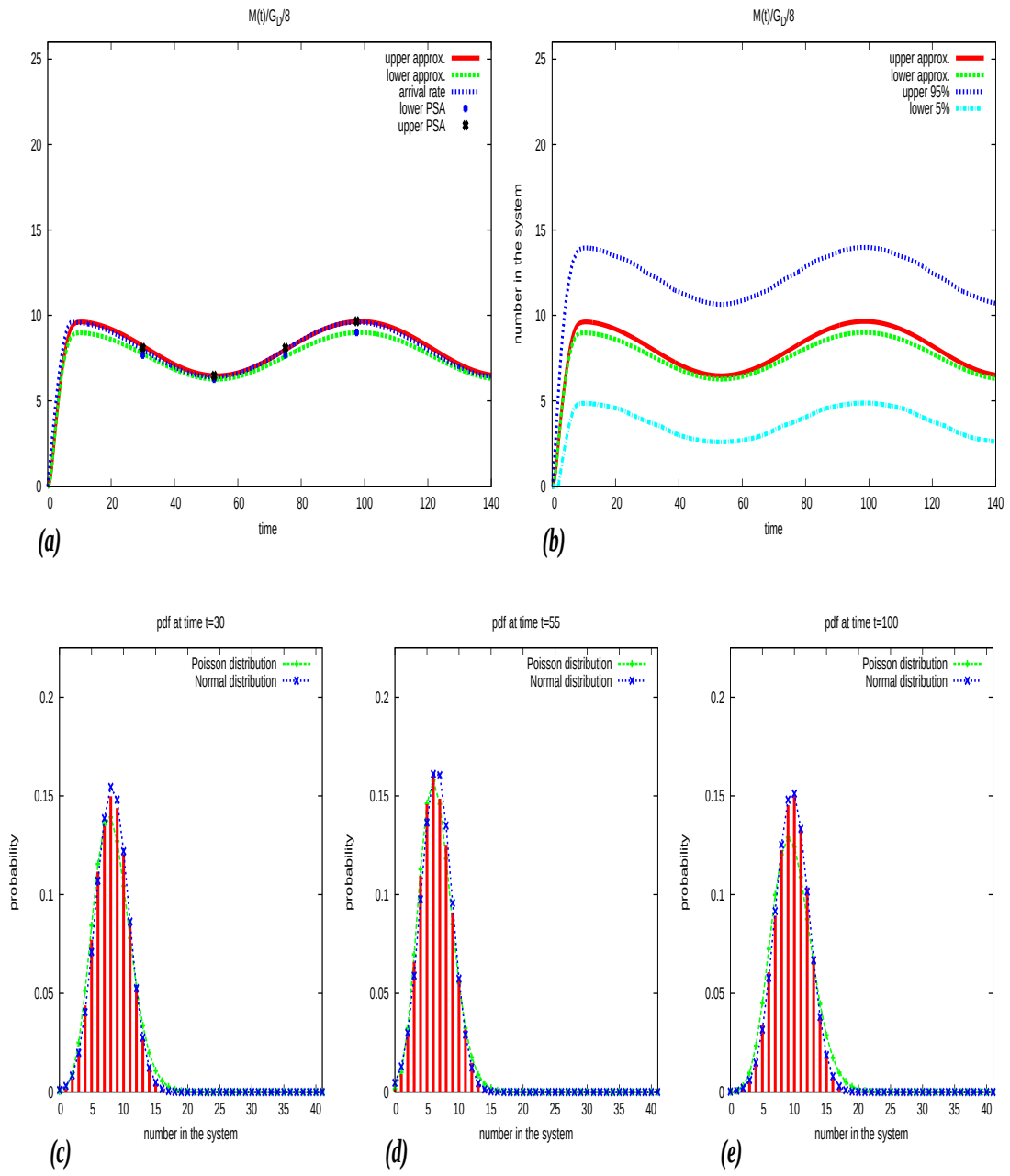


Figure 7.13: A quiet system that has a sinusoidal arrival rate with mean $\lambda = 8$, mild variation defined by the amplitude which is 20% of the mean, service rate $\mu = 1$, $s = 8$ servers, and balking defined by the balking coefficient $b = 0.9$.

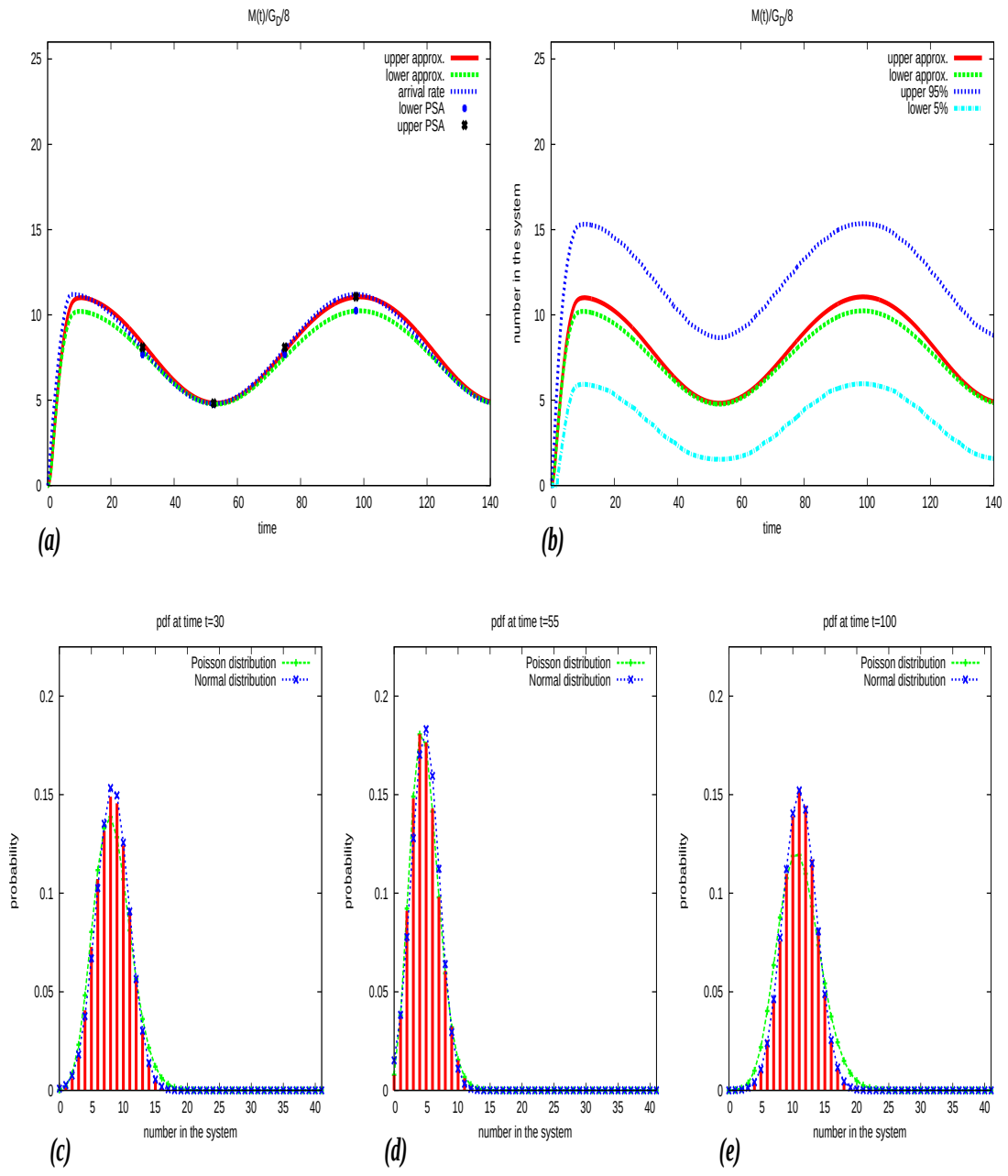


Figure 7.14: A quiet system that has a sinusoidal arrival rate with mean $\lambda = 8$, variation defined by the amplitude which is 40% of the mean, service rate $\mu = 1$, $s = 8$ servers, and balking defined by the balking coefficient $b = 0.9$.

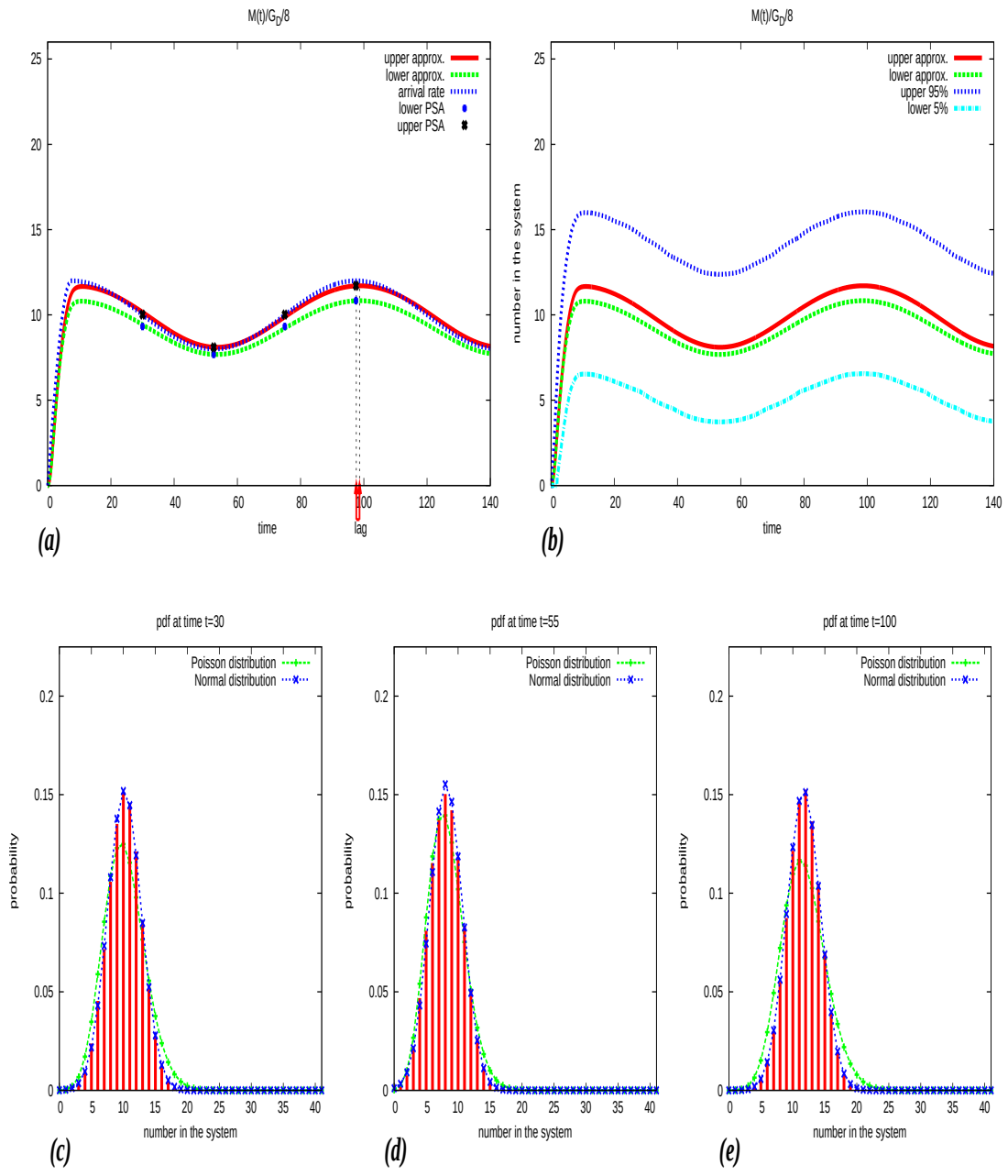


Figure 7.15: A medium busy system that has a sinusoidal arrival rate with mean $\lambda = 10$, mild variation defined by the amplitude which is 20% of the mean, service rate $\mu = 1$, $s = 8$ servers, and balking defined by the balking coefficient $b = 0.9$.

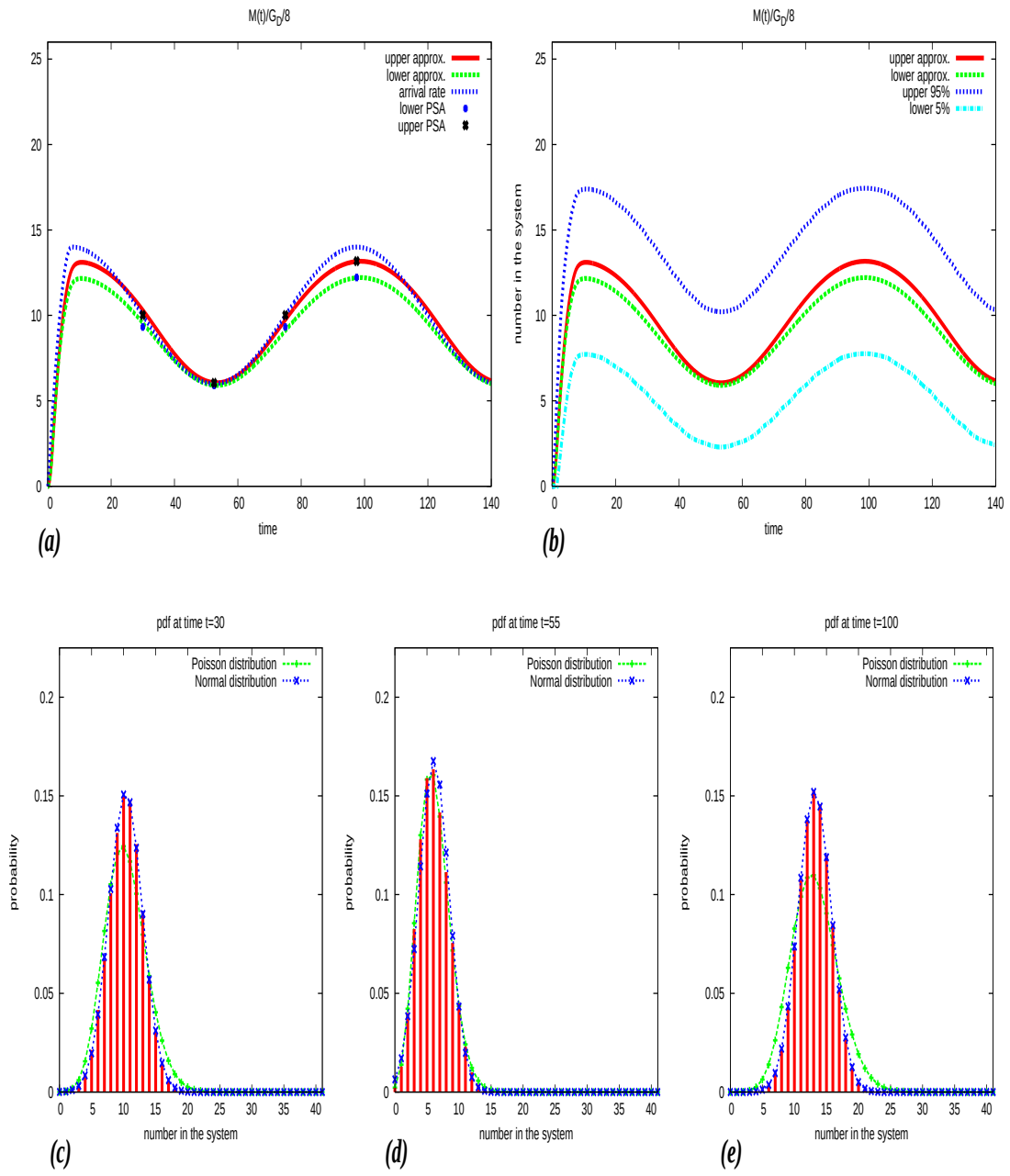


Figure 7.16: A medium busy system that has a sinusoidal arrival rate with mean $\lambda = 10$, variation defined by the amplitude which is 40% of the mean, service rate $\mu = 1$, $s = 8$ servers, and balking defined by the balking coefficient $b = 0.9$.

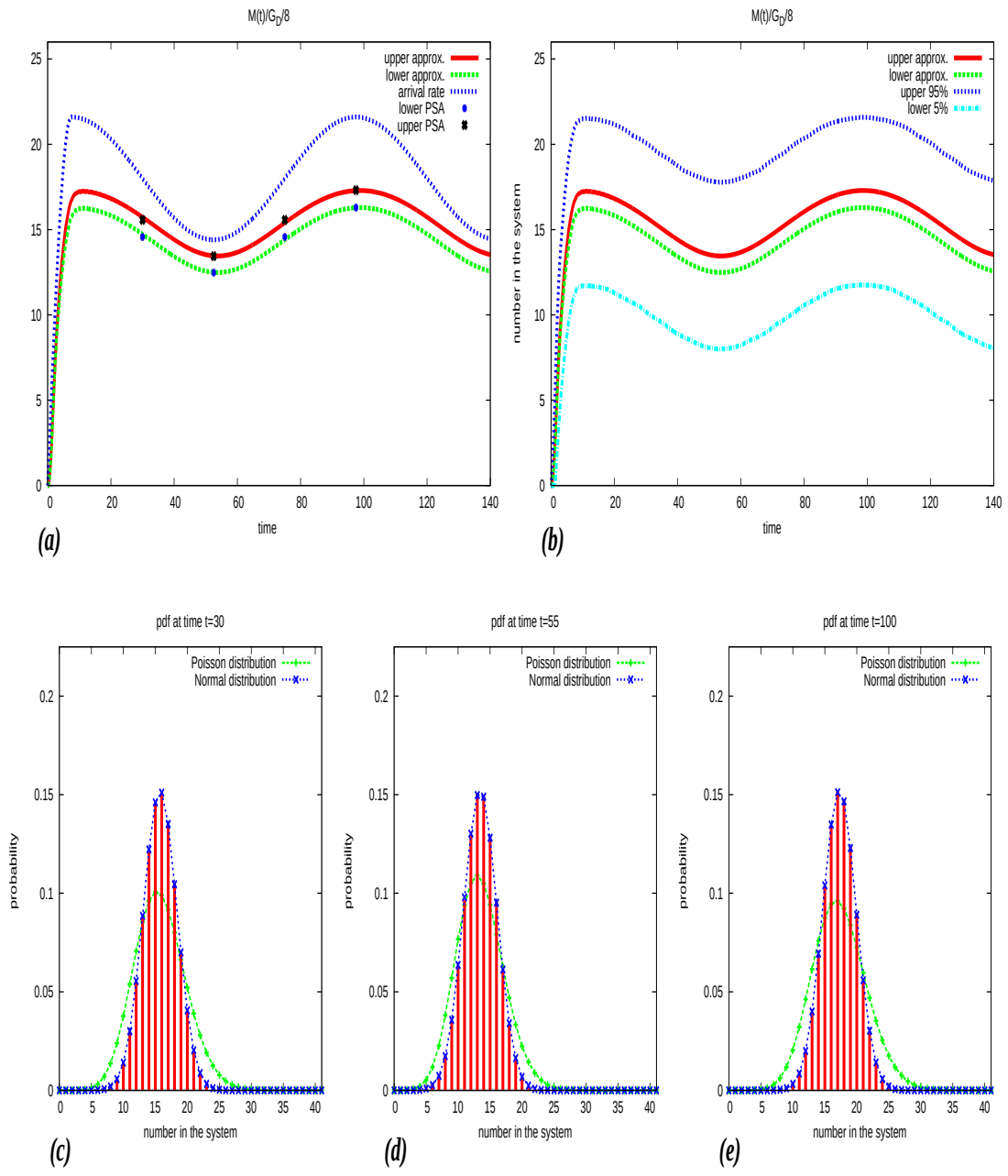


Figure 7.17: A very busy system that has a sinusoidal arrival rate with mean $\lambda = 18$, mild variation defined by the amplitude which is 20% of the mean, service rate $\mu = 1$, $s = 8$ servers, and balking defined by the balking coefficient $b = 0.9$.

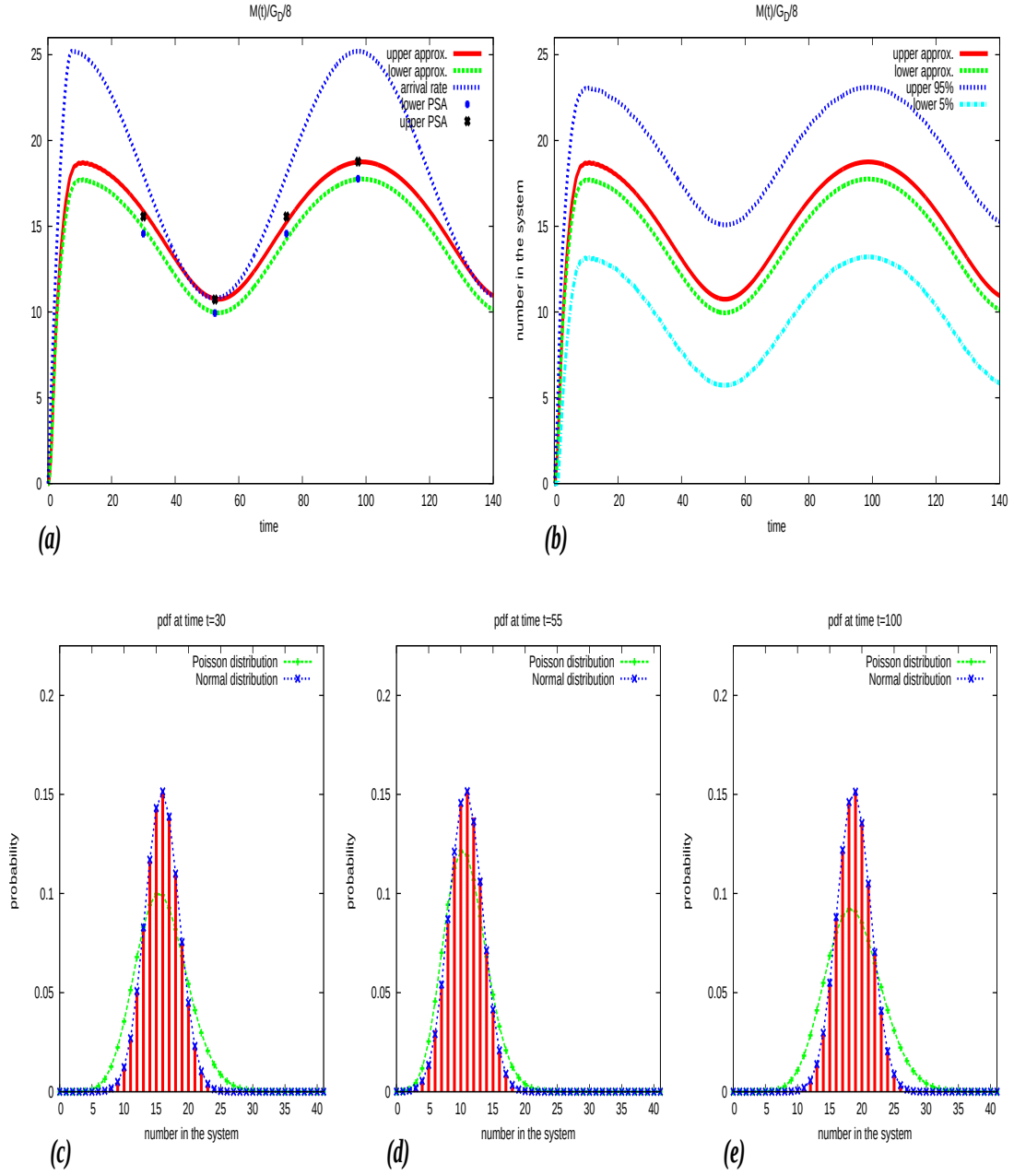


Figure 7.18: A very busy system that has a sinusoidal arrival rate with mean $\lambda = 18$, variation defined by the amplitude which is 40% of the mean, service rate $\mu = 1$, $s = 8$ servers, and balking defined by the balking coefficient $b = 0.9$.

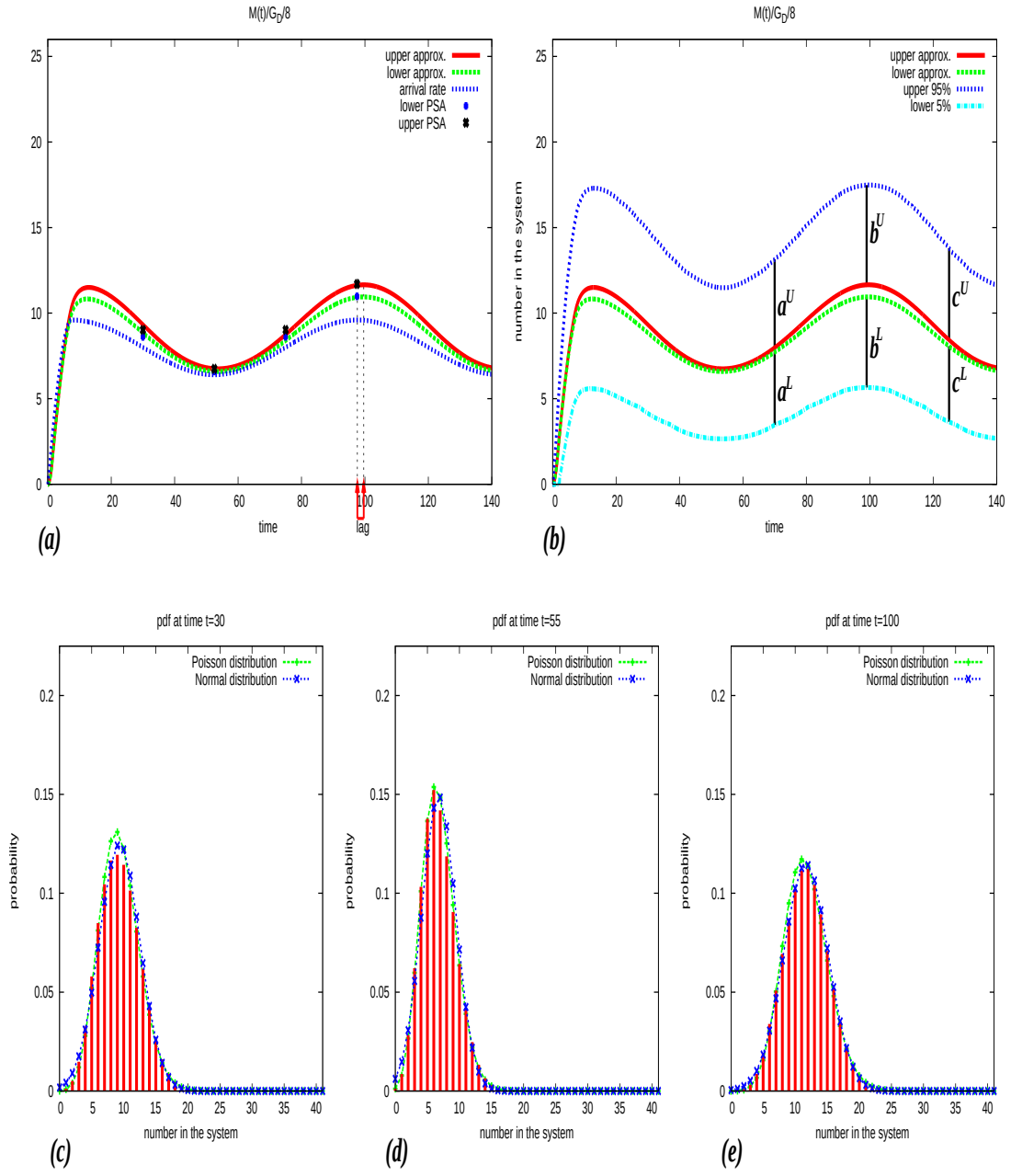


Figure 7.19: A quiet system that has a sinusoidal arrival rate with mean $\lambda = 8$, mild variation defined by the amplitude which is 20% of the mean, service rate $\mu = 1$, $s = 8$ servers, and weak balking defined by the balking coefficient $b = 0.95$.

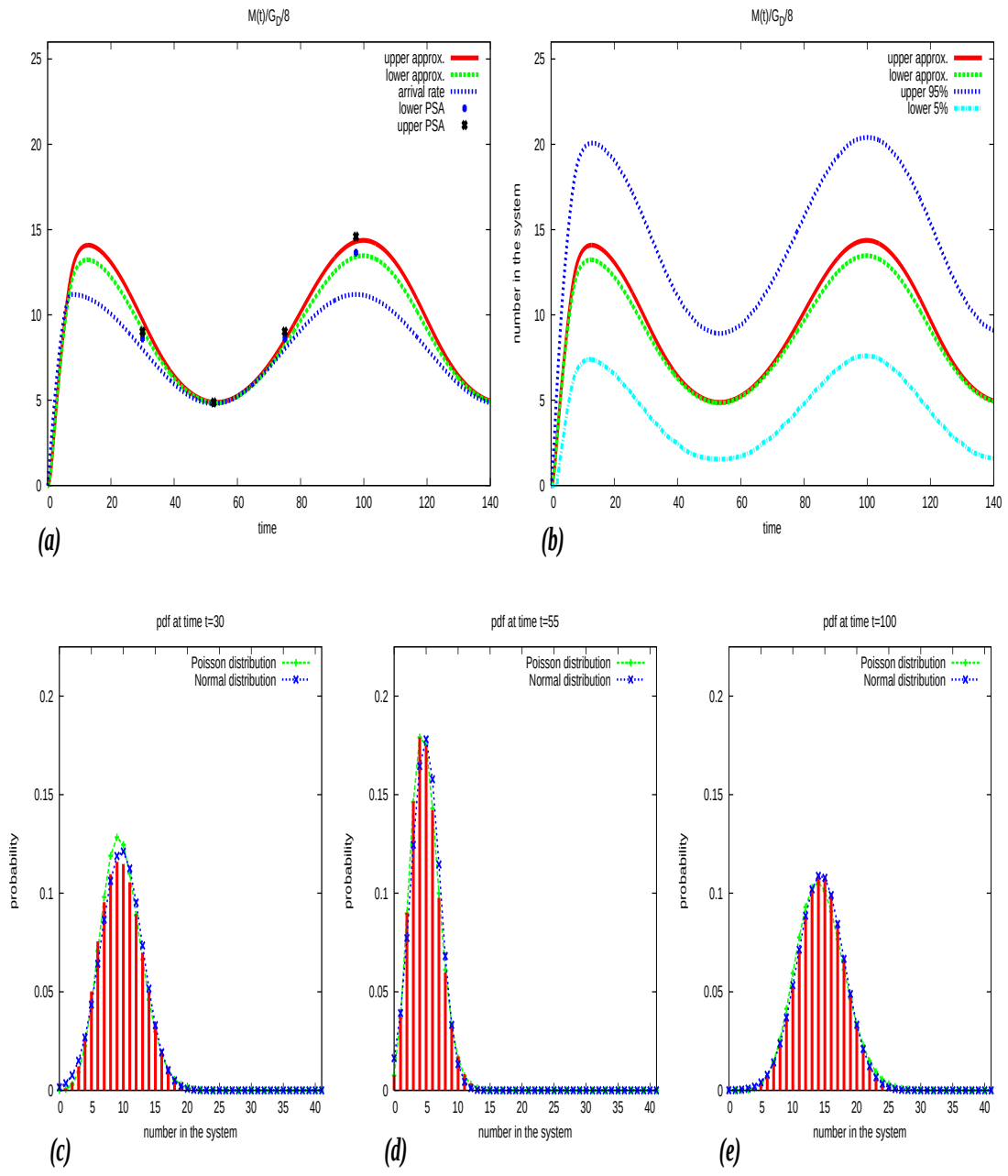


Figure 7.20: A quiet system that has a sinusoidal arrival rate with mean $\lambda = 8$, variation defined by the amplitude which is 40% of the mean, service rate $\mu = 1$, $s = 8$ servers, and weak balking defined by the balking coefficient $b = 0.95$.

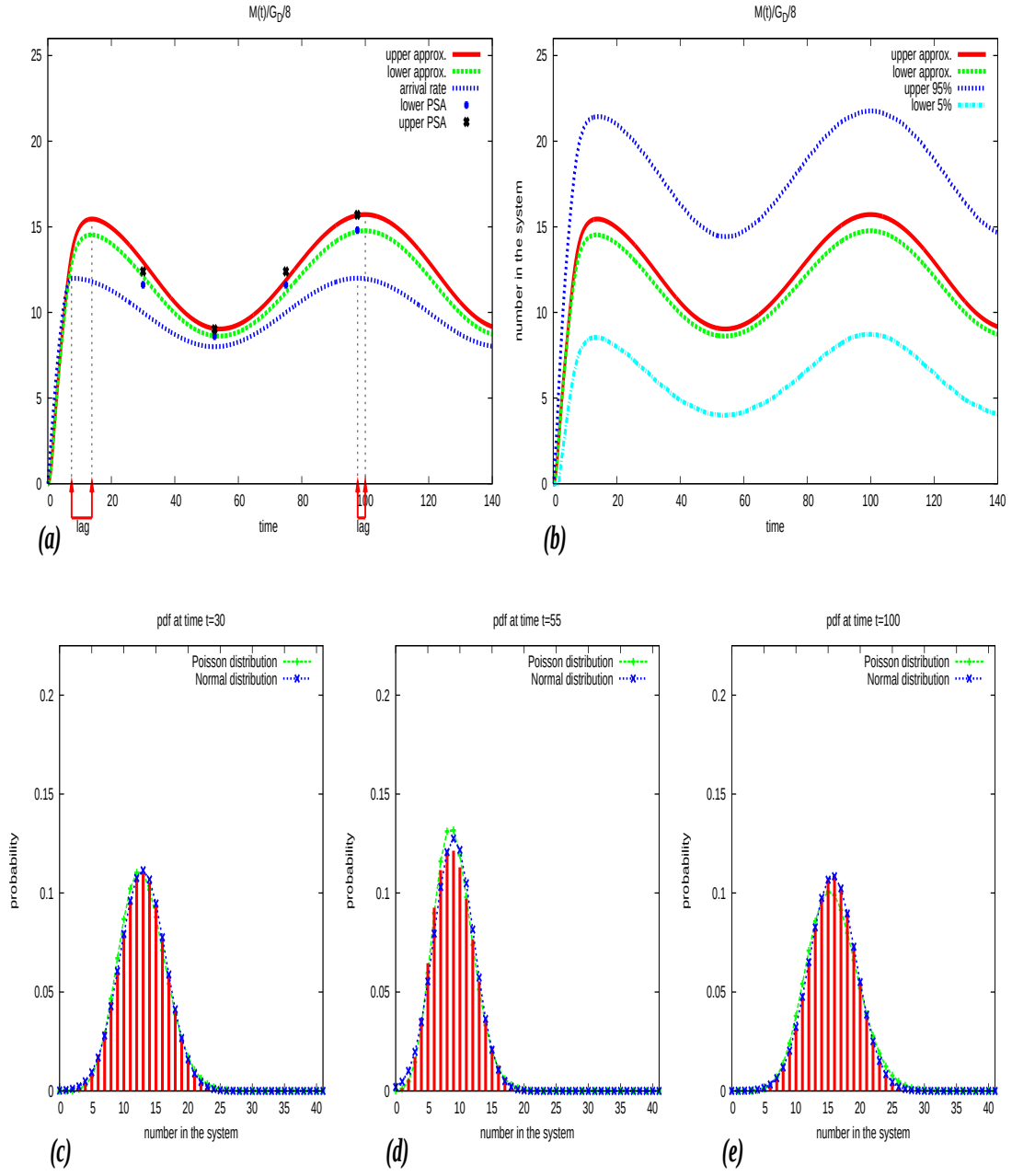


Figure 7.21: A medium busy system that has a sinusoidal arrival rate with mean $\lambda = 10$, mild variation defined by the amplitude which is 20% of the mean, service rate $\mu = 1$, $s = 8$ servers, and weak balking defined by the balking coefficient $b = 0.95$.

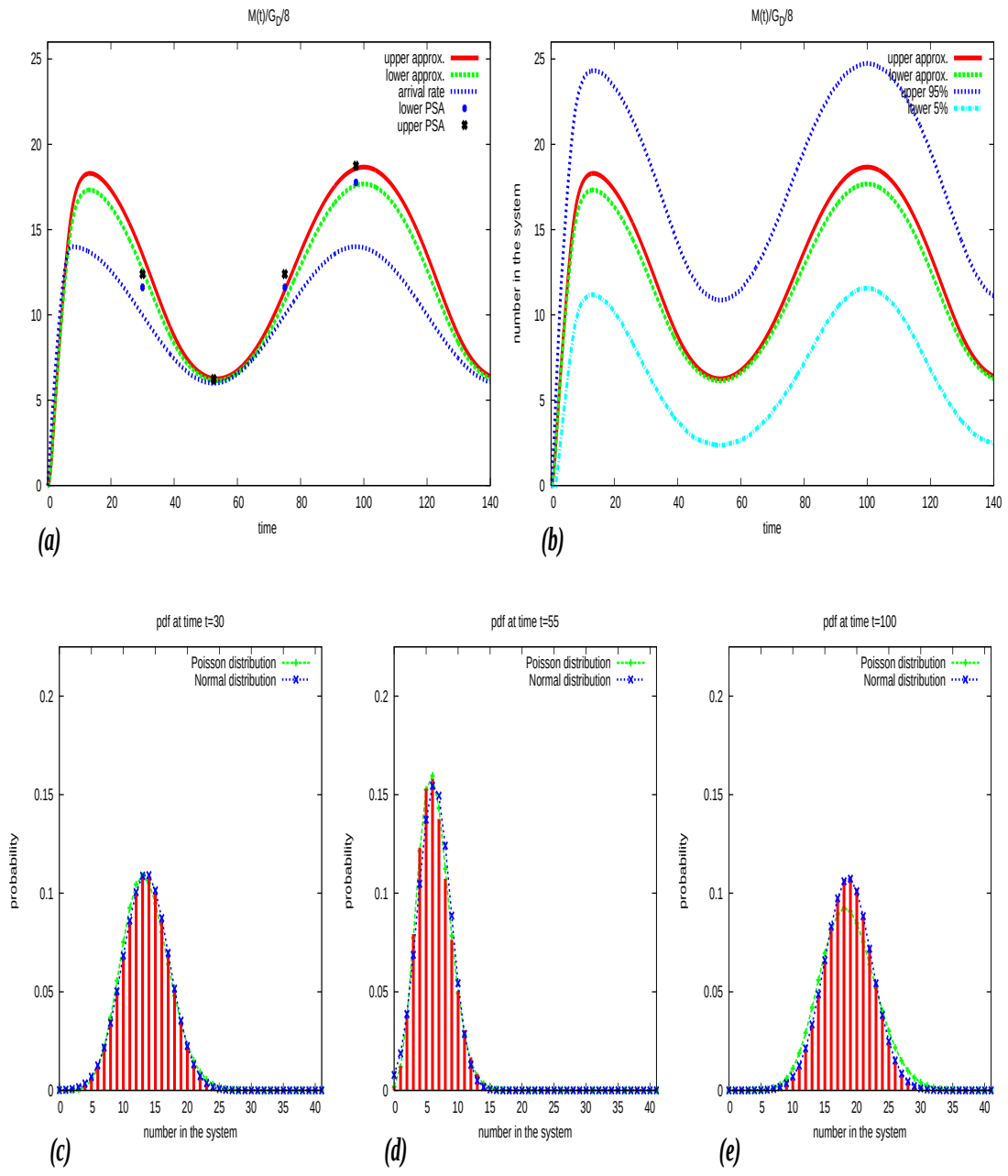


Figure 7.22: A medium busy system that has a sinusoidal arrival rate with mean $\lambda = 10$, variation defined by the amplitude which is 40% of the mean, service rate $\mu = 1$, $s = 8$ servers, and weak balking defined by the balking coefficient $b = 0.95$.

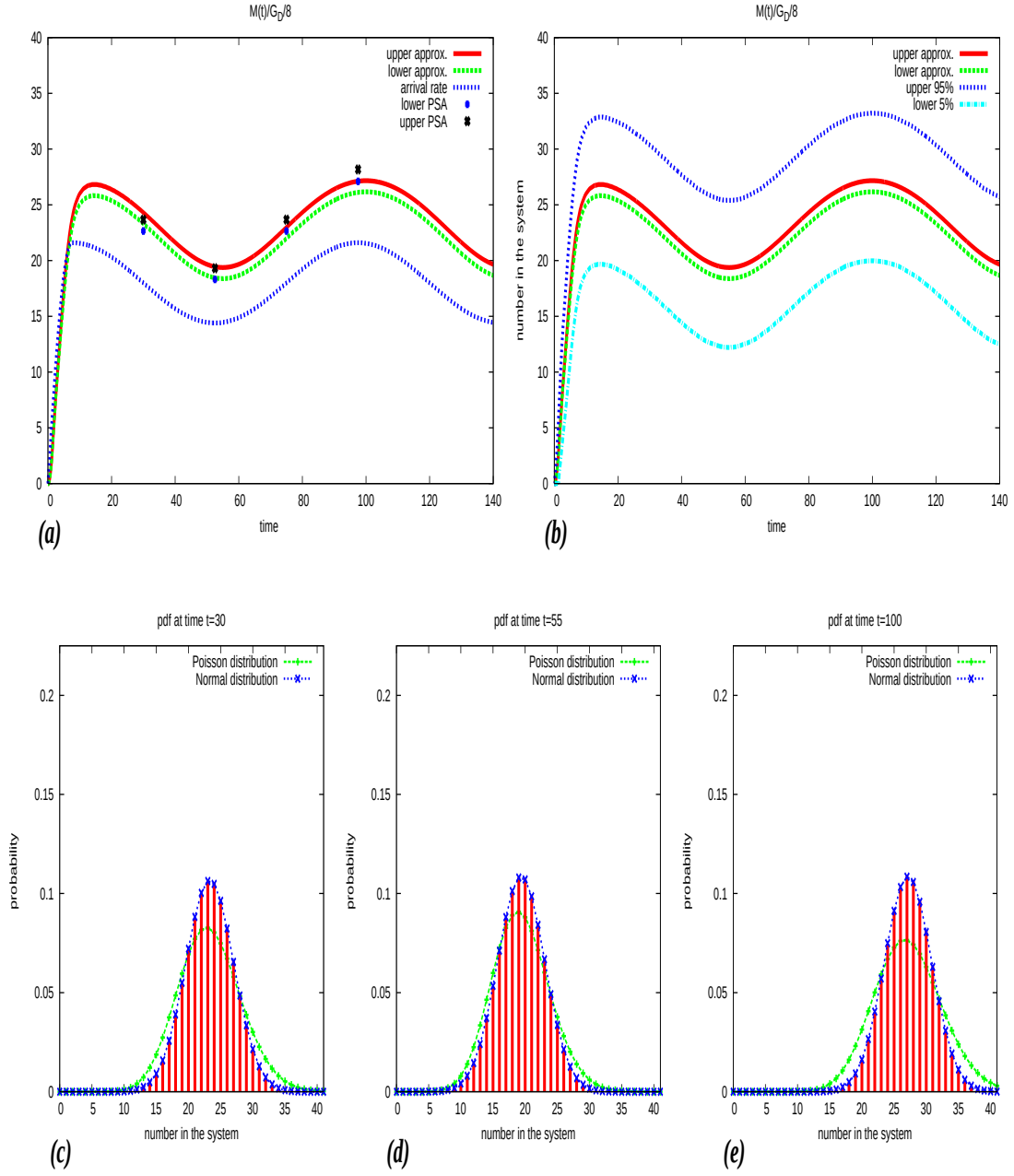


Figure 7.23: A very busy system that has a sinusoidal arrival rate with mean $\lambda = 18$, mild variation defined by the amplitude which is 20% of the mean, service rate $\mu = 1$, $s = 8$ servers, and weak balking defined by the balking coefficient $b = 0.95$.

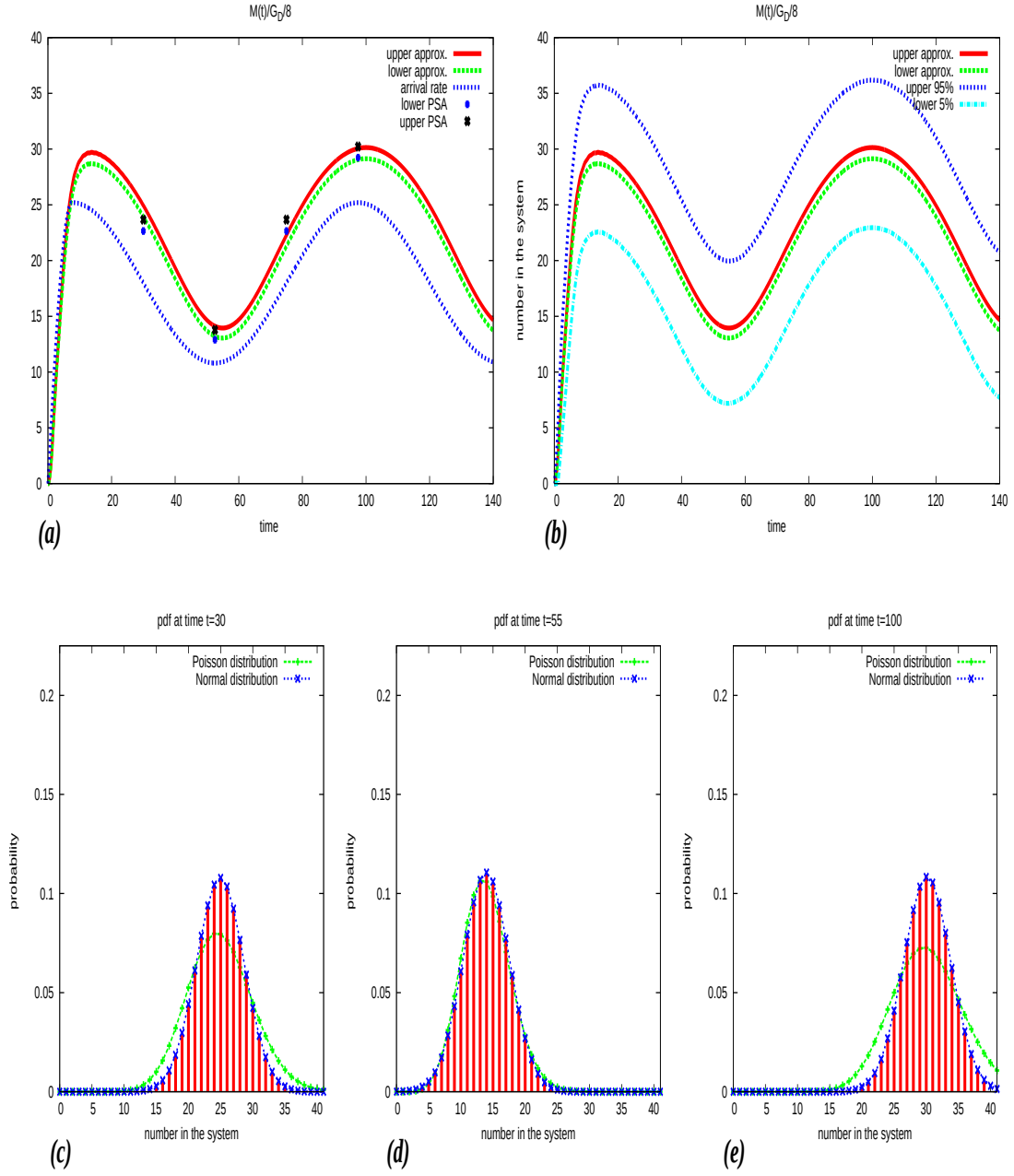


Figure 7.24: A very busy system that has a sinusoidal arrival rate with mean $\lambda = 18$, variation defined by the amplitude which is 40% of the mean, service rate $\mu = 1$, $s = 8$ servers, and weak balking defined by the balking coefficient $b = 0.95$.

Chapter 8

Conclusions and further research

8.1 Introduction

This research is concerned with developing queueing theory models for call centres and demonstrating the potential value of these models to provide understanding and insights to call centre queues. For this reason Chapter 1 describes call centres and their basic characteristics, explains how call centres can be modelled as queueing systems, and points out that methods currently used make restrictive assumptions and as a result provide very crude approximations.

A review of the relevant literature for important call centre characteristics, which include multi-server systems with time-dependent and state-dependent arrival rates and general service time distributions, was undertaken in Chapter 2. From this review it was concluded that, based on queueing theory, discrete-time modelling is the most appropriate analytic technique to model basic call centres. This is because this approach has been used previously to provide very accurate approximations for multi-server queueing systems with time-dependent arrival rates and general service time distributions. Moreover the explicit description of DTM in Chapter 3 indicates that this method has the potential to be developed to include the other main characteristic, of call centres, i.e. state-dependent arrival rates.

Having set the background for this research we proceed in Chapters 4-7 to develop and apply DTM for systems with state-dependent balking. This chapter summarises

the main conclusions from this work and indicates issues for which further research would be useful.

Our conclusions can be grouped in two main categories. The first category, described in Section 8.2, are conclusions derived from the procedure of modelling discrete-time systems with state-dependent arrivals. This category is more addressed to queueing theorists. The second category, described in Section 8.3, includes insights related to the performance of call centres. This category could be useful to practitioners interested in call centre queue management problems.

After reporting these conclusions we discuss in Section 8.4 issues subject to further research.

8.2 Conclusions about modelling state-dependent discrete-time systems

Chapter 4 provides the theoretical contribution to incorporate balking into the DTM approach. In order to introduce balking we need to introduce state-dependent arrivals to the DTM algorithm. Our first step is to formulate state-dependent arrivals as a pure birth process, i.e. departures are ignored. The convolution of two, then three and so on, negative exponential distributions with different arrival rates was calculated and was given different forms so that a pattern could be identified. This has led to a recursive formula for calculating the state-dependent arrival probabilities, which was proved by induction in Section 4.3 (Theorem 4.1). An alternative formula for calculating these arrival probabilities is also proved in Section 4.4 (Theorem 4.2). It is noted that both these theorems require all arrival rates to be distinct, whereas in balking systems this will not necessarily be the case. Hence in Section 4.5 the formulae are extended to state-dependent arrival probabilities with a recurrent arrival rate.

In systems without balking, departures and arrivals are independent events. However since departures reduce the system's state, in systems with balking they will affect the arrival rates. During a slot, arrivals and departures can occur at any

point. However, DTM tracks the system under consideration only at epochs, and thus residual service times are rounded up to the next integer in order to introduce the Markov chain. For this reason no information is available about the exact time at which departures will occur during the slot. This makes impossible to introduce state-dependent balking in the DTM without making assumptions about the time at which departures will occur.

Therefore we have introduced two approximations: an 'early departure' one and a 'late departure' one. In the 'early departure' approximation departures are assumed to occur before arrivals (i.e. departures occur at the beginning of the slot) and as a result arrivals 'see' a less congested system than the actual one. Arrivals seeing a less congested system face weaker balking and thus this approximation seems likely to overestimate the actual congestion. In the 'late departure' approximation departures are assumed to occur after arrivals (i.e. departures occur at the end of the slot) and as a result arrivals 'see' a more congested system than the actual one. Arrivals seeing a more congested system face stronger balking and thus this approximation seems likely to underestimate the actual congestion. The reason for defining the approximations as above is because intuitively they seem to bound the actual system's congestion. We could for example introduce an alternative approximation that assumes half of the departures occur at the beginning of the slot and the other half at the end. Intuitively this would seem likely to lead to a more accurate approximation than the early and late departure approximations. However, bounds of the actual solution for which the accuracy can be controlled are more useful than approximations of unknown precision.

In Chapter 5 we undertake a theoretical investigation of the bounding behaviour of the two approximations. To make a general statement on whether the two approximations bound the actual solution we need to consider an $M(t,n)/G_D/s$ system. However the formulation for this system proved to be very complicated and so we limited our investigation to an $M(t,n)/D/s$ system, i.e. a system with a deterministic service time distribution. Even for this system the theoretical proof proved to be a challenging task.

The general framework of this task is the performance comparison between two

time-dependent queueing systems in discrete time. This issue is not addressed in the relevant literature. The proofs presented in Chapter 5 are all based on induction, i.e. they assume that the desired relationship for the selected performance measure is valid at an arbitrary epoch and use induction to prove that it is also valid for the next epoch. One crucial step in the proofs was the selection of the performance measure. For example, in our preliminary attempts we used the mean number in the system as performance measure since it was thought that this would involve easier calculations. Indeed this simplified some parts of the formulation, however it was impossible to get the desired result. In the end the required result was obtained by using the cumulative distribution of the number in the system, having shown that if the cumulative probabilities are ordered, their mean values are also ordered.

The proofs start by writing the Chapman-Kolmogorov equations for two successive epochs. It is common after this stage in much queueing theory to use transforms to ease the calculations. Indeed difference equations reduce to algebraic equations by applying for example the z-transform, however getting the inverse transform is often very difficult. Thus most queueing theory results are limited to steady state. Our analysis takes place in the time domain.

We have shown in Section 5.6 (Theorem 5.2) that an $M(t,n)/D/s$ discrete-time system with 'early departures' has at each epoch a smaller cumulative distribution of the number in the system than the actual system, and as a result is always more congested than the actual one. We have also shown in Section 5.7 (Theorem 5.3) that an $M(t,n)/D/s$ discrete-time system with 'late departures' has at each epoch a larger cumulative distribution of the number in the system than the actual system, and as a result is always less congested than the actual one. Thus for an $M(t,n)/D/s$ system, at each epoch, the approximations introduced in Chapter 4 provide bounds of the actual congestion levels. The proofs in this chapter demanded extensive algebraic calculations and caution in the allowable increments/decrements each time an inequality is formed. From this viewpoint, the comparison between the exact and the lower approximation was more difficult to accomplish, as it demanded more subtle formation of inequalities. In conclusion, both the development of the proofs

and the successful results of this chapter contribute to the time-dependent analysis of state-dependent systems in discrete time.

Having seen that even for $M(t,n)/D/s$ systems results were arduously established, the complexity of $M(t,n)/G_D/s$ systems was thought to be prohibitive, in the context of this research. For this reason, in order to provide a broader investigation of the bounding behaviour of the two approximations, a simulation model was developed and employed. Chapter 6 describes and validates this simulation model. It also compares results produced by this model with results produced by the two approximations for various systems. We conclude that the two approximations always bound the actual solution and that the difference between the bounds can be controlled by controlling the duration between two successive epochs. As a result the approximations provide bounds of controllable accuracy.

Chapter 7 studies the performance of systems with state-dependent balking using the new approximate models. The form of balking for these tests is assumed to be geometrical. One of the issues addressed is the effect of the service time distribution. It is concluded, based on empirical results, that in systems with balking the mean number in the system is in fact insensitive to higher than the first moment of the service time distribution. This implies that, at any point of time, the mean number in the system in an $M(t,n)/G_D/s$ system will be very similar to the mean number in the system in an $M(t,n)/D/s$ system. As a result the theory developed in Chapter 5 will also hold for $M(t,n)/G_D/s$ systems, i.e. the approximations provide bounds for the general case.

8.3 Conclusions related to the performance of systems with state-dependent balking and call centres

Having developed DTM to model basic characteristics of call centres, Chapter 7 applies the DTM approximations to study the performance of systems with state-

dependent balking. Our empirical results indicate that:

- In call centres with balking, higher than the first moments of the service time distribution are insignificant for the mean number in the system. In other words systems with balking are insensitive to the service time distribution. As a result when modelling systems with balking and we are interest in the mean number in the system, the distribution which is most convenient for this particular modelling can be used. For example if the modelling method is DTM, a deterministic service time distribution, which requires only one point for its description, might be used, decreasing significantly memory and speed demands.

- It is no surprise that balking reduces the mean number in the system. Furthermore, balking affects the standard deviation of the number in the system. The stronger the balking the smaller the standard deviation of the number in the system. As a result in call centres with balking the mean number in the system becomes a strong indication about the congestion levels.

- The distribution of the number in the system fits closely to a normal distribution for any congestion level and strength of balking. The distribution of the number in the system is useful to indicate the range of delays to expect.

- The degree of balking affects lags between arrival peaks and peaks in congestion. Moreover, the stronger the balking the smaller the delay between the traffic intensity peak and the congestion peak. This indicates that in call centres with balking arrival rate peaks will be converted into congestion peaks faster than in call centres without balking.

- The PSA generally provides a good approximation for systems with balking. The stronger the balking the better PSA performs. This is mainly a consequence of the shorter lags mentioned above, and the fact that balking guarantees that steady-state results are meaningful.

Finally, motivated by the fact that PSA performs well for systems with balking, in Section 7.5 we provide a simple formula to calculate the approximate mean number

in the system for busy systems. This formula is straightforward to apply, and only requires the number of servers, the mean service rate, the balking coefficient and the arrival rate function. In this way practitioners are equipped with a simple formula that can be used to obtain reasonably accurate time-dependent behaviour of systems with balking, without the need to resort to numerical methods or simulation.

Overall we advise call centre modellers to use the simple PSA formula to get quick estimations of congestion levels and to identify likely problematic times of the day. In cases in which a more accurate representation of the system is needed we suggest use of the two DTM approximations to get bounds of the actual congestion levels. An arbitrary step size, for example equal to one unit of time, could be used in the an initial run of the DTM programmes. If the accuracy is not satisfactory and 'k' times better accuracy is needed (i.e. the gap between the approximations needs to be reduced 'k' times) we recommend the user to re-run the models with new step size 'k' times smaller than the initial one. If the call centre under consideration has more sophisticated characteristics that affect its performance, for example priority calls, or multi-skilled agents, simulation may be required.

8.4 Further research

There is a number of possible directions in which this research could be extended in future work.

A possible way involves the use of geometric distributions. Our empirical results in Chapter 7 have indicated that systems with balking are insensitive to higher than the first moment of the service time distribution. This is an important finding that has serious implications in modelling these systems, as mentioned in the previous sections. This finding is particularly important for the DTM applicability since use of a geometric service time distribution reduces considerably the memory and runtime demands. The geometric distribution is the discrete-time version of the negative exponential one, and as such is memoryless. Hence the state space only needs to record the number in the system and not the remaining service stages. For these

reasons it would be interesting to pursue a further and more systematic investigation on this issue. Moreover, as we have already mentioned in Chapter 3 current studies of discrete-time $Geo/Geo/s$ systems are limited to steady-state results. Our work could be extended to study the time-dependent behaviour of $Geo/Geo/s$ systems in the light of the formulation that was used in Chapter 5.

Another issue that could be investigated is extending DTM using parallel programming. Though this research was not concerned with the borders that limit the feasibility of DTM we are aware and concern of its limitations. The number of possible states in which the system can be found increases drastically with the number of points that are used to describe the discrete time distribution and the number of servers. To go beyond these limits we would need to use parallel programming. Running DTM in parallel processors is a challenging task, as the aim is to reduce both memory and runtime. A brief investigation on this issue showed that the processes in the DTM algorithm can be rearranged to take a parallel formation. In order to implement this parallel task the message passing interface (MPI) should be considered as it is positively reviewed and its library is supported by C++ in which the programmes are currently written.

Other areas of application of this work could also be sought. As we have seen in Chapter 3 discrete-time queueing models have recently received a lot of attention in computer communications. It would be interesting to investigate this sector for possible areas of application of theory similar to the one developed in Chapter 5. This theory could be modified for example for single-server systems with a general discrete distribution, or for multi-server systems with a geometrical instead of a deterministic service time distribution. State-dependent routing seems to have direct connection with the systems we have studied. This case could be of special interest as the balking function is prespecified and known by the routing policy. In state-dependent routing if the state of a node is high, arrivals are directed in other nodes while if it is low, arrivals are brought from other nodes, i.e. based on a threshold the arrival rate function increases when the queue is 'low' and decreases when the queue is 'high'. In this case we would adjust our approximations so that arrivals in the 'upper'

approximation will see departures to occur at the beginning of the interval when queue is above the threshold and at the end of the interval when the queue is below the threshold, and vice versa for the 'lower' approximation. Our theory could be used to evaluate the performance of nodes with different routing policies.

Finally, the work in Chapter 7 could be extended for different forms of balking in order to provide more robust results.

8.5 Final Conclusions

This work aimed to develop queueing theory models to model basic call centre characteristics. These included time-dependent arrival rates in multi-server systems with general service time distributions and state-dependent balking. The discrete-time approach was selected as the most appropriate technique. In order to include all the characteristics mentioned above, this research successfully extended the discrete-time approach to model state-dependent balking. This was done by introducing two approximations that bound the actual solution within controllable accuracy. Theoretical and empirical work was undertaken successfully to support this bounding behaviour. We have proved that these two approximations bound the actual solution for $M(t,n)/D/s$ systems. A simulation model was developed and used to show this bounding behaviour for more general systems.

Systems with state-dependent balking have then been studied using the approximations and some interesting findings have been obtained, providing important insights into the behaviour of call centre queues, and providing practical ways for modelling them.

Finally, there is clearly scope for developing the DTM approach further, and applying it in other related problem areas.

Appendix A

We give here some simple proofs of results that we have used in chapter 4 and in chapter 5.

It is well known, that the pdf of a random variable, that consists of the summation of two other random variables, is given by the convolution of their pdfs (see for example [72]), thus for continuous random variables we have:

$$\begin{aligned} Z = X + Y \quad \Rightarrow \quad & f_Z(t) = \int_{-\infty}^{\infty} f_X(t-y) f_Y(y) dy \Rightarrow \\ & \text{(for nonegative random variables)} \\ & f_Z(t) = \int_0^t f_X(t-y) f_Y(y) dy \end{aligned} \tag{A.1}$$

In the case where X and Y have exponential pdfs e.g. $f_X(x) = \lambda_X e^{-\lambda_X x}$, $f_Y(y) = \lambda_Y e^{-\lambda_Y y}$ by applying (A.1) we have :

$$\begin{aligned} f_Z(t) = f_X(x) \star f_Y(y) &= \lambda_X e^{-\lambda_X x} \star \lambda_Y e^{-\lambda_Y y} = \int_0^t \lambda_X e^{-\lambda_X (t-y)} \lambda_Y e^{-\lambda_Y y} dy \\ &= \lambda_X \lambda_Y e^{-\lambda_X t} \int_0^t e^{(\lambda_X - \lambda_Y) y} dy = \frac{\lambda_X \lambda_Y e^{-\lambda_X t}}{\lambda_X - \lambda_Y} e^{(\lambda_X - \lambda_Y) y} \Big/_0^t \\ &= \frac{\lambda_X \lambda_Y}{\lambda_X - \lambda_Y} e^{-\lambda_X t} (e^{(\lambda_X - \lambda_Y) t} - 1) \\ &= \frac{\lambda_X \lambda_Y}{\lambda_X - \lambda_Y} (e^{-\lambda_Y t} - e^{-\lambda_X t}) \end{aligned} \tag{A.2}$$

The calculation of $\int t^k e^{\alpha t} dt$ is straightforward and can also be found in any math-

ematical handbook of formulas e.g. Spiegel [73]. It is:

$$\int t^k e^{\alpha t} dt = e^{\alpha t} \left[\sum_{j=0}^{k} \frac{(-1)^j}{\alpha^{j+1}} \frac{k!}{(k-j)!} t^{k-j} \right]$$

Using this we also give the expression for the definite integration from $[0, T]$. It is:

$$\begin{aligned}
\int_0^T t^k e^{\alpha t} dt = \quad & e^{\alpha T} \sum_{j=0}^{k} \frac{(-1)^j}{\alpha^{j+1}} \frac{k!}{(k-j)!} T^{k-j} - \sum_{j=0}^{k} \frac{-1}{(-\alpha)^{j+1}} \frac{k!}{(k-j)!} 0^{k-j} \Leftrightarrow \\
& \left\{ 0^{k-j} \text{ is always zero unless } j = k \right\} \\
\int_0^T t^k e^{\alpha t} dt = \quad & e^{\alpha T} \sum_{j=0}^{k} \frac{(-1)^j}{\alpha^{j+1}} \frac{k!}{(k-j)!} T^{k-j} + \frac{k!}{(-\alpha)^{k+1}} \Leftrightarrow \\
\int_0^T t^k e^{\alpha t} dt = \quad & e^{\alpha T} \sum_{j=0}^{k} \frac{(-1)}{(-\alpha)^{j+1}} \frac{k!}{(k-j)!} T^{k-j} + \frac{k!}{(-\alpha)^{k+1}} \qquad \text{(A.3)}
\end{aligned}$$

Appendix B

```
//Simulation  programme
//Function balking defines the form of balking.
//Balking can have a form of discouraged arrivals,
// or finite population system (machine interference problem).
//MM is the finite population
// lambda*(MM-n) the breakdown rate were n is the number in the system
//(programme needs minor adjustments when we want n to represent
// the number in the queue)

#include <fstream.h>
#include <cmath>
#include <iostream>
#include <stdlib.h>
#include <algorithm>
#include <string.h>
#include <time.h>

#define MM 10

void vectori(int *& v,int nl,int nh);
void vectord(double *& v,int nl,int nh);
void free_vectori(int * v,int nl);
void free_vectord(double * v,int nl);
double arrival(double LA, int &randomseed);
int service(double *Ser,int &randomseed);
int balking(int qu, int &randomseed);
double ran0(int &idum);
double min(double a, double b);

int main(void){

int k, r, i, sb, place, q, minpl, flag, NN, MAXSIM,
        arrch, sch, maxss, randomseed, count;
char fname[20], otp[20], ch, title[80], blurb[80];
ofstream outp;
char mess[40],sername[80];
double servar,sermean, calccut;
int **T;
```

```
int     arrchange=0,
        cchange=0,
        *n,
        *C,
        ta,
        tc,
        td,
        tb,
        ts;
double  *Ta,
        *Tc,
        *Av,
        *Ba,
        *Tb,
        *per,
        arrmean,
        *Ser,
        rt, maxarr, rate, endtime;

time_t  t1;
char    name[80];
double  *s, minsi, initime, IA, temp, v;

//The programme uses as input and service files
//identical files with the ones used for the DTM programmes
//Service distribution file
        cout << "\n\nEnter the name of the service distribution file  ";
        cin.get(name,20);
        ifstream inn(name);
  if (!inn) {
                strcat(mess,name);
                cout<< "\nProgram Runtime Error\n\n"
                << "\n\n Press the key ENTER to exit program";
                cin.get(ch);
                cin.get(ch);
                exit(0);
        }
        inn.getline(blurb,80);
        inn.getline(blurb,80);
        inn.getline(blurb,80);
        inn >> v >> ws;       //v is the step size
        inn.getline(blurb,80);
        inn >> ts >> ws;
        vectord(Ser,1,ts);
        inn.getline(blurb,80);
        for (i=1;i<=ts;i++)
                inn >>  Ser[i] >> ws;
        inn.close();
//Input file (contains information about arrival rates, number of servers etc)
```

```
        cout << "\n\nEnter the name of the input file  ";
        cin >> fname; //cin.get(fname,20);
        strcpy(otp,fname);
        strcat(otp,".sim");
        outp.open(otp);
        ifstream in(fname);
        if (!in) {
                strcat(mess,fname);
                cout<< "\nProgram Runtime Error\n\n"
                << "\n\n Press the key ENTER to exit program";
                cin.get(ch);
                cin.get(ch);
                exit(0);
        }
        in.getline(blurb,80);
        in.getline(blurb,80);
        in.getline(blurb,80);
        in.getline(blurb,80);
        in >> td >> ws;
        in.getline(blurb,80);
        for(i=1;i<=td;i++) in.getline(blurb,80);
        in.getline(blurb,80);
        in >> endtime >> ws;
        in.getline(blurb,80);
in >> rt >> ws;
        in.getline(blurb,80);        // Read in balking data
        in >> tb >> ws;
        in.getline(blurb,80);
        for (i=1;i<=tb;i++)  in.getline(blurb,80);
        in.getline(blurb,80);        // Read in arrival data
        in >> ta >> ws;
        in.getline(blurb,80);
        vectord(Av,1,ta);
        vectord(Ta,1,ta+1);
        for (i=1;i<=ta;i++) {
                in >> Ta[i] >> rate >> ws;
                Av[i]=rate;
                // if (maxarr<Av[i]) maxarr = Av[i];
        }
        Ta[i]=endtime;
        in.getline(blurb,80);          // Read in server data
        in >> tc >> ws;
        in.getline(blurb,80);
        maxss=0;
        vectori(C,1,tc);
        vectord(Tc,1,tc+1);
        for (i=1;i<=tc;i++) {
                in >> Tc[i] >> C[i] >> ws;
                if ( C[i] > maxss ) maxss=C[i];
```

```
                // maxc= max(maxc,C[i]);
        }
        Tc[i]=endtime;
        in.close();


//-------------------------------------------------------MAIN  ALGORITHM

        NN=(int)(endtime/v);
// Number of epochs :Length of time to run the model in units of service time
        cout << "\n\n  How many times would you like to run the simulation?  ";
        cin >> MAXSIM;
        vectori(n,0,NN-1);
        vectord(s,0,maxss);
        for (r=0; r<NN; r++)
                n[r]=0;
        time(&t1);
        randomseed=rand();
for (k=1; k<=MAXSIM; k++){
      for (i=0; i<maxss; i++)
           s[i]=0;
      arrch=1;
      sch=1;
      IA=arrival(Av[arrch],randomseed);
      initime=0;
      q=0;
      place=0;
      sb=0;
      flag=1;
      r=0;
      minpl=0;
      while (r<NN){
                minsi=1000;
                flag=1;
//we know that the system is empty
                for (i=C[sch]-1; i>-1; i--)
                        if ( s[i] < minsi && s[i]>0) {
                                minsi=s[i];
                                minpl=i;
//we store here the place of the server with the minimum remaining service time
                        }
                        else    if (s[i]==0) place=i;
//we store here the place of a free server
                temp=(minsi!=1000?min(IA,minsi):IA);
                if (r*v < initime+temp) {
// an epoch occurs before the next event, so we need to store the system's state
                        for (i=0; i<C[sch]; i++)
                                    if (s[i]>0) s[i]-=r*v-initime;
                        IA-=r*v-initime;
```

```
                        initime=r*v;
                        n[r]+=q;
                        r++;
                        if (Ta[arrch+1] < r*v) {
//check whether the arrival rate changes
                                arrch++;
                                IA=arrival(Av[arrch],randomseed);
                        }
                        if (Tc[sch+1] < r*v) {
//check whether the number of servers changes
                                sch++;
                                if (C[sch-1]>C[sch]){
//we count again the busy servers, when the number of servers is decreased
                                        sb=0;
                                        for (i=C[sch]-1; i>-1; i--)
                                        if ( s[i] > 0) sb++;
                                }
                                else {
                         for (count=C[sch]-C[sch-1]-1; count>-1; count--){
//when the number of servers is increased
                                           if (q>0){
//if the queue is not empty, move someone from there to the service
     s[C[sch-1]+count]=v*service(Ser, randomseed)+(0.5-ran0(randomseed))/1000;
                                                   q--;
                                                   sb++;
                                           }
                                           else s[C[sch-1]+count]=0;
//if the queue is empty make the new server idle
                                        }
                                }
                        }
                }
//The check about change of servers and arrival rate is done only at epochs
                else {   //It is an event that occurs now
                        if (IA==temp) {
//Check whether an arrival is going to be the next event
                                flag=0;
                                for (i=0; i<C[sch]; i++)      // maxss
                                    if (s[i]>0) s[i]-=temp;
//The time counter (current time or time 0) is going to change
//so we need to update the service already done
                                initime+=IA;
                                IA=arrival(Av[arrch],randomseed);
                                if (sb==C[sch]) q+=balking(q,randomseed);
// When sb==number of servers and q=0 we can still have balking
                                else {
                                        s[place]=balking(q,randomseed);
//temporary allocate in s[place] whether there is an entry
                                        sb+=s[place];
```

```
// when we have an entry the balking function returns 1 else 0
                              if (s[place]) s[place]=v*service(Ser,randomseed);
//Since an idle server exists allocate the arriving
                                 }
//if the arrival will not entry the system,
//nothing changes, still the time was updated
                           }
                           if (minsi==temp){
//Check whether a departure is going to be the next event
                                 if (flag){
                                        if (minsi!=1000)
                                               for (i=0; i<C[sch]; i++)
                                                      if (s[i]>0) s[i]-=temp;
                                        initime+=temp;
                                        IA-=temp;
                                 }
                                 sb--;
                                 if ( q>0 ){
//If the queue is not empty move someone
//from the queue to the server that just became idle
                                        s[minpl]=v*service(Ser,randomseed);
                                        q--;
                                        sb++;
                                 }
                           }
                  } //else loop
        }//r Loop
} //k Loop
////////////////////////////////Output
// Initialise output files
        cout <<"\n Output file is : " << fname << ".sim \n";
//        outp << "Data file name " << fname << ", created " << ctime(&t1);
//        outp << "from general Input file " << fname <<endl;
//        for (i=1;i<=ta;i++)
//              outp << Ta[i] << "\t" << Av[i] << endl;
//        outp<<"Time"<<"\t"<<"No. in the system"<<endl;
        for (i=0; i<NN; i++){
                outp<<i*v<<"\t"<<(double)n[i]/MAXSIM<<endl;
                i++;
        }
//        outp<<"Runtime was " << difftime(time(NULL),t1)<<" seconds"<<endl;
        outp.close();
        free_vectori(n,0);
        free_vectori(C,1);
        free_vectord(Ta,1);
        free_vectord(Tc,1);
        free_vectord(Av,1);
        free_vectord(Ser,1);
        free_vectord(s,0);
```

```
return(1);
}

void vectori(int *& v,int nl,int nh){
// create vector of integers

char ch;

        v=new int[nh-nl+1];
        if (!v) {
                cout<< "\nProgram Runtime Error\n\n"
                << "\n\n Press RETURN a couple of times to exit program";
                cin.get(ch);
                cin.get(ch);
                exit(0);
        }
        v-= nl;
}

void vectord(double *& v,int nl,int nh){
// create vector of double
char ch;

        v=new double[nh-nl+1];
        if (!v) {
                cout<< "\nProgram Runtime Error\n\n"
                << "\n\n Press Enter to exit program";
                cin.get(ch);
                cin.get(ch);
                exit(0);
        }
        v-= nl;
}
void free_vectori(int *v,int nl)
// free vector of integers
{
        delete(v+nl);
}

void free_vectord(double *v,int nl)
// free vector of double
{
        delete(v+nl);
}

double arrival(double LA, int &randomseed){

double temp;
temp=0;
```

```
while (temp==0 || temp==1)
        temp=(double)ran0(randomseed);
return   -log(temp)/LA;
}

int service(double *Ser, int &randomseed){

double temp, cum;
int flag2, i;

flag2=1;
i=1;
cum=Ser[i];
temp=(double)ran0(randomseed);
while (flag2)
        if (temp <= cum) {
                flag2=0;
                return i;
        }
        else {
                i++;
                cum+=Ser[i];
        }
}
int balking(int qu, int &randomseed){

double temp, coef=0.9;

temp=(double)ran0(randomseed);
if (temp < pow(coef,1+qu) ) return 1;
//if (temp < (1.0-(qu / double (MM))) ) return 1;
else return 0;
}

double ran0(int &idum){

const int IA = 16807, IM=2147483647, IQ=127773, IR=2836, MA=123459876;
const double AM=1.0/double (IM);
int k;
double ans;

idum ^= MA;
k=idum/IQ;
idum=IA*(idum-k*IQ)-IR*k;
if (idum <0) idum+=IM;
ans=AM*idum;
idum^=MA;
return ans;
```

```
}

double min(double a, double b){

if ( a > b ) return b;
else return a;
}
```

Appendix C

In order to prove that $\sum P_t(n) = 1$, where $P_t(n)$ is given by Equation (7.2) we apply the following trapezoidal rule or else 2-point Newton-Cotes formula [73]:

$$\int_{x_1}^{x_2} f(x)dx = \frac{1}{2}(x_2 - x_1)(f(x_1) + f(x_2)) - \frac{1}{12}(x_2 - x_1)^3 f''(\xi)$$

which for $x_2 - x_1 = 1$ gives:

$$\int_{x_1}^{x_2} f(x)dx = \frac{1}{2}(f(x_1) + f(x_2)) - \frac{1}{12}f''(\xi) \tag{C.1}$$

The final term gives the amount of error (which, since $x_1 < \xi < x_2$, is no worse than the maximum value of $\frac{1}{12}f''(\xi)$ in this range). Since our aim is to use the above formula for $f(n) = P_t(n)$ from Equation (7.2), $f(x)$ has the form of a normal function so we can find an analytical expression for $f''(x)$. It is:

$$f''(x) = \begin{cases} 0, & x \ll s \text{ or } x \gg s \\ \frac{-1}{\sigma^3\sqrt{2\pi}}[1 - \frac{(x-m)^2}{\sigma^2}]e^{\frac{-(x-m)^2}{2\sigma^2}}, & elsewhere \end{cases} \tag{C.2}$$

Having in mind that the second derivative of a function is the rate of change of the first derivative, $f''(x)$ becomes negligible for smooth functions. In order to have a smooth normal curve, we need a large variance (σ^2), so in this case we can approximate the integral by:

$$\int_{x_1}^{x_2} f(x)dx = \frac{1}{2}(f(x_1) + f(x_2))$$

Expanding the above equation for more than one intervals gives:

$$\begin{aligned}\int_{x_1}^{x_n} f(x)dx &= \int_{x_1}^{x_2} f(x)dx + \int_{x_2}^{x_3} f(x)dx + \ldots + \int_{x_{n-1}}^{x_n} f(x)dx \\ &= \frac{1}{2}(f(x_1) + f(x_2)) + \frac{1}{2}(f(x_2) + f(x_3)) + \ldots + \frac{1}{2}(f(x_{n-1}) + f(x_n)) \\ &= \frac{1}{2}(f(x_1) + f(x_n)) + \sum_{x_2}^{x_{n-1}} f(x_k)\end{aligned}$$

We use the above equation for $f(n) = P_t(n)$ defined by Equation (7.2) and $x_1 = 0$, $x_n = \infty$ in order to prove that $\sum_{k=0}^{\infty} P_t(k) = 1$. We have:

$$\begin{aligned}\sum_{k=0}^{\infty} P_t(k) &= \int_0^{\infty} P_t(x)dx + \frac{1}{2}[P_t(0) + P_t(\infty)] \\ &= \int_0^{\infty} P_t(x)dx = 1\end{aligned}$$

Bibliography

[1] V. Mehrotra, "Ringing up big business," *OR/MS*, vol. 24, no. 4, pp. 18–24, 1997.

[2] Incomes Data Services Ltd, *Pay and conditions in call centres 2004*. 2004.

[3] UK Department of Trade and Industry, *The UK Contact Centre Industry: A Study*. 2004.

[4] T. A. Grossman, D. A. Samuelson, S. L. Oh, T. R. Rohleder, "Call Centers," *Encyclopedia of Operations Research and Management Science*, November 1999. 2nd Edition, S. I. Gass and C. M. Harris, editors.

[5] National Audit Office, *Using Call Centres to Deliver Public Services*. London: The Stationary Office, 2002.

[6] B. Cleveland and J. Mayben, *Call Center Management on Fast Forward: Succeeding In Today's Dynamic Inbound Environment*. Annapolis, Maryland: Call Centre Press, 1997.

[7] A. Mandelbaum, A. Sakov, and S. Zeltyn, "Empirical analysis of a call center," *working paper*, March 2001.

[8] G. Koole, "Mathematical Modeling of Call Centers," tech. rep., Vrije Universiteit, Amsterdam, April 2001.

[9] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, L. Zhao, "Statistical analysis of a telephone call center: A queueing-science perspective," *working paper*, November 2002.

[10] D. M. Rappaport, "Key role of integration in call centres," *Business Communications Review*, pp. 44–48, July 1996.

[11] W. Whitt, "Improving service by informing customers about anticipated delays," *Management Science*, vol. 45, pp. 192–207, 1999.

[12] A. Mandelbaum, W. A. Massey, M. I. Reiman and B. Rider, "Time varying multiserver queues with abandonment and retrials," *Proceedings of the 16th International Teletraffic Conference*, pp. 355–364, 1999.

[13] L. Kleinrock, *Queueing Systems, Volume I: Theory.* New York: John Wiley & Sons, 1975.

[14] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory.* John Wiley & Sons, 1985.

[15] F. S. Hillier and O. S. Yu, *Queueing Tables and Graphs.* New York: North Holland, 1981.

[16] L. P. Seelen, H. C. Tijms, M. H. Van Hoorn, *Tables for Multi-Server Queues.* Amsterdam: North Holland, 1985.

[17] O. P. Sharma, *Markovian Queues.* New Delhi: Allied Publisherts, 1997.

[18] D. Worthington and A. Wall, "Using the discrete time modelling approach to evaluate the time-dependent behaviour of queueing systems," *Journal of the Operational Research Society*, vol. 50, pp. 777–788, 1999.

[19] Á. Ingólfsson, E. Akhmetshina, S. Budge, Y. Li, X. Wu, "A survey and experimental comparison of service level approximation methods for non-stationary $M(t)/M/s(t)$ queueing systems," *working paper, University of Alberta*, June 2003.

[20] G. F. Newell, "Queues with time dependent rates: Part I. The transition through saturation; Part II. The maximum queue and the return to equilibrium; Part III.

A mild rush hour," *Journal of Applied Probability*, vol. 5, pp. 436–451, 579–590, 591–606, 1968.

[21] L. Kleinrock, *Queueing Systems, Volume II: Computer Applications*. New York: John Wiley & Sons, 1975.

[22] A. Duda, "Diffusion Approximations for Time-Dependent Queueing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 4, pp. 905–918, September 1986.

[23] O.B. Jennings, A. Mandelbaum, W.A. Massey, W. Whitt, "Server staffing to meet time-varying demand," *Management Science*, vol. 42, no. 10, pp. 1383–1394, 1996.

[24] L. Green, P. Kolesar, and S. Svoronos, "Some effects of nonstationarity on multi-server markovian queuing systems," *Operations Research*, vol. 39, pp. 502–511, 1991.

[25] L. Green and P. Kolesar, "The pointwise stationary approximation for queues with nonstationary arrivals," *Management Science*, vol. 37, January 1991.

[26] M. F. Neuts, "The single server queue in discrete time-numerical analysis I," *Naval Res Logistic Q*, vol. 20, pp. 297–304, 1973.

[27] S.E. Omosigho and D. J. Worthington, "The single server queue with inhomogeneous arrival rate and discrete service time distribution," *European Journal of Operational Research*, vol. 22, pp. 397–407, 1985.

[28] M. Brahimi and D. J. Worthington, "The finite capacity multi-server queue with inhomogeneous arrival rate and discrete service time distribution-and its application to continuous service time problems," *European Journal of Operational Research*, vol. 50, pp. 314–324, 1991.

[29] A.J. Brigandi, D.R. Dargon, M.J. Sheehan and T. Spencer, "AT&T's call processing simulator (cpps) operational design for in-bound call centers," *Interfaces*, no. 24, pp. 6–28, 1998.

[30] A. Brandt and M. Brandt, "On the $M(n)/M(n)/s$ queue with impatient calls," vol. 35, pp. 1–18, 1999.

[31] A. Mandelbaum and N. Shimkin, "A model for rational abandonments from invisible queues," *Queueing Systems*, no. 36, pp. 141–173, 2000.

[32] N. Gans, G. Koole, and A. Mandelbaum, "Telephone call centers: Tutorial, review, and research prospects," *Manufacturing and Service Operations Management*, no. 5, pp. 79–141, 2003.

[33] A. Mandelbaum, W. A. Massey, M. I. Reiman, B. Rider, and A. Stolyar, "Queue lengths and waiting times for multiserver queues with abandonment and retrials," *working paper*, 2000.

[34] G. Koole and A. Mandelbaum, "Queueing models of call centres: An introduction," *working paper, abridged version appeared in Annals of Operations Research*, vol. 112, 2002.

[35] S. G. Eick, W. A. Massey and W. Whitt, "$M(t)/G/\infty$ queues with sinusoidal arrival rates," *Operations Research*, vol. 39, no. 2, pp. 241–252, 1993.

[36] S. G. Eick, W. A. Massey and W. Whitt, "The physics of the $M(t)/G/\infty$ queue," *Operations Research*, vol. 41, no. 4, pp. 731–742, 1993.

[37] M. F. Neuts, *Matrix-geometric Solutions in Stochastic Models:An Algorithmic Approach.* Baltimore: The Johns Hopkins University Press, 1981.

[38] R. A. Marie and J. M. Pellaumail, "Steady-state probabilities for a queue with a general service distribution and state-dependent arrivals," *IEEE Transactions on Software Engineering*, vol. 9, no. 1, pp. 109–113, 1983.

[39] W. C. Driscoll, "$E_m/E_k/S$ queuing systems with state-dependent arrivals," *Computers and Industrial Engineering*, vol. 28, no. 3, pp. 473–484, 1995.

[40] U. C. Gupta and S. Rao, "Computing steady state probabilities in $\lambda(n)/G/1/K$ queue," *Performance Evaluation*, vol. 24, pp. 265–275, 1996.

[41] M. Pidd, *Computer Simulation in Management Science*. Chichester: John Wiley & Sons, 1998.

[42] G. Iazeolla, P. J. Curtois, and A. Hordijk,(Eds.), *Mathematical Computer Performance and Reliability*. North-Holland: Elsevier, 1984.

[43] M. E. Woodward, *Communication and Computer Networks: Modelling with Discrete-Time Queues.* London: Pentech Press, 1993.

[44] H. M. Srivastava and B. R. K. Kashyap, *Special Functions in Queuing Theory.* New York: Academic Press, 1982.

[45] H. P. Galliher and R. C. Wheeler, "Nonstationary queuing probabilities for landing congestion of aircraft," *Operations Research*, vol. 6, pp. 264–275, 1958.

[46] S. Dafermos and M. F. Neuts, "A single server queue in discrete time," *Cahiers du Centre de Recherche Operationnelle*, vol. 13, pp. 23–40, 1971.

[47] E. M. Klimko and M. F. Neuts, "The single server queue in discrete time-numerical analysis II," *Naval Res Logistic Q*, vol. 20, no. 2, pp. 305–319, 1973.

[48] M. F. Neuts and E. M. Klimko, "The single server queue in discrete time-numerical analysis III," *Naval Res Logistic Q*, vol. 20, pp. 557–567, 1973.

[49] D. L. Minh, "The discrete-time single-server queue with time-inhomogeneous compound poisson input and general service time distribution," *Journal of Applied Probability*, vol. 15, pp. 590–601, 1978.

[50] S. E. Omosigho, *Approximate Methods for Single Server Queues with Time Dependent Arrival Rate*. PhD thesis, Lancaster University, 1985.

[51] M. Brahimi, *Approximating Multi-Server Queues with Inhomogeneous Arrival Rates and Continuous Service Time Distributions*. PhD thesis, Lancaster University, 1990.

[52] A.D. Wall, *Extending the scope of discrete time models to provide practical results for continuous time queueing systems*. PhD thesis, Lancaster University, 1995.

[53] S.E. Omosigho and D. J. Worthington, "An approximation of known accuracy for single server queues with inhomogeneous arrival rate and continuous service time distribution," *European Journal of Operational Research*, vol. 33, pp. 304–313, 1988.

[54] A. D. Wall and D. J. Worthington, "Using discrete distributions to approximate general service time distributions in queueing models," *Journal of the Operational Research Society*, vol. 45, no. 12, pp. 1398–1404, 1994.

[55] J. J. Hunter, *Mathematical Techniques of Applied Probability, Vol. 2 Discrete Time Models: Techniques and Applications*. Academic Press, 1983.

[56] H. Takagi, *Queueing Analysis, vol. 3: Discrete-Time Systems*. North-Holland, 1993.

[57] H. Bruneel and B. G. Kim, *Discrete-Time Models for Communication Systems Including ATM*. Kluwer Academic Publishers, 1992.

[58] P. Gao, S. Wittevrongel, H. Bruneel, "Discrete-time multiserver queues with geometric service times," *Computers & Operations Research*, vol. 31, pp. 81–99, 2004.

[59] P. Parthasarathy and N. Selvaraju, "Exact transient solution of a state-dependent queue in discrete time," *Operations Research Letters*, no. 28, pp. 243–248, 2001.

[60] E. S. Pearson, *Karl Pearson's Early Statistical Papers.* London: Cambridge University Press, 1948.

[61] W. Whitt, "Decomposition approximations for time-dependent markovian queueing networks," *Operations Research Letters*, no. 24, pp. 97–103, 1999.

[62] S. Ramesh and G. N. Rouskas, "Computing blocking probabilities in multiclass wavelength-routing networks with multicast calls," *IEEE Journal on Selected Areas in Communications*, vol. 20, January 2002.

[63] W. Feller, *An Introduction to Probability Theory and Its Applications, volume 2.* John Wiley & Sons, 1971.

[64] W. H. Press, *Numerical Recipes in C++: The art of scientific computing.* Cambridge: Cambridge University Press, 2002.

[65] L.G. Peck and R.N. Hazelwood, *Finite Queuing Tables.* Cambridge, Mass.: John Wiley & Sons, 1958.

[66] L. V. Green and P. J. Kolesar, "The lagged PSA for estimating peak congestion in multiserver markovian queues with periodic arrival rates," *Management Science*, vol. 43, pp. 80–87, January 1997.

[67] K. M. Malone and A. Ingólfsson, "On the timing of the peak mean and variance for the number of customers in an $M(t)/M(t)/1$ queueing system," *working paper, Massachusetts Institute of Technology, Operations Research Center*, p. 19, Oct 1994.

[68] C. Palm, *Intensity Variations in Telephone Traffic.* Amsterdam: North-Holland, 1988. (translation of 1943 article in Ericsson Technics, 44, 1-189).

[69] A. Y. Khintchine, *Mathematical Methods in the Theory of Queueing.* London: Charles Griffin and Co., 1960. (translation of 1955 Russian book).

[70] D. Worthington, "A normal approximation for the $M(\lambda_Q)/G/S//N$ queueing model," *Journal of the Operational Research Society*, vol. 39, no. 10, pp. 973–976, 1988.

[71] E. Chassioti and D. J. Worthington, "A new model for call centre queue management," *Journal of the Operational Research Society*, vol. 55, pp. 1352–1357, 2004.

[72] R. W. Wolff, *Stochastic Modeling and the Theory of Queues.* Prentice Hall, 1989.

[73] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables.* Schaum's outline series, McGraw-Hill, 1968.